

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Факультет прикладної математики**

**ПРИКЛАДНА МАТЕМАТИКА
ТА КОМП'ЮТИНГ
ПМК' 2022**



П'ятнадцята конференція магістрантів та аспірантів

Київ, 16 – 18 листопада 2022 р.

Збірник тез доповідей

**Рекомендовано Вченою радою
факультету прикладної математики**

Просвіта
Київ
2022

**MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
NATIONAL TECHNICAL UNIVERSITY OF UKRAINE
“IGOR SIKORSKY KYIV POLYTECHNIC INSTITUTE”
APPLIED MATHEMATICS FACULTY**

**APPLIED MATHEMATICS
AND COMPUTING
AMC' 2022**



**Fifteenth Scientific Conference of Master
and Postgraduate Students
Kyiv, November 16-18, 2022**

PROCEEDINGS

**Recommended by the Academic Council
of Applied Mathematics Faculty**

Prosvita
Kyiv
2022

УДК 519.85+004](06)
ББК 22.18я43+32.973я43
П75

Рекомендовано Вченою радою факультету прикладної математики
(протокол № 4 від 28.11.2022)

Редакційна колегія:

д.т.н., проф. Дичка І.А., д.т.н., проф. Чертов О.Р., д.т.н., проф. Романкевич В.О., д.т.н., доц. Сулема Є.С., д.т.н., проф. Легеза В.П., д.т.н., проф. Зайцев В.Г., д.т.н., проф. Терейковський І.А., к.т.н., доц. Маслянюк П.П., к.т.н., доц. Марченко О.І., к.т.н., доц. Тесленко О.К., к.т.н., доц. Заболотня Т.М., к.т.н. доц. Петрашенко А.В.

Головний редактор к. т. н., доц. Сирота С. В.

П75 Прикладна математика та комп'ютеринг. ПМК, 2022 : п'ятнадцята наук. конф. магістрантів та аспірантів, Київ, 16—18 лист. 2022 р. : зб. тез доп. / [редкол.: Дичка І. А. та ін.]. — К. : Просвіта, 2022. — 532 с.

ISBN 978-617-7010-23-3 (Друк)
ISBN 978-617-7010-24-0 (Online)

ISBN 978-617-7010-25-7 (Eng. Title)

У збірнику опубліковано тези доповідей на конференції які представляють наукові результати здобувачів освіти та їх наукових керівників. Опубліковані результати є частиною атестаційних робіт.

УДК 519.85+004](06)
ББК 22.18я43+32.973я43

UDC 519.85+004](06)

Recommended by the academic council of Applied Mathematics Faculty
(Protocol No. 4 dated November 28, 2022)

Editorial board:

Dr. prof. Dychka, I., dr. prof. Chertov, O., dr. prof. Romankevich, V., dr. prof. Sulema, dr. prof. Legeza, V., dr. prof. Zaitsev, V., dr. prof. Trejkovski, I., E., dr. Maslianko, P., dr. Marchenko, O., dr. Teslenko, O., dr. Zabolotnia, T. and dr. Petrashenko, A.

Chief editor dr. Syrota S.

Dychka, I., & (Eds.). (2022). *Fourteenth Scientific Conference of Masters and Postgraduate Students Kyiv, November 16-18, 2022 Proceedings* (p. 532). Kyiv, Ukraine: Prosvita Ltd.

ISBN 978-617-7010-23-3 (Print)
ISBN 978-617-7010-24-0 (Online)

ISBN 978-617-7010-25-7 (Eng. Title)

Abstracts of reports at the conference are published in the Proceedings, which present the scientific results of the students and their academic supervisors. The published results are part of the qualification papers.

Використання матеріалів збірника можливе за умови обов'язкового посилання.

The materials of the collection can be used under the condition of a mandatory reference.

ISBN 978-617-7010-23-3 (Print)
ISBN 978-617-7010-25-7 (Eng. Title)
ISBN 978-617-7010-24-0 (Online)

© Факультет прикладної математики КПІ ім. Ігоря Сікорського, 2022

© Applied mathematics faculty Igor Sikorsky Kyiv Polytechnic Institute, 2022

УДК 519.688

К. ф.-м. н., старший викладач Бай Ю. П., студент Городецький Д. С.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ПРОГНОЗУВАННЯ РИЗИКІВ ВИНИКНЕННЯ ДІАБЕТУ

Abstract

Yulia Bai, PhD; Dmytro Horodetskyi, student

Mathematics and software for a diabetes risk prediction system

This article examines diabetes prediction algorithms based on a set of real patient data. The typical set of classical machine learning methods is used, such as logistic regression, support vector machine, the method of k nearest neighbors, decision tree and random forest. These methods are evaluated to pick the best diabetes prediction method for a real dataset.

Вступ

Діабет є однією з найбільш розповсюджених хвороб у всьому світі, яка потребує постійного контролю за пацієнтом, своєчасного прийому ліків протягом всього життя. Діабет має дуже серйозні наслідки для людського організму і може загрожувати життю людини [1]. Актуальною є задача прогнозування ризиків виникнення діабету на основі реальних поточних даних про пацієнта та своєчасне призначення профілактичних заходів з метою запобігання захворюванню.

На сьогоднішній день існує велика кількість теоретичних робіт в сфері прогнозування виникнення діабету за допомогою методів машинного та глибокого навчання [2, 3]. В даній роботі буде виконано порівняння існуючих методів машинного навчання та вибір найкращого методу для набору реальних даних.

Постановка задачі

Метою даної роботи є розробка математичного та програмного забезпечення системи прогнозування ризиків виникнення діабету. На початковому етапі необхідно провести порівняння існуючих методів машинного навчання на реальному наборі даних про пацієнтів з метою

визначення оптимального методу. При застосуванні того чи іншого методу необхідно провести класифікацію та визначити за наданими параметрами, чи належить людина до категорії здорових або хворих на цукровий діабет.

Опис набору вхідних даних

Використання правильного набору вхідних даних відіграє важливу роль у вирішенні будь-якої проблеми в машинному навчанні. В даній роботі використовується датасет, зібраний Національним інститутом діабету, захворювань системи травлення та нирок в Сполучених Штатах Америки, відомий також під назвою “Pima Indians dataset” [4]. Датасет налічує близько 800 записів, де параметрами є рівень глюкози, рівень інсуліну, кров’яний тиск, індекс маси тіла, вік тощо, рис. 1. Одним із параметрів є наявність (1) чи відсутність (0) діабету у пацієнта.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Рис. 1. Вигляд датасету

Перед застосуванням алгоритмів машинного навчання, вхідний датасет необхідно нормувати за наступною формулою [5]:

$$z = \frac{(x-u)}{s} \quad (1)$$

де u – середнє значення, s – стандартне відхилення.

Для подальшого поділу датасету на навчальну та тестову вибірки використовується стратифікація [6]. Стратифікація – процес розділу даних з вибірки на однорідні підгрупи перед безпосереднім відбором вибірки. Вона дозволяє створити такий тестовий набір, який найкраще представляє всю досліджувану сукупність.

Опис алгоритму

Для розв'язання поставленої задачі буде використовуватися одразу декілька методів – логістична регресія, дерева рішень, метод опорних векторів, метод k -найближчих сусідів та випадковий ліс [5]

Логістична регресія — статистичний регресійний метод, що моделює ймовірність події шляхом його порівняння з логістичною кривою. В якості результату застосування методу видається ймовірність бінарної події, 0 чи 1.

Метод опорних векторів (SVM – support vector machine) — це метод аналізу даних для класифікації та регресійного аналізу за допомогою моделей з керованим навчанням. Ціллю SVM класифікатора є знаходження гіперплощини в N -вимірному просторі, де N – кількість параметрів, що недвозначно класифікує дані, вирішуючи, з якої сторони вони будуть знаходитись відносно гіперплощини.

Метод k -найближчих сусідів – непараметричний метод навчання з учителем, що використовується для класифікації та регресії. При класифікації k -NN результатом є належність до певного класу. Об'єкт класифікується за допомогою множини голосів його сусідів і відноситься до того класу, до якого відноситься більшість з його k найближчих сусідів, рис. 2.

Дерево рішень – класичний керований метод машинного навчання для задач регресії та класифікації. Він має структуру дерева, що описує процес класифікації об'єктів на основі параметрів. Алгоритм ділить навчальну вибірку на підмножини відповідно до значення параметра поточного вузла. Процес продовжується для кожної з утворених підмножин, рекурсивно формуючи дерево рішень.

Випадковий ліс – мультифункціональний метод машинного навчання, що виводить концепцію дерев рішень на новий рівень. Класифікація відбувається за допомогою генерації багатьох дерев рішень. Коли за допомогою цього методу відбувається прогнозування, кожне згенероване дерево дасть свій прогноз і загальний результат лісу буде обрахований на

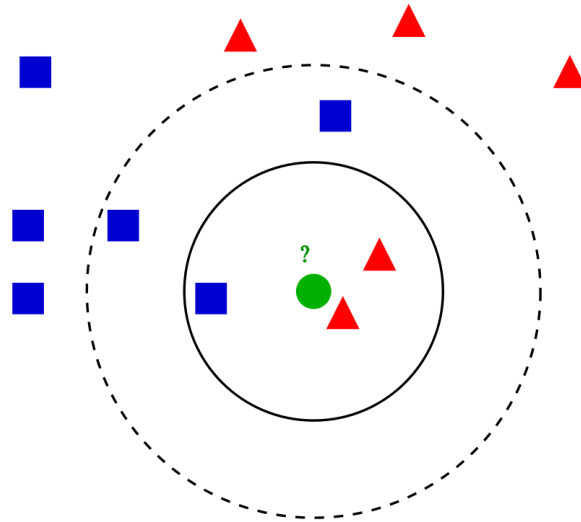


Рис. 2. Приклад k -NN класифікації

основі рішень кожного з дерев. Для задачі регресії це буде середнє значення або середнє арифметичне серед прогнозів по всіх деревах.

Перехресне затвердження (cross-validation) – певна методика валідації побудованої моделі, необхідна для оцінювання того, наскільки результати статистичного аналізу узагальнюватимуться на незалежному наборі даних. Перехресне затвердження є методом, що використовує різні частини даних для тестування та тренування моделі на різних ітераціях. Мета цього методу полягає в тому, щоб протестувати можливість моделі прогнозувати нові дані, які не були використані для її тренування, щоб вирішити таку проблему, як перенавчання.

Висновок про те, чи є пацієнт хворим на діабет приймається на основі застосування перелічених вище п'яти методів прогнозування. Розроблена система має архітектуру, представлену на рис. 3.



Рис. 3. Архітектура системи

Результати моделювання

Створена система була застосована для аналізу стандартизованого датасету, утвореного на основі реальних даних [4]. Результати обчислення тренувальної та тестової точності обраних методів представлені в табл. 1.

Як показують проведені розрахунки, точність перших трьох методів коливається в межах 0.77–0.82, але найвищу точність на тестовій вибірці має логістична регресія.

Таблиця 1

Порівняння точності розглянутих методів

Метод	Тренувальна точність	Тестова точність
Логістична регресія	0.76910	0.80208
Метод опорних векторів	0.77083	0.78646
Метод k-найближчих сусідів	0.81597	0.77083
Дерево рішень	1.00000	0.68229
Випадковий ліс	1.00000	0.78125

Таблиця 2

Точність методів після застосування перехресного затвердження

Метод	Точність
Логістична регресія	0.779956
Метод опорних векторів	0.773411
Метод k-найближчих сусідів	0.742139
Дерево рішень	0.680947
Випадковий ліс	0.764269

Проте отримані результати вказують на те, що методи дерева рішень та випадкового лісу є перенавченими, їх тренувальна точність дорівнює одиниці. Отже, виникає необхідність використати перехресне затвердження. Результати обчислення точності методів після використання перехресного затвердження представлені в табл. 2.

З отриманих результатів можна зробити висновок, що для досліджуваного датасету логістична регресія має найвищу точність для задачі прогнозування ризиків виникнення діабету. Матриця невідповідностей [5, 6] для логістичної регресії має вигляд, представлений на рис. 4.

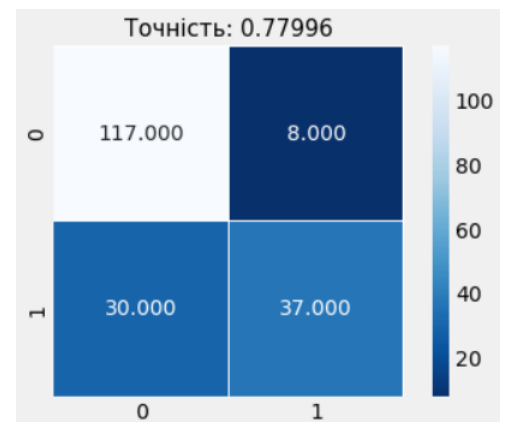


Рис. 4. Матриця невідповідностей

Висновки

Розроблено математичне та програмне забезпечення системи прогнозування ризиків виникнення діабету на основі відомих даних про пацієнтів. В системі застосовано низку класичних методів машинного

навчання, серед яких визначається найкращий метод для заданого набору вхідних даних. Запропоновані методи протестовані на реальному датасеті Національного інституту діабету, захворювань системи травлення та нирок Сполучених Штатів Америки. В результаті проведеного дослідження встановлено, що для обраного датасету найкращим методом для прогнозування ризику виникнення діабету є логістична регресія. Результуюча точність алгоритму складає 0.816, а із застосуванням перехресного затвердження – 0.78.

Література

1. *American Diabetes Association*. Diagnosis and classification of diabetes mellitus. *Diabetes Care*, 2012. S64-71. doi: 10.2337/dc12-s064.
2. *Larabi-Marie-Sainte S., Aburahmah L., Almohaini R., Saba T.* Current techniques for diabetes prediction: review and case study // *Appl. Sci.* 2019. - 9(21):4604, doi: 10.3390/app9214604
3. *Zia U.A., Khan N.* Predicting diabetes in medical datasets using machine learning techniques // *Int. J. Sci. Eng. Res.* 2017. - 8(5):257–267.
4. *Schulz L.O., Bennett P.H., Ravussin E., Kidd J.R., Kidd K.K., Esparza J., Valencia M.E.* Effects of traditional and western environments on prevalence of type 2 diabetes in Pima Indians in Mexico and the US // *Diabetes Care.* – 2006. - 29(8):1866–1871
5. *Mitchell T.* *Machine Learning.* - New York: McGraw Hill, 1997. – 421 p.
6. *Bishop, C. M.* *Pattern Recognition and Machine Learning.* – Springer, 2006. – 758 p.

УДК 519.688

Канд. фіз.-мат. наук Бай Ю.П., студент Лисий П.О.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

ЗАСТОСУВАННЯ МЕТОДУ WORD2VEC ПРИ ПОБУДОВІ КОНТЕНТНО-ОРІЄНТОВАНОЇ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ

Abstract

Yulia Bai, PhD; Pavlo Lysyi, student

Application of the word2vec embeddings selection method for building a content-based recommendation system

One of the actual machine learning topics is the task of comparing textual data and finding the most similar words, sentences or even whole texts. There are many approaches to this task, the main ones being based on the so-called "one-hot-encoding". Modern research produces more capable alternatives such as word2vec embeddings for finer control and precision. This algorithm was used to process news portal content to build recommendation system in order to boost the conversion rate.

Вступ

На сьогоднішній день важливою є задача порівняння текстових даних та знаходження максимально схожих слів, речень або навіть цілих текстів. Є багато підходів до розв'язання цієї задачі, більшість з яких будуються на так званому алгоритмі «one-hot-encoding» [1]. Але цей підхід є застарілим, на зміну йому приходять нові сучасні рішення, зокрема, метод побудови простору ембеддингів word2vec, що дозволяє досягти значно точніших результатів [2]. В даній роботі алгоритм word2vec буде використано для обробки контенту новинного порталу та побудови рекомендаційної системи, яка збільшує кількість конверсійних переходів користувачів.

Постановка задачі

В якості вхідного набору даних виступають текст та векторний простір, побудований на великій базі даних текстів, призначеній для навчання нейронної мережі. Задача полягає в тому, щоб перевести заданий текст у числовий вигляд та отримати можливість порівнювати тексти за змістовним навантаженням. Основна ідея полягає в перетворенні заданого тексту на множину векторів, оскільки нейронна мережа маніпулює тільки числами. Задача ускладнюється різним варіантом написання слів,

синонімами, формою подачі інформації, різної довжиною слів тощо. Розглянемо основні алгоритми обробки і порівняння текстів.

Алгоритм «one-hot encoding»

Одним із найпопулярніших рішень для порівняння двох текстів є метод «one-hot encoding» [1], що працює наступним чином. Алгоритм ділить речення на слова та кожному з них присвоює 1 або 0 в масиві значень. Приклад роботи алгоритму «one-hot encoding» для тексту «The cat sat on the mat» зображений на рис. 1.

Основні недоліки даного підходу виявляються при необхідності врахування чіткої відповідності між словами. Наприклад, слова «cat» та «cats» будуть перетворені у вектор в форматі $[[1,0], [0,1]]$, де перша складова вектора відповідає за слово «cat», а друга – за слово «cats». Оскільки в цьому випадку кожному слову ставиться у відповідність окрема булева змінна у векторі, то між отриманими векторами немає ніякої схожості, хоча це семантично близькі слова. Цю проблему вирішують шляхом «нормалізації слів», при використанні якої два слова «cat» та «cats» зводяться до одного слова «cat». В цьому випадку складові вектора стають ідентичними $[[1,0], [1,0]]$, що є некоректним.

При роботі з реальними текстами при застосуванні підходу «one-hot encoding» виникають такі проблеми, як врахування малих та великих літер, врахування знаків пунктуації тощо. При видаленні всіх знаків пунктуації отримується текст, у якому явно втрачено сенсове навантаження. Також досить серйозною проблемою даного методу є те, що алгоритм не враховує синонімічність слів, речень. Наприклад, речення «Hi, I'm Pasha» та «Hello, my name is Pavlo» не будуть мати нічого спільного при їх розписуванні за допомогою методу «one-hot encoding», хоча, очевидно, що ці два речення синонімічні. Саме тут дослідники приходять до того, що необхідно застосовувати значно розумніший метод, наприклад, метод векторного простору ембеддингів.

Метод векторного простору ембеддингів

У сфері обробки природних мов (Natural Language Processing, NLP) зазвичай працюють зі словниками, що складаються з тисяч слів. Словники вводять у модель з використанням методики швидкого кодування «one-hot encoding», що в математичному сенсі рівносильне наявності окремого

The cat sat on the mat

The: [0 1 0 0 0 0 0]

cat: [0 0 1 0 0 0 0]

sat: [0 0 0 1 0 0 0]

on: [0 0 0 0 1 0 0]

the: [0 0 0 0 0 1 0]

mat: [0 0 0 0 0 0 1]

Рис.1. Приклад роботи алгоритму «one-hot encoding»

вектора для кожного зі слів. Коли слово передається в модель, у відповідному елементі вектору записується одиниця, тоді як всі інші елементи вектору дорівнюють нулю. Це призводить до появи дуже сильно розріджених наборів даних. Вирішення проблеми полягає у створенні векторного простору ембеддингів. Ключова ідея ембеддингів полягає в переведенні слів у n-вимірний простір, в якому буде можливо записати n-вимірний вектор та порівнювати одержані вектори. Застосування ембеддингів дає можливість зменшення розмірності ознак для підвищення продуктивності моделі.

Word2vec — це алгоритм для побудови векторного представлення слів, також відомий як вбудовування слів. Вектор для кожного слова є семантичним описом того, як це слово використовується в контексті, тому два слова, які однаково використовуються в тексті, отримують подібні векторні репрезентації. Щоб знайти слова, які мають подібну семантику, при відображенні слів у векторний простір застосовується векторна математика.

У векторному просторі синоніми слів мають близькі репрезентації, що дозволяє, опрацювавши текст через наперед навчений простір, отримати чітке представлення заданого тексту у ньому [3]. Ілюстрація до роботи простору ембеддингів зображена на рис. 2.

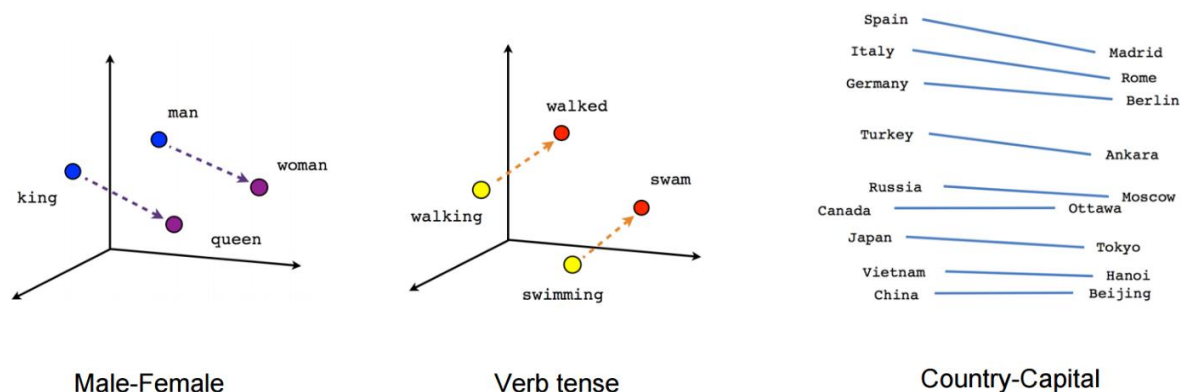


Рис. 2. Ілюстрація розподілу слів у n-вимірному просторі ембеддингів

Метод word2vec вирішує більшість проблем, з якими стикаються стандартні методи на основі алгоритму «one-hot encoding». До недоліків методу слід віднести складність побудови n-вимірного простору, адже для цього необхідно навчати нейронну мережу на величезних базах текстів. Існують готові рішення, як, наприклад, від платформи Natural Language Toolkit (NLTK), де наперед навчений n-вимірний простір можна використовувати майже без змін. Сучасна бібліотека Gensim для Python забезпечує гарну реалізацію методу word2vec, яка ідеально працює з вхідними даними у форматах NLTK.

Порівняння ембедингів

Однією із ключових проблем при застосуванні методу word2vec є коректний вибір функції ρ , що характеризує відстань між ембеддингами. У переважній більшості випадків звичайна евклідова відстань є хорошим вибором, але існує велика кількість й інших метрик [4], найпопулярніші з яких зображені на рис. 3.

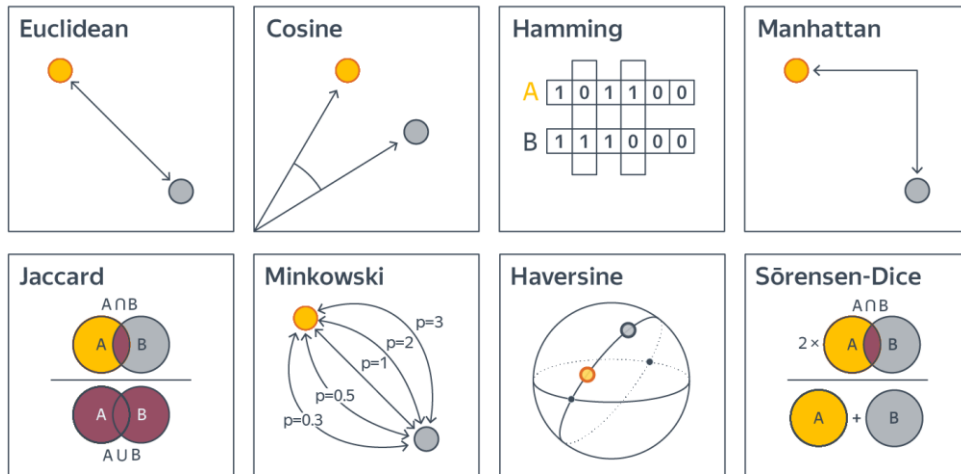


Рис. 3. Найпопулярніші метрики для порівняння ембедингів

Оскільки задача полягає в порівнянні ембедингів, які отримані з текстів, і відображень ембедингів в n -вимірному просторі, то для цієї задачі найкращою буде метрика cosine-similarity, що використовує кут між двома векторами в просторі та приймає значення від 0 до 1:

$$\cos(\theta) = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (1)$$

Як показують проведені обчислення, для речень «Hi, everyone!» та «Hello, all» при використанні векторного простору ембедингів з метрикою cosine-similarity одержується схожість 0,824 замість значення 0, одержаного за методом «one-hot encoding», рис. 4.

```

sentence_a sentence_b similarity
Hi, everyone! Hello, all 0.823691

```

Рис. 4. Застосування метрики cosine-similarity для двох речень, що не мають спільних слів

Імплементация методу word2vec при побудові контентно-орієнтованої рекомендаційної системи

З використанням word2vec простору від NLTK розроблено систему, яка дозволяє більш якісно знаходити зв'язок між текстами. Система легко інтегрується у будь-який сервіс. Ембеддинги від NLTK здатні чітко співставляти речення, навіть, якщо вони написані з вживанням синонімів.

Завдяки отриманню більш вдалим ембеддингів, що враховують не тільки конкретні слова, але і суть речення, можна отримати більше корисної інформації для цільових дій. Наприклад, побудована рекомендаційна система для новинного сайту на базі ембеддингів, отриманих зазначеним чином, здатна збільшити конверсію кліків на рекомендації більш, ніж на 25%, в порівнянні з рекомендаційним алгоритмом, що базується на самих популярних статтях.

Висновки

Зроблено огляд методу word2vec створення ембеддингів у n-вимірному векторному просторі. Метод був застосований в системі надання динамічних рекомендацій користувачам новинного порталу. Побудована контентно-орієнтована рекомендаційна система збільшила кількість конверсійних переходів користувачів на 25%.

Література

1. *Wickham H., Francois R.* dplyr: A Grammar of Data Manipulation, 2016. Available from: <https://cran.r-project.org/web/packages/dplyr/index.html>
2. *Mikolov T., Chen K., Corrado G., Dean J.* Efficient Estimation of Word Representations in Vector Space, 2013. Available from: <https://arxiv.org/abs/1301.3781>
3. *Schwenk H.* Continuous space language models // Computer Speech and Language, Vol. 21, 2007. – P. 492–518.
4. *Shaker K., Zahoor M., Biadaa M.* Similarity metrics for classification: A Review, 2020, 2nd International Scientific Conference of Al-Ayen University (ISCAU-2020), doi:10.1088/1757-899X/928/3/032052

УДК 519.688

Канд.фіз.-мат.н. Вовк Л. Б., студентка Нікітіна М. А.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

АЛГОРИТМ ПОШУКУ ПОДІБНИХ ЗОБРАЖЕНЬ НА ОСНОВІ ГЛИБОКОГО АВТОЕНКОДЕРА

Abstract

Liliia Vovk, associate professor; Maryna Nikitina, student;

Algorithm for Similar Images Retrieval Based on Autoencoder

This paper concerns the task of similar images retrieval basing on autoencoder. A new algorithm for similar images retrieval is proposed and discussed. The analysis of the algorithm efficiency is presented. The ways for further research and upgrades are proposed as well.

Вступ

Функції пошуку подібних зображень необхідні в багатьох програмах комп'ютерного зору, таких як зіставлення зображень, вирівнювання зображення, розпізнавання об'єктів та пошук. Останнім часом було запропоновано багато детекторів і дескрипторів зображень. Різні підходи до вирішення проблеми потребують різних функцій пошуку подібних або споріднених зображень.

Пошук подібних зображень є основною сферою досліджень комп'ютерного зору та аналізу зображень. Це передбачає пошук двох схожих зображень або частин зображення за допомогою різних функцій. Основними методами пошуку подібних зображень є методи машинного навчання, серед яких особливою точністю результатів виділяються глибокі нейронні мережі.

У статті запропоновано рішення для пошуку подібних зображень на основі глибокого автоенкодера, що дозволяє суттєво покращити точність роботи системи.

Постановка задачі

Метою дослідження є створення системи пошуку зображень на основі глибокого автоенкодера для швидкого знаходження подібних або споріднених зображень.

Предметом дослідження є математичне та програмне забезпечення системи пошуку подібних зображень на основі автоенкодера.

Задачею є розробка спеціалізованої системи для пошуку подібних або споріднених зображень на основі глибокого автоенкодера.

Основним внеском цієї роботи є покращення алгоритму для знаходження подібних зображень шляхом удосконалення архітектури глибокої нейронної мережі.

Термінологія

Content-Based Image Retrieval (CBIR) – техніка для відновлення зображень на основі автоматично-виявлених характеристик, таких як колір, текстура, чи форма [1].

Structural Similarity Index (SSIM) – це метод для виявлення різниці двох зображень, використовується як метрика для вимірювання подібності між двома зображеннями, яка заснована на деградації структурної інформації [2].

Опис алгоритму

Автоенкодер – це специфічна форма штучної нейронної мережі, яку використовують для CBIR систем. Метою автоенкодера є перетворення та представлення вхідних даних в стисненому або розрідженому представленнях. Точніше, автоенкодер є неконтрольованим методом навчання, який встановлює цільові значення рівними вхідним значенням, тобто $y_i = x^{(i)}$ [3].

Запропонований автокодер містить один вхідний шар, один або декілька прихованих шарів і один вихідний шар, який має точно таку саму розмірність, як і вхідний шар, як показано на Рис. 1. Різні шари мережі застосовують серію перетворень (нелінійних у більшості випадків) для перетворення даних. Приховані шари працюють з різними представленнями вхідних даних.

Функціонально автокодер містить два компоненти під час процесу навчання: кодер і декодер. Кодер використовується для кодування вхідних даних до бажаного стисненого (або розрідженого) представлення із втратами шляхом застосування перетворень h^j (рівень j), тоді як декодер декодує це стиснене подання наближаючи до вхідних даних $\hat{x}^{(i)}$, з $\hat{x}^{(i)}$ якнайближче до $x^{(i)}$. Зазвичай (нелінійне) перетворення h є сигмоподібною

функцією, тобто логістична функція $h(z) = \frac{1}{1 - \exp(-z)}$, де W — матриця вагів, а b — вектор зсуву. Матриця ваг W' в декодері представляє собою транспоновану форму матриці W в кодері, тобто $W' = W^T$

На етапі навчання автокодера параметри W і b оптимізовані таким чином, що середня помилка відновлення зведена до мінімуму.

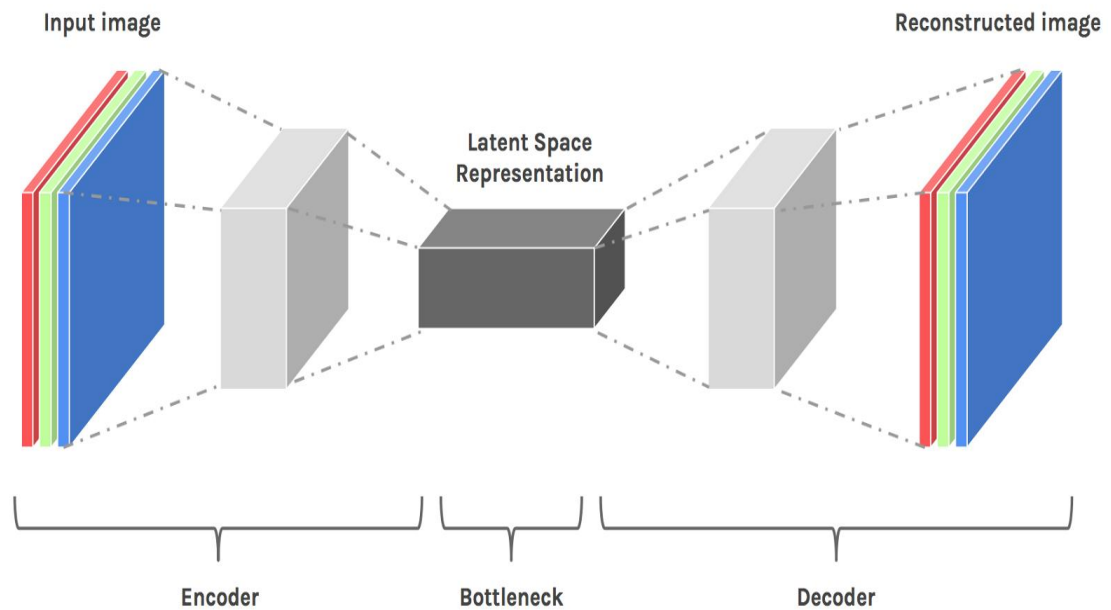


Рис. 1. Структура простого автоенкодера [4]

Оцінка роботи алгоритму

Базуючись на [5],[6], для оцінки якості роботи алгоритму було обрано в якості метрики SSIM. В запропонованій системі при стисканні зображень декодером на 75% відсотків отримано середнє значення SSIM як індексу структурної схожості в 57%. Тобто, при стисканні картинки у 4 рази втрачається тільки 43% інформації.

Висновки

1. Запропоновано алгоритм для реалізації системи пошуку споріднених зображень на основі глибокого автоенкодера.
2. В результаті аналізу та оцінки роботи запропонованого алгоритму для системи було встановлено, що при стисканні зображення у 4 рази індекс структурної схожості дорівнює 57%.
3. Перспективи подальших досліджень спрямовані на покращення результатів роботи системи, додавання більшої кількості метрик, для оцінки роботи системи пошуку подібних зображень.

Література

1. *Eakins J. P., Graham M. E.* Content-based image retrieval, a report to the JISC Technology Applications programme. – 1999.
2. *Wang Z. et al.* Image quality assessment: from error visibility to structural similarity //IEEE transactions on image processing. – 2004. – Т. 13. – №. 4. – С. 600-612.
3. *Masci J. et al.* Stacked convolutional auto-encoders for hierarchical feature extraction //International conference on artificial neural networks. – Springer, Berlin, Heidelberg, 2011. – С. 52-59.
4. *Shahriyar S. A. et al.* Speech enhancement using convolutional denoising autoencoder //2019 International Conference on Electrical, Computer and Communication Engineering (ECCE). – IEEE, 2019. – С. 1-5.
5. *Hore A., Ziou D.* Image quality metrics: PSNR vs. SSIM //2010 20th international conference on pattern recognition. – IEEE, 2010. – С. 2366-2369.
6. *Sara U., Akter M., Uddin M. S.* Image quality assessment through FSIM, SSIM, MSE and PSNR—a comparative study //Journal of Computer and Communications. – 2019. – Т. 7. – №. 3. – С. 8-18.

УДК 681.301

канд. фіз.-мат. наук Вовк Л. Б., студент Пастернак М.О.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

СИСТЕМА ДЛЯ ПОШУКУ РІВНОВАГИ ДВОХ ПІДПРИЄМСТВ З УРАХУВАННЯМ ВИПАДКОВОГО ПОПИТУ ЗА НЕШЕМ ТА ШТАКЕЛЬБЕРГОМ

Abstract

Liliia Vovk, Associate Professor, PhD; Maryna Pasternak, student
System for Finding an Equilibrium of Two Enterprises Taking into Account a Random Demand after Nash and Stackelberg

The task is to design and implement a system for finding a balance between two enterprises. In order to achieve the set goals, the methods of game theory and decision-making are studied – the search for equilibrium according to Nash and Stackelberg. In the final result, the model must find a balance for the benefit of enterprises.

Вступ

Останнім часом відбувається поступова еволюція традиційних ринкових інститутів конкурентної боротьби у партнерські чи коопераційні механізми досягнення умовної рівноваги. При цьому терміном «рівновага» можна називати багато різних станів економічних систем – від відсутності змін до стабільних прогнозованих динамічних станів.

В даній роботі будуть використовуватись елементи теорії ігор. А саме, ми розглянемо рівновагу за Нешем та Штакельбергом та порівняємо ці два алгоритми.

Постановка задачі

Задача полягає в проектуванні та реалізації системи пошуку рівноваги між двома підприємствами. Для досягнення поставлених цілей досліджуються методи теорії ігор та прийняття рішень – пошук рівноваги за Нешем та Штакельбергом.

В кінцевому результаті модель має знаходити рівновагу для двох підприємств.

Термінологія

Теорія ігор – галузь прикладної математики, яка надає інструменти для аналізу ситуацій, у яких сторони – гравці, приймають взаємозалежні рішення. Ця взаємозалежність змушує кожного гравця враховувати можливі рішення або стратегії іншого гравця при формулюванні власної стратегії. Вирішення гри описує оптимальні рішення гравців, які можуть мати подібні, протилежні або змішані інтереси, а також результати, які можуть бути результатом цих рішень.

Рівновага Неша — це концепція теорії ігор, яка визначає оптимальне рішення в некооперативній грі, в якій гравці не змінюють свою початкову стратегію. Ніякі зміни в стратегії гравця не покращують його ситуацію. Гра може включати декілька рівноваг Неша або жодної.

Модель Штакельберга — це послідовна модель, що містить дві фірми –одного лідера та одного послідовника. Вони обидві мають досконалу інформацію про функцію ринкового попиту. Фірма-лідер рухається першою та приймає рішення з урахуванням реакції послідовника. Послідовник раціонально встановлює свій власний рівень виходу відповідно до будь-якої величини, встановленої лідером, з надією, що лідер не буде протидіяти. Подібним чином лідер може очікувати, що послідовник буде відповідати його найкращій стратегії.

Опис алгоритму

Ми розглядаємо виробника, який виробляє один продукт за вартістю одиниці $c \geq 0$. Він розповсюджує його через свій канал продажів, який повністю належить йому, за ціною p_m через незалежного роздрібного продавця і канал ($i = 1, \dots, n$), який продає за оптовою ціною w_i .

Кожен канал має обмежену місцеву монополію в результаті певної форми диференціації (з точки зору розташування, назви бренду, цін, обслуговування та наданої підтримки тощо). Загальний попит ринку розподіляється між каналами. Клієнти посилаються на кожен канал як на свій перший вибір для задоволення своїх потреб попиту. Це називається первинним попитом. Позначимо $\bar{D}_j(j = M, R_1, \dots, R_n)$ основні вимоги каналу, де M вказує на канал виробника і R_i вказує роздрібного продавця і канал. Для аналітичного розгляду задачі ми припускаємо, що вимоги в різні періоди є незалежними та однаково розподіленими.

Рішення про запаси в ланцюжку поставок приймаються періодично та інвентаризація контролюється відповідно до політики базових запасів. Послідовність подій, які пояснюють динаміку моделі, наступна.

На початку кожного періоду спостерігається поточна інвентаризація та замовлення відповідного поповнення. Ці замовлення надходять відразу. Ми робимо припущення щодо потужності постачання, а також – що час на виконання є нульовим. В цьому випадку вимоги каналу виконано, незадоволений попит втрачається, а надлишкові запаси переносяться на наступний період. S_j і h_j вказують базовий рівень запасів і вартість утримання каналу j відповідно.

У цій моделі передбачається, що терміни поповнення дорівнюють нулю та виробник має нескінченну потужність постачання у верхньому ешелоні, що означає, що він може миттєво виконувати замовлення виробників і роздрібних продавців.

Виробник не має жодних запасів на цьому рівні; запаси зберігаються виробником і роздрібними продавцями лише для задоволення потреб кінцевого споживача.

Загальний складений попит для кожного каналу, позначеного \bar{D}_j , включає перший вибір покупців, а також будь-які витрати на покупців, які замінюють продукт. Це задається формулою:

$$D_j = \bar{D}_j + \sum_k \alpha_{kj} (\bar{D}_k - S_k) (j, k = M, R_1, \dots, R_n, j \neq k). \quad (1)$$

Тому, незважаючи на незалежність першого вибору, фактичний попит, з яким стикається кожен канал, залежить від відповідного першого попиту на вибір та рівень запасів інших каналів.

Тепер з формули (1) ми отримуємо D_m . Завдяки змінним у формулі, можна зрозуміти, що розподіл D_m може бути наслідком розподілів \check{D}_m або \check{D}_r . Вважається, що розподіл отримують з лінійної комбінації двох вищезгаданих змінних. Аналітичне рішення для знаходження розподілу є складним, тому ми вирішуємо цю задачу чисельним методом.

1. Обрати S (складський інвентар) випадковим чином із потрібного діапазону.

$$\check{D}_m$$

2. \check{D}_r .

Щоб розв'язати рівняння (1), ми створюємо випадкові числа для S , \check{D}_m , і використовуємо ці числа, щоб знайти D_m . Ця операція повторюється 1000000 разів.

Щоб знайти рівноваги Неша і Штакельберга, ми розв'язуємо рівняння

$$p(D_m < S_m) = \frac{p_m - c - \sum_{i=1}^n \infty_{mri} (w_{ri} - c) p(D_{ri} \leq S_{ri}, D_m > S_m)}{p_m - c + h_m},$$

$$p(D_{mj} < S_{mj}) = \frac{p_{ri} - w_{ri}}{p_{ri} - w_{ri} - h_{ri}} \quad (i = 1, \dots, n),$$

$$p(D_m < S_m), \quad p(D_m < S_m), \quad p(D_{ri} \leq S_{ri}, D_m > S_m).$$

Спочатку знайдемо невідомі значення. Для цього ми здійснимо наступні кроки:

Кроки алгоритму знаходження рівноваги Неша:

1. Обрати S випадковим чином із потрібного діапазону.

$$\tilde{D}_m$$

2. Обрати випадкове значення для \tilde{D}_r .

3. Знайти

$$p(D_m < S_m), \quad p(D_m < S_m), \quad p(D_{ri} \leq S_{ri}, D_m > S_m).$$

4. Повторювати вищезазначені кроки, доки дві сторони рівняння не стануть рівними.

Рівновагу Штакельберга складніше обчислити порівняно зі стратегією рівноваги Неша через вимогу, щоб лідер вибирав стратегію на основі реакції послідовника. Це накладає складне обмеження для оптимізації лідера. Стратегія рівноваги Штакельберга приваблива, коли один гравець має достатньо інформації, щоб бути лідером. Однак потенційний лідер може виявити, що йому краще бути послідовником, і в цьому випадку він може розкрити достатньо інформації, щоб інший гравець став потенційним лідером.

Кроки алгоритму знаходження рівноваги Штакельберга:

1. Знайти згори алгоритм рівноваги Неша.

2. Обрати випадкове значення для $\tilde{D}_m / \tilde{D}_r$.

3. Знайти $p(D_m < S_m), \quad p(D_m < S_m), \quad p(D_{ri} \leq S_{ri}, D_m > S_m)$.

4. Повторювати вищезазначені кроки, доки дві сторони рівняння не стануть рівними.

Висновки

Ми розглянули модель, яка складається з одного виробника та довільної кількості продавців.

Було розглянуто рівновагу за Нешем. Також ми обговорили гру Штакельберга, коли виробник відіграє роль лідера та спочатку вибирає базовий рівень запасів. Наведено алгоритми знаходження рівноваги Нешата Штакельберга.

Література

1. *Seger C.* An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing. – 2018.
2. *Deegalla S., Bostrom H.* Reducing high-dimensional data by principal component analysis vs. random projection for nearest neighbor classification // 2006 5th International Conference on Machine Learning and Applications (ICMLA'06). – IEEE, 2006. – P. 245-250.
3. *Luaces O. et al.* Binary relevance efficacy for multilabel classification // Progress in Artificial Intelligence. – 2012. – V. 1. – №. 4. – P. 303-313.
4. *Read J. et al.* Classifier chains for multi-label classification // Machine learning. – 2011. – V. 85. – №. 3. – P. 333-359.
5. *Tenenboim-Chekina L., Rokach L., Shapira B.* Identification of label dependencies for multi-label classification // Working Notes of the Second International Workshop on Learning from Multi-Label Data. – 2010. – P. 53-60.
6. *Zhang M. L., Zhou Z. H.* A k-nearest neighbor based algorithm for multi-label classification // 2005 IEEE international conference on granular computing. – IEEE, 2005. – V. 2. – P. 718-721.
7. *S. J. Rubio.* On Coincidence of Feedback Nash Equilibria and Stackelberg Equilibria in Economic Applications of Differential Games. // Journal of Optimization Theory and Applications: Vol. 128, No. 1, pp. 203–221, January 2006

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

ЗАСТОВУВАННЯ КОНФОРМНИХ ПРЕДИКТОРІВ І СТЕПЕРЕВИХ МАРТИНГАЛІВ ДЛЯ ВИЯВЛЕННЯ ПІДОЗРЛИХ МАТЧІВ ФУТБОЛЬНИХ ТУРНІРІВ

Abstract

Ivan Zhuk, PhD student

Approach based on conformal predictors and power martingales for detection of suspicious matches in football tournaments

This paper concerns the task of suspicious match detection. The modern approaches for detection of suspicious matches is studied and discussed. The in-place detection algorithm based on conformal predictors and power martingale is proposed. The quality analysis of algorithm based on recall and precision metrics is fulfilled. The ways for further research are proposed as well.

Вступ

Однією з популярних для пошуку аномалій у даних є сфера спортивних змагань. Однією із задач у цій сфері є пошук матчів з фіксованим результатом (договірних) [1], який може розглядатись як аномалія. На практиці пошук аномалій у цій сфері, принаймні, у сфері футболу зводиться до аналізу ставок або дій учасників матчу протягом всієї гри [1, 2]. Виникає необхідність розробки методу, який дозволяв би здійснити пошук підозрливих матчів на основі відкритої інформації про футбольні матчі.

В монографії [3] розроблено математичний апарат конформних предикторів і степеневих мартингалів для розв'язання задач класифікації і регресії, зокрема, для класифікації зображень і лінійної регресії. Переваги даного математичного апарату полягають у поєднанні процесу навчання і прогнозування у одну стадію і, як наслідок, у використанні результатів класифікації (або регресії) попередніх об'єктів для прогнозування результату для поточного об'єкта. Також даний математичний апарат дозволяє шукати аномалії серед об'єктів. Зокрема, в [4] показано, як саме цей математичний апарат можна практично використати для покращення результатів класифікації методами опорних векторів і найближчих сусідів, і для виявлення змін у потоках даних.

В роботі запропоновано використати апарат конформних предикторів і степеневих мартингалів для пошуку підозрілих матчів футбольних турнірів.

Основна ідея методу полягає у виявленні матчів завдяки зміні значення відповідного степеневого мартингалу. Значення мартингалу для матчу формується на основі результатів конформного предиктору для поточного і попередніх матчів. Конформний предиктор визначає ймовірність того, на скільки поточний матч відповідає всій групі матчів на основі даних про успішність команд-учасниць і результатів як поточного матчу, так і попередніх матчів турніру.

Постановка задачі

Задача полягає в розробці методу виявлення підозрілих матчів у футбольних турнірах на основі використання даних про команди-учасниці і результати матчів.

Термінологія

Конформні предиктори – клас методів машинного навчання, які прогнозують належність об'єкту до певного класу на основі міри відмінності (конформності) поточного об'єкта від попередніх об'єктів [3].

Степеневий мартингал – математична величина, розроблена для перевірки гіпотези обмінюваності (exchangeability) випадкових подій [3].

Обмінюваність: ймовірнісний розподіл F випадкових подій $z_1, z_2, \dots, z_n \in Z$, якщо для будь-якої перестановки π подій z_1, z_2, \dots, z_n ймовірність $p(z_1, z_2, \dots, z_n) = p(\pi(z_1), \pi(z_2), \dots, \pi(z_n))$ [3]

Опис вхідних даних

Одиницею вхідних даних для методу є спостереження z_k , що описує матч футбольного сезону, k є порядковим номером матчу у сезоні. Спостереження z_k є набором значень $z_k = (i_k, j_k, \alpha_k, \beta_k, T_k)$, де i_k і α_k є рангом і результатом домашньої команди матчу, а j_k і β_k є рангом і результатом виїзної команди матчу, T_k є датою проведення матчу.

Ранг i_k матчу може бути визначений на основі успішності команд (наприклад, її місця в загальному заліку) в цьому або попередніх сезонах. Прикладом обчислення успішності є формування таблиці результатів команд сезону, де успішність $s(t)$ команди t обчислюється як $s(t) = 3w(t) + d(t)$, де $w(t)$ та $d(t)$ є відповідно кількістю перемог та ігор у нічию для команди t .

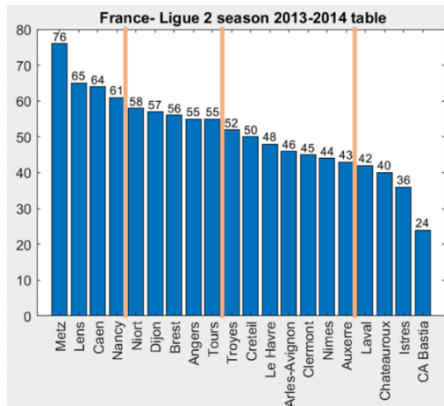


Рис 1. Успішність команд сезону 2013-2014 II Ліги Франції

Будемо далі розглядати приклад сезону 2013-2014 II Ліги Франції. Успішність команд визначається на основі результатів команд в поточному сезоні (рис.1). Після формування таблиці успішності, команди розбиваються на групи за близькістю значень їх успішностей, наприклад шляхом проведення одновимірної кластеризації. Таким чином, у цьому сезоні було

сформовано 4 кластери (класи) команд (рис. 1): команди №№1-4 належать до кластеру №1, команди №№5-9 належать до кластеру №2, команди №№10-16 належать до кластеру №3, команди №№17-20 належать до кластеру №4. Потім на основі сформованих груп формуються класи матчів. Для сезону 2013-2014 II Ліги Франції було сформовано множину з 16 класів матчів $C = \{(i, j) \mid i, j \in \{1, 2, 3, 4\}\}$. На такій множині задана функція $class(z_k)$, яка визначає клас матчу і використовується в подальшому алгоритмі.

Після етапу ранжування, до кожної групи матчів застосовується алгоритм, описаний в наступному розділі.

Опис алгоритму

Алгоритм розроблено на основі загального методу, запропонованого в [3], шляхом його адаптації для задачі, розглянутої вище.

Для кожного нового матчу z_k :

1) обчислюється міра відмінності a_k , яка є першим етапом при розрахунку конформного предиктора:

$$a_k = 1,5^{1 - \text{sgn}(\text{avg}_k(i, j) * (\alpha_k - \beta_k))} \left| (\alpha_k - \beta_k) - \text{avg}_k(i, j) \right|$$

де $\text{avg}_k(i, j) = \text{mean}_{k \neq l, \text{class}(z_k(i, j)) = \text{class}(z_l(s, t))} \{\alpha_l - \beta_l\}$, k є номером

поточного матчу в хронологічному порядку, l є номером іншого матчу, який відноситься до того класу, що й поточний матч, i, j є відповідно номером домашньої і виїзної команди поточного матчу, s, t є відповідно номером домашньої і виїзної команди матчу під номером l .

2) використовуючи міри відмінності поточного k матчу і попередніх матчів групи, обчислюється ймовірність відмінності матчу p_k , яка є результатом конформного предиктора:

$$p_k = p(z_1, z_2, \dots, z_k) = \frac{\#\{i : a_i \geq a_k, 1 \leq i \leq k\}}{k}$$

3) на основі наявного на поточний момент ряду ймовірностей, обчислюється інтегральний мартингал M_k :

$$M_k = \int_0^1 M_k^{(\varepsilon)} d\varepsilon$$

де $M_k^{(\varepsilon)}$ є степеневим мартингалом і визначається як $M_k^{(\varepsilon)} = \varepsilon^k \prod_{i=1}^k p_i^{\varepsilon-1}$.

4) Множина підозрілих S матчів формується за наступним правилом:

$$S = \{z_k \mid M_k > M_{k-1}\}$$

Результати

Продемонструємо роботу вказаного методу на матчах класу (1, 4) (табл. 1). Підозрілі матчі в таблиці 1 позначено сірим кольором. В цю групу входять матчі, у яким домашня команда характеризується одним з найнижчих значень успішності, а виїзна команда є однією з найуспішніших в цьому сезоні.

Таблиця 1
Матчі класу (1,4) сезону 2013-2014 II Ліги Франції

№	Домашня команда	Виїзна команда	Результат	Дата
1	Metz	Laval	1:0	02-Aug-2013
2	Lens	CA Bastia	1:0	04-Aug-2013
3	Metz	Chateauroux	1:0	04-Oct-2013
4	Caen	Istres	4:0	08-Nov-2013
5	Nancy	Chateauroux	2:0	08-Nov-2013
6	Lens	Chateauroux	2:0	23-Nov-2013
7	Lens	Istres	1:2	19-Dec-2013
8	Nancy	Laval	2:1	20-Dec-2013
9	Caen	Laval	2:1	17-Jan-2014
10	Nancy	CA Bastia	0:1	17-Jan-2014
11	Lens	Laval	0:0	01-Feb-2014
12	Caen	Chateauroux	1:1	21-Mar-2014
13	Metz	Istres	2:1	07-Apr-2014
14	Metz	CA Bastia	1:0	18-Apr-2014
15	Caen	CA Bastia	6:1	02-May-2014
16	Nancy	Istres	3:1	02-May-2014

Середній результат по групі $avg_{k(i,j)}$ дорівнює 1,25. Отже, очікуваним результатом матчу є виграш домашньої команди на рівні 1 або 2 м'ячів. Всі матчі, результат яких відхиляється від очікуваного, розглядаються як підозрілі.

Характеристика M_k , обчислена для групи матчів

(рис. 2) в подальшому за принципом, описаним формулою з п.4 опису алгоритму: точки, у яких значення мартингалу зростає в порівнянні з

попередніми, відповідають матчам, що є підозрілими. За таким принципом, для групи матчів (1,4) алгоритм за метрикою повноти (recall) спрацював на 100%: виявлено всі очікувані підозрілі матчі. За метрикою точності алгоритм спрацював на 67%: всі очікувані матчі виявлено, але виявлено і такі матчі, які не вважаються підозрілими.

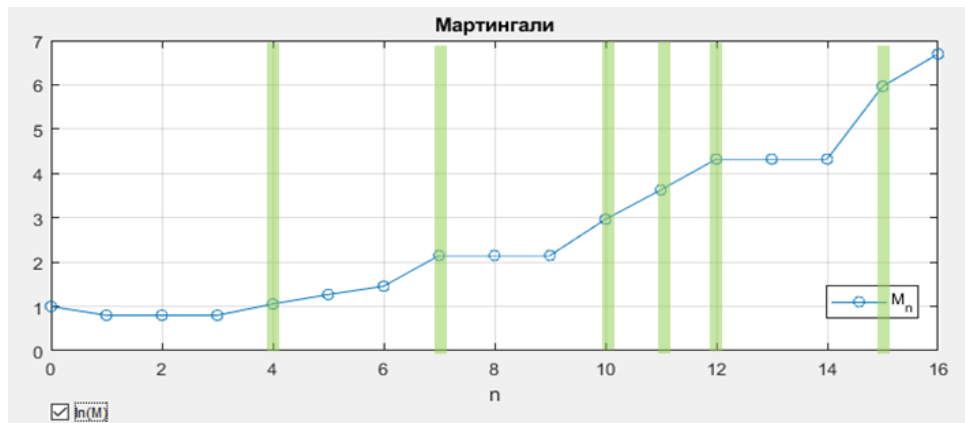


Рис 2. Характеристика $\ln(M_k)$ для матчів класу (1,4)

Висновки

Запропонований алгоритм дає можливість визначати підозрілі матчі за відкритими даними: історією сезону футбольного турніру. Підозрілість матчу визначається за рахунок успішності команд-учасниць і результату матчу. Завдяки врахування хронології матчів, алгоритм може «навчатись» і з плином історії сезону покращувати точність виявлення підозрілих матчів. Якість роботи алгоритм за метрикою повноти досягає 100%.

В подальшому, для збільшення точності алгоритму буде доцільно розглянути інші правила прийняття рішень про визначення підозрілого матчу. Також для цього буде доцільно розглянути інші типи ймовірності та міри відмінності, які доступні у розробленому математичному апараті конформних предикторів.

Література

1. V. Constandt and E. Manoli, *Understanding match-fixing in sport: Theory and practice*. 2022.
2. Chertov, Oleg, Ivan Zhuk, and Anatolii Serdyuk. 'Search of the Deviation from the Natural Process Using Stepanets Approach for Classification of Functions', *2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, 2:720–24, 2021. doi: 10.1109/IDAACS53288.2021.9660997.

3. V. Vovk, *Algorithmic Learning in a Random World*. New York: Springer-Verlag, 2005. doi: 10.1007/b106715.
4. S.-S. Ho and H. Wechsler, 'A Martingale Framework for Detecting Changes in Data Streams by Testing Exchangeability', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, pp. 2113–2127, Dec. 2010, doi: 10.1109/TPAMI.2010.48.

УДК 519.688

Старший викладач Ліскін В.О., студент Кошолепенко І.О.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ДІАГНОСТИКИ ОКА ЗА ЗОБРАЖЕННЯМ ОЧНОГО ДНА

Abstract

Viacheslav Liskin, senior lecturer; Illia Kosholapenko, student
System of eye disease detection on images

This paper concerns the task of eye disease detection on images. Neural networks of image classification are studied and discussed. Neural model of VGG19 is proposed. The comparative analysis of different optimization methods is studied and proposed. Images dataset preprocessing is also proposed.

Вступ

Задача класифікації зображень – поширена тема для вивчення в сучасному світі. Охорона здоров'я потребує багату кількість експертів у різних галузях здоров'я, і іноді людського ресурсу може не вистачати для діагностування та лікування хвороб. Система з попередньої діагностики захворювання очей пацієнта допомогла б вирішити одну з проблем.

Ця робота пропонує систему для діагностування хвороб ока на основі зображення очного дна та за допомогою нейронних мереж.

Дана система має на меті полегшення роботи офтальмологів з діагностики захворювань ока та дає можливість зробити попередній аналіз для пацієнта.

Постановка задачі

Метою дослідження є виявлення наявності та класифікація типу захворювання (при його наявності) у пацієнта за знімками очного дна.

Предметом дослідження є математичне та програмне забезпечення системи діагностики ока за знімком очного дна.

Задача полягає в розробці програмного забезпечення для виявлення та класифікації захворювання ока по зображенню очного дна. Класифікація проводиться за рахунок розробленої нейронної мережі, для якої підбирається найефективніший оптимізатор з мінімальним використанням ресурсів та найбільшою точністю.

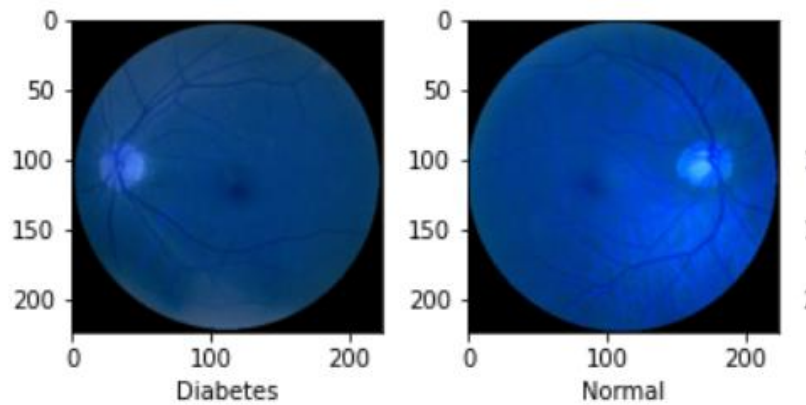


Рис. 3. Вхідні дані зображень пацієнта

Компоненти архітектури системи повинні забезпечити виконання таких основних функцій:

- групування та сортування даних зібраних в процесі роботи чи від користувача;
- видозміну вхідних даних під формат для обробки всередині системи;
- класифікація нових зображень;
- тренування, ініціювання та оптимізацію нейронної мережі класифікатору;
- візуалізація отриманих результатів.

Компоненти яка виконують вищевказані вимоги є (рис.4) : вхідні дані, класифікатор, перетворення даних, навчання моделі, візуалізація результатів.

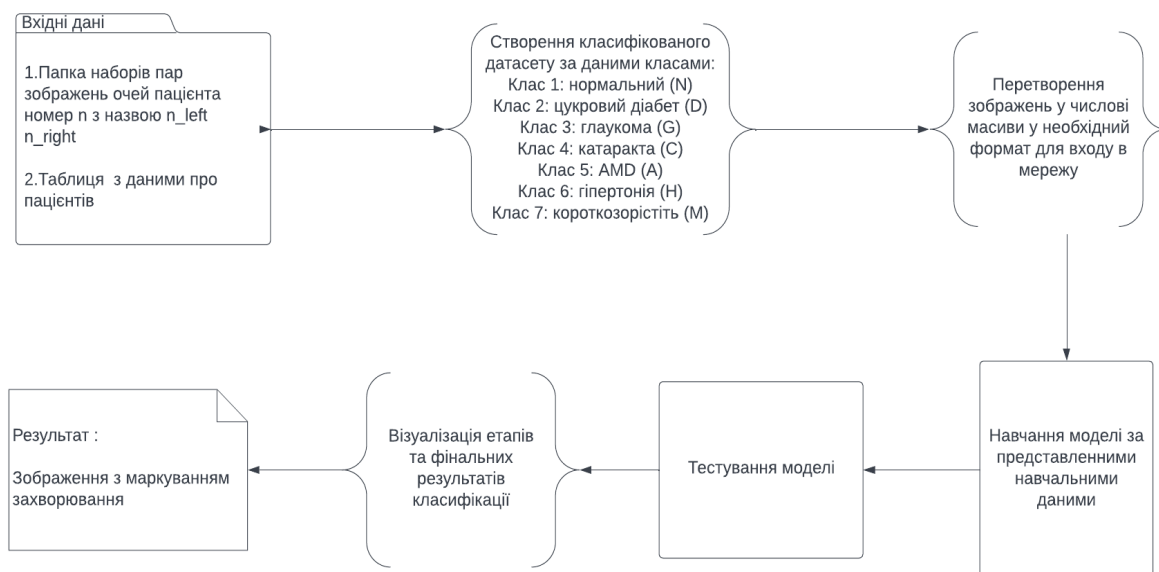


Рис. 4. Модель системи класифікації захворювань ока

Була обрана модель нейронної мережі VGG19 (рис.5), яка надає 19 шарів для успішної класифікації зображення за алгоритмом вище:

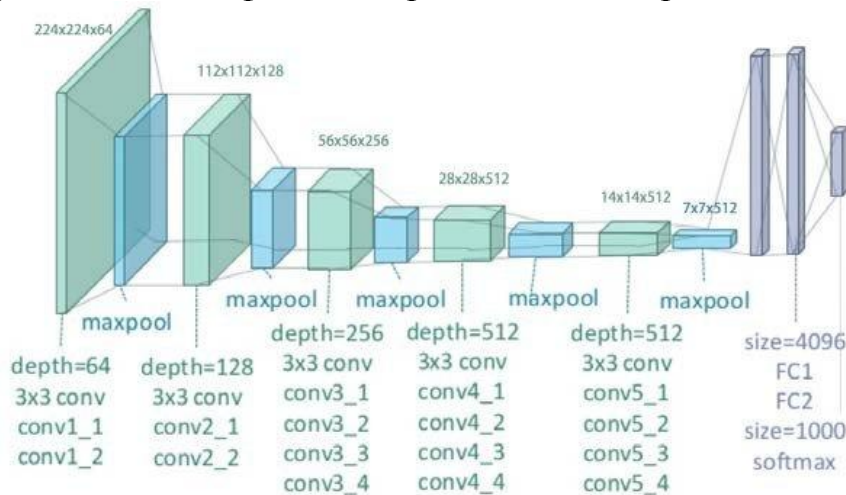


Рис. 5. Архітектура нейронної мережі VGG19

Оптимізатор – це частина алгоритму роботи нейронної мережі, який перераховує ваги нейронів кожного шару мережі. Міра навчання (learning rate) – число у діапазоні [0;1], яке вказує наскільки сильно повинна змінитись вага того чи іншого нейрону [2]. У цій роботі були розібрані наступні оптимізатори:

1. SGD (стохастичний градієнтний спуск) – стандартний градієнтний спуск
2. Adagrad (адаптивний градієнтний спуск) – градієнтний спуск зі змінною (не сталою) мірою навчання
3. RMSprop – Розширення методу Adagrad, але з використанням середнього квадратів градієнта для оновлення вагів.
4. Adam – поєднання оптимізатора RMSprop з ідеєю збереження імпульса [3].

Порівняння проводилось на невеликій кількості епох (5-10) через велику кількість вхідних даних. Результати порівняння оптимізаторів маємо наступні при невеликій різниці у часі виконання:

Таблиця 1
Результати порівняльного аналізу

Назва методу	Точність
SGD	60%
Adagrad	68%
RMSprop	76%
Adam	88%

Виходячи з отриманих результатів точності, для розробленої системи було обрано оптимізатор Adam.

Тестування системи виконувалось на окремій частині масиву вхідних даних. Візуалізовані результати виглядають наступним чином:

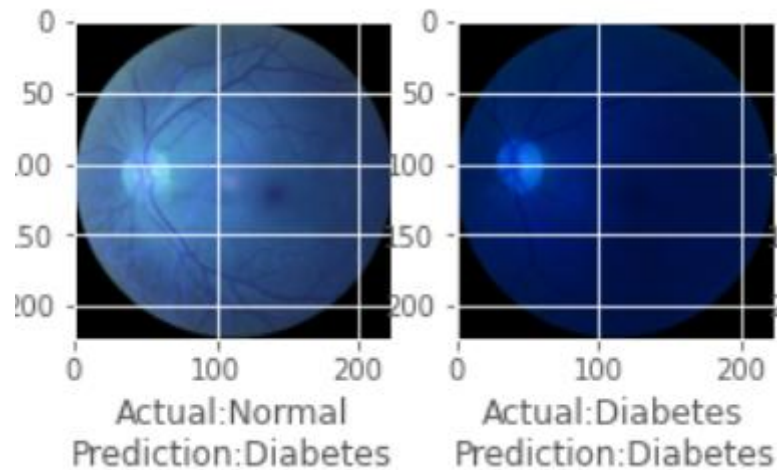


Рис. 6. Приклад результатів навчання системи

У перспективі дана система може бути покращена додавання інтерфейсу або наприклад кастомізацією ядер згорки для більшої якості класифікації.

Висновки

Було розроблено нейронні мережу для розпізнавання захворювань ка на основі моделі VGG19.

Проведено порівняльний аналіз чотирьох методів оптимізації. При порівнянні використаних методів оптимізації, найефективнішим виявився оптимізатор Адам з точністю 88%.

Дане програмне забезпечення може використовуватись для попередньої діагностики захворювань ока, та в перспективі, може бути змінено для класифікації інших захворювань.

Література

1. Liu Y. et al. Fusion that matters: convolutional fusion networks for visual recognition //Multimedia Tools and Applications. – 2018. – Т. 77. – №. 22. – С. 29407-29434.
2. O'Shea, Keiron, and Ryan Nash. "An introduction to convolutional neural networks." arXiv preprint arXiv:1511.08458 (2015)
3. Bera, Somenath, and Vimal K. Shrivastava. "Analysis of various optimizers on deep convolutional neural network model in the application of hyperspectral remote sensing image classification." International Journal of Remote Sensing 41.7 (2020): 2664-2683.

УДК 519.684

Професор Лось В.М., студент Борознюк Д.О.

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ РОЗПІЗНАВАННЯ COVID-19 ЗА РЕНТГЕНІВСЬКИМИ ЗНІМКАМИ

Abstract

Valerii M. Los, professor, Dmytro Borozniuk, student
Mathematical and software system for determination of the COVID-19 based on X-ray images

In this paper, the mathematical and software system for determination of the COVID-19 recognition system based on X-ray images developed on the basis of convolutional neural networks, which allows detecting the disease in the early stages of development, is proposed, the new architecture of four models is created, the approaches to processing input data - images are described, the system efficiency and accuracy is presented, and the ways of further research are determined.

Вступ

COVID-19 є однією з найбільш піддатливих до профілактики хвороб, вона розповсюджена не тільки на просторах України, але й по всьому світу. Сьогодні розроблено багато методів для аналізу зображень рентгенівських знімків, але не існує точних критеріїв для виділення кращого методу для вирішення задачі ідентифікації COVID-19 за допомогою рентгенів.

Тому актуальним питанням є необхідність розробки математичного забезпечення розпізнавання COVID-19 за рентгенівськими знімками на основі нейронних мереж, яке б дозволило прискорити процес виявлення пневмонії на ранніх стадіях розвитку хвороби.

Постановка задачі

Метою дослідження є класифікація та ідентифікація ознак COVID-19 за рентгенівськими знімками легень людини.

Предметом дослідження є математичне та програмне забезпечення системи розпізнавання COVID-19 за рентгенівськими знімками на основі згорткових нейронних мереж (CNN – Convolutional Neural Network), яке б дозволило прискорити процес виявлення захворювання у пацієнтів на ранніх стадіях розвитку хвороби.

Об'єктом дослідження є розробка, підбір параметрів та навчання моделі, яка зможе ідентифікувати COVID-19 за рентгенівськими знімками.

Для розв'язку поставленої задачі використовувалися методи: моделювання та проектування систем DataScience, класифікації, машинного наавчання, візуалізації даних, web-розробки. Вимоги до системи:

- система надає можливість загрузки рентгенівських знімків
- система надає користувачеві повідомлення про помилки, що можуть виникати при неправильному користуванні програмою
- виводити результат розпізнавання COVID-19
- мати високі показники ефективності розпізнавання (більше 80%)

Термінологія

COVID-19 - це потенційно тяжка гостра респіраторна інфекція, що викликається коронавірусом SARS-CoV-2. Найбільш частим ускладненням захворювання є вірусна пневмонія, здатна призводити до гострого респіраторного дистрес-синдрому та подальшої гострої дихальної недостатності.

Рентген - медичне неінвазивне дослідження, засноване на отриманні сумарного проекційного зображення анатомічних структур організму за допомогою проходження через них рентгенівських променів та реєстрації ступеня ослаблення рентгенівського випромінювання.

Згорткова нейронна мережа (convolutional neural network, CNN, ConvNet) в машинному навчанні - це клас глибоких штучних нейронних мереж прямого поширення, який успішно застосовується для аналізу візуальних зображень.

Огляд існуючих рішень

Таблиця

Порівняльний аналіз математичних підходів

	ANN штучна нейронна мережа	CNN згорткова нейронна мережа	RNN рекурентна нейронна мережа
Тип даних	Табличний	Зображення	Послідовний
Обмін параметрами	-	+	+
Фіксована довжина вхідних даних	+	+	-
Рекурентні відносини	-	-	+
Проблема зникання градієнту	+	+	+
Просторові відносини	-	+	-

В результаті порівняльного аналізу підходів вирішення поставленої задачі, було обрано метод згорткових нейронних мереж (CNN), в основному, через те, що типом даних є зображення і даний метод є більш потужнішим, аніж ANN.

Компонентна модель системи ідентифікації оціночних суджень

Архітектура системи має відповідати поставленим вимогам та мати необхідні властивості для забезпечення розпізнавання COVID-19 за рентгенівськими знімками. Задачею системи є видання результату щодо наявності хвороби.

Компоненти архітектури системи повинні забезпечити виконання таких основних функцій:

- збір, завантаження та збереження даних;
- класифікація даних;
- попередня обробка зображень;
- тренування, ініціювання та валідація нейронної мережі класифікатора;
- підбір параметрів моделей;
- надання web-інтерфейсу для користувача.

Компонентна модель системи розпізнавання COVID-19, що реалізує вказані функції, складається з таких компонентів (рис.1): збір рентгенів, обробка зображень, навчання моделей, ідентифікатор та web-застосунок.

Модулі – це сукупність засобів, правил та методів, що дозволяють виконувати обмін інформацією між компонентами системи.

- MTS (Module Text Save) — інтерфейс завантаження даних з набору даних до бази даних;
- MTP (Module Text Processing) — інтерфейс передачі даних у процес початкової обробки даних, який налаштовує кожне зображення під заданий розмір, трансформує його в список цілих чисел від 0 до 255, нормує та інше;
- MIMS (Module Identification Model Save) — інтерфейс передачі навченої моделі для використання в задачах класифікації.
- MWAR (Module Web Application Raw) — інтерфейс передачі даних, отриманих від користувача з використанням web-інтерфейсу.
- MCR (Module Classification Result) — інтерфейс передачі результату класифікації навченої моделі на основі завантаженого рентгену.

Модель системи формалізована на основі мови графічного моделювання UML, що дозволяє формалізувати відношення компонентів, їх внутрішню реалізацію засобами природної обробки мови, науки про дані, машинного навчання та web-розробки.

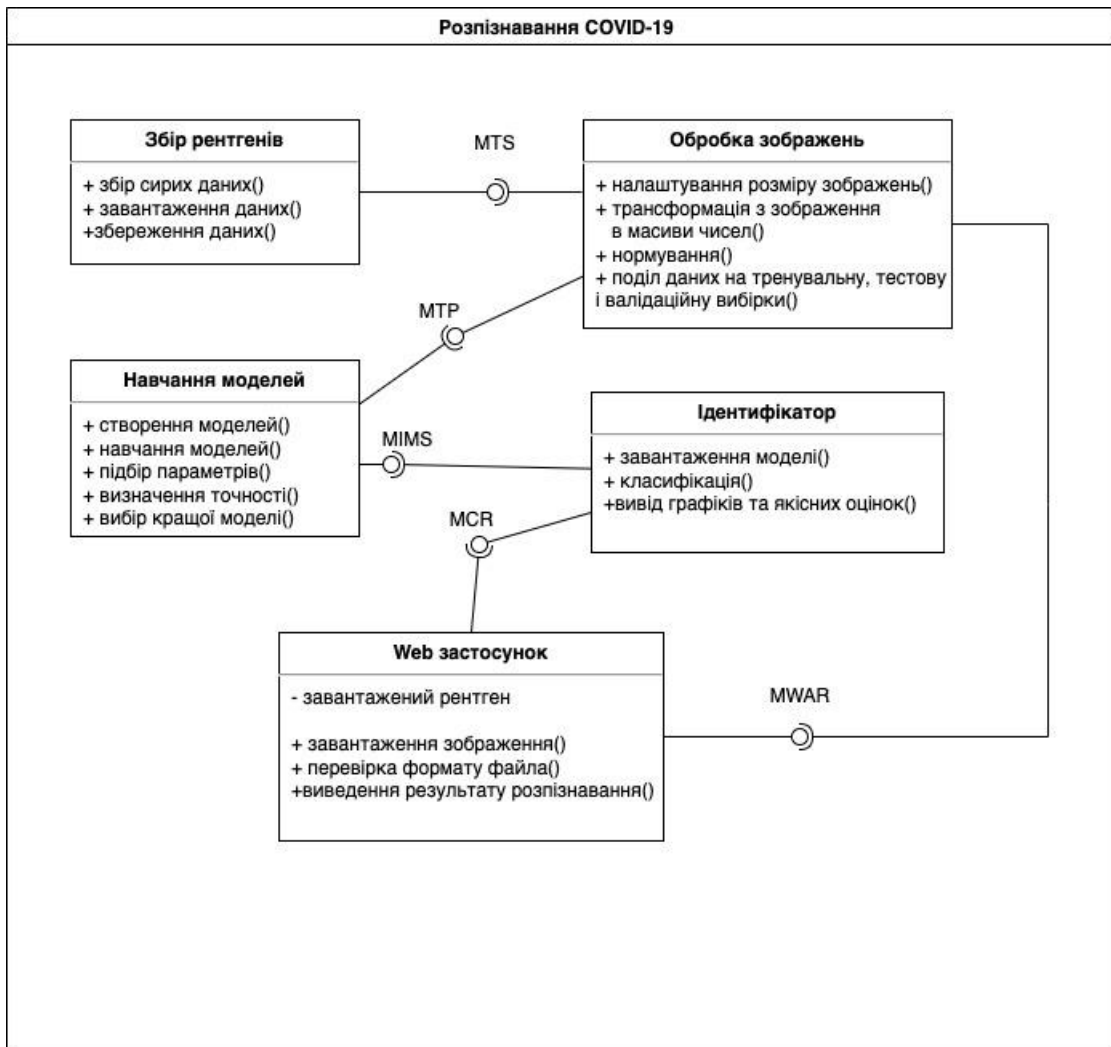


Рис 1. Модель системи розпізнавання COVID-19.
 Діаграма компонентів в нотатції UML

Обробка даних

В даній роботі було нормалізовано дані шляхом min-max нормалізації, було поділено кожний елемент на 255, так як максимальне значення пікселя 255, а мінімальне 0. Таким чином значення пікселів лежатиме в межах [0; 1].

Нормалізація min-max – один із найпоширеніших способів нормалізації даних. Для кожної ознаки мінімальне значення перетворюється на 0, а максимальне значення перетворюється на 1, тим самим кожне інше значення перетворюється на десяткове число від 0 до 1. Формула виглядає наступним чином:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Архітектури моделей

На основі обраного метода CNN, було створено 4 моделі архітектури моделей для навчання, кожна наступна з яких є складнішою за структурою, ніж попередня. Нижче наведені зображення архітектури кожної з моделей:

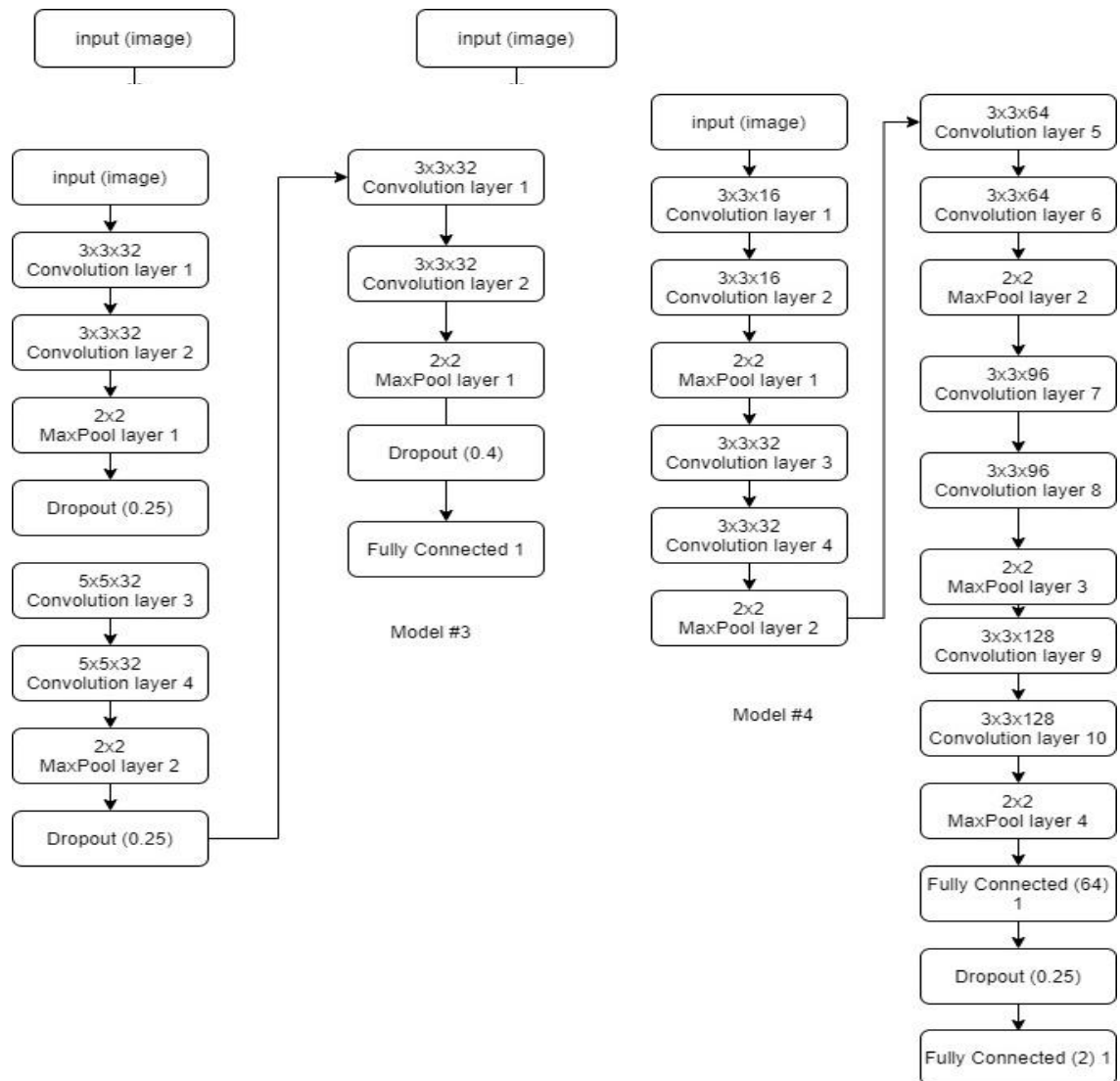


Рис. 2. Архітектури моделей для класифікації COVID-19

Для того, щоб відібрати моделі для подальшого підбору параметрів, спочатку було їх протестовано, в результаті отримали наступне:

- model№1 показала точність на тестовому наборі у 90.1 %;
- model№2 показала точність на тестовому наборі у 92.4 %;
- model№3 показала точність на тестовому наборі у 56.7 %;
- model№4 показала точність на тестовому наборі у 65.8 %.

З результатів вище можна сказати, що 3 та 4 моделі погано підходять для вирішення поставленої задачі, тому було сконцентровано увагу на перших двох, а саме для них підбиралися оптимальні параметри.

Результати

Для покращення точності моделей було вирішено підбирати наступні параметри: learningrate, optimizemethod та dropout. В результаті отримано:

Таблиця

Результати підбору параметрів моделей

Model №	Accuracy	Sensitivity	Precision	Specificity	F1 score	DOR	Conf. matrix
1	96.95%	0.967	0.973	0.9725	0.97	1036.26	$\begin{pmatrix} 1071 & 30 \\ 37 & 1060 \end{pmatrix}$
2	96.63%	0.9706	0.962	0.962	0.966	835.77	$\begin{pmatrix} 1059 & 42 \\ 32 & 1065 \end{pmatrix}$

Виходячи з точності моделей, можна зробити висновок, що model№1 є найкращою.

Висновки

В роботі:

- а) здійснено огляд літератури та пошук методів реалізації задачі класифікації зображень. Серед методів на основі виконаного порівняльного аналізу обрано метод згорткової нейронної мережі;
- б) підібрано архітектури для чотирьох моделей;
- в) проаналізовано кожну модель та виділено дві кращих. Для моделей, які залишилися, підібрано оптимальні параметри та вибрано найкращу модель, точність якої досягла 96,95 %.

В подальшому планується вдосконалення обробки даних та нейронної мережі класифікатора для досягнення ще більш точних результатів. Також планується розгорнути веб-застосунок на хмарному сховищі.

Література

1. Pneumoniadiagnosis[Електронний ресурс] – Режим доступу: <https://www.mayoclinic.org/diseases-conditions/pneumonia/diagnosis-treatment/drc-20354210>.
2. DeepLearningNetwork[Електронний ресурс] – Режим доступу: <https://thedata scientist.com/what-deep-learning-is-and-isnt/>
3. DifferencebetweenANN, CNN, RNN [Електронний ресурс] – Режим доступу: <https://www.geeksforgeeks.org/difference-between-ann-cnn-and-rnn/>

4. Convolutionalneuralnetworkswithkeras [Электронный ресурс] –Режим доступа: <https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/>.
5. Math behind Convolutional Neural Networks [Электронный ресурс] – Режим доступа: <https://towardsdatascience.com/gentle-dive-into-math-behind-convolutional-neural-networks-79a07dd44cf9>.

Д-р фіз.-мат. наук, проф. Лось В.М.,
магістрант Герасименко В.Р., ст. викл. Копичко С.М.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

МОДИФІКОВАНИЙ МЕТОД ІТЕРАЦІЙ ДЛЯ СИСТЕМ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ

Abstract

This paper deals with the iterative Picard procedure: a powerful, yet mostly unused in practice, approximate method for solving the Cauchy problem for m ordinary differential equations. One of the main problems of this method, which, in particular, hinders its use on computers, is the fact that it relies on a multiple integration of potentially non-polynomial functions, which could result in very complex quadratures, that may even not be expressed in elementary functions.

This study proposes a natural solution to this problem, which consists in modifying the Picard method itself by replacing the subintegral function by its Taylor series. The convergence of such a modification is then proved and an approximation of the error is given and possible ways for further studies are provided.

Вступ

Добре відомо, що диференціальні рівняння, а зокрема системи таких рівнянь, представляють собою надважливий клас задач, які виникають у багатьох розділах науки. Існує велика кількість чисельних та точних методів для розв'язання задач, пов'язаних з диференціальними рівняннями, в тому числі і для знаходження розв'язку задачі Коші для системи з m диференціальних рівнянь. Одним з таких методів є ітеративна процедура Пікара, проте застосування цього методу, як правило обмежується ілюстративними прикладами в рамках курсів чисельних методів, адже його застосування на практиці, і тим паче на ЕОМ, значно ускладнено необхідністю багаторазового інтегрування потенційно не поліноміальних функцій. Ця робота, власне, і присвячена вирішенню цієї проблеми.

Постановка задачі

Метою дослідження є математичне обґрунтування збіжності модифікованого методу ітерацій при заміні в ньому підінтегральних функцій їх частковими сумами Тейлора.

Предметом дослідження є метод ітерацій для систем диференціальних рівнянь, а саме його модифікація, що полягає у наближенні підінтегральних функцій їх рядами Тейлора.

Об'єктом дослідження виступає метод ітерацій для системи диференціальних рівнянь.

Задачею є, в першу чергу, математичний опис модифікації методу Пікара, а також математичне доведення збіжності такої модифікації.

Для розв'язання поставленої задачі використовувалися методи : теорії диференціальних рівнянь, функцій багатьох змінних, степеневих рядів, функціонального аналізу, зокрема метричних просторів, та чисельних методів диференціювання.

Класичний метод Пікара

Перш за все, коротко нагадаємо у чому полягає класичний метод Пікара та покажемо у чому полягає задача цього дослідження.

Нехай задана задача Коші для системи рівнянь наступного вигляду:

$$\begin{cases} y_1' = f_1(x, y_1(x) \dots y_m(x)) \\ \dots \\ y_m' = f_m(x, y_1(x) \dots y_m(x)) \end{cases} \quad (1)$$

$$y_1(x_0) = y_{0,1} \dots y_m(x_0) = y_{0,m}$$

Нехай D – область в \mathbf{R}^{m+1} , розглянемо в цій області функції $f_i(x, y_1 \dots y_m)$, $i = \overline{1, m}$ – визначені, неперервні та ліпшицеві у ній.

Нагадаємо, що умова Ліпшиця в цьому випадку формулюється таким чином.

Вектор-функція $\bar{f}(x, \bar{y}) : D \rightarrow \mathbf{R}^m$ є локально ліпшицевою, щодо \bar{y} , якщо для кожної точки $(x_0, \bar{y}_0) \in D$ знайдуться числа $r = r(x_0, \bar{y}_0) > 0, L = L(x_0, \bar{y}_0) > 0$ такі, що :

$$\left\| \bar{f}(x, \bar{y}') - \bar{f}(x, \bar{y}'') \right\| \leq L \|\bar{y}' - \bar{y}''\| \quad \forall (x, \bar{y}'), (x, \bar{y}'') \in B_r(x_0, \bar{y}_0) \subset \mathbb{R}^m$$

Тут $B_r(x_0, \bar{y}_0)$ – $(m + 1)$ -вимірний куля, радіуса r з центром в точці (x_0, \bar{y}_0) .

Відомо, що на деякому інтервалі $|x - x_0| \leq d$ існує єдиний розв'язок $y_i = \varphi_i(x), i = \overline{1, m}$ такої задачі Коші [2, с. 392].

Слідуючи [1, с. 80], розглянемо тоді простір C_m^* , елементами якого є вектори $\varphi = (\varphi_1, \dots, \varphi_m)$ з m функцій, визначених, неперервних на певному

сегменті $|x - x_0| \leq d$ і таких, що $|\varphi_i - y_{0,i}| \leq Kd$, де константу K визначено з умови $|f_i(x, y_1, \dots, y_m)| \leq K$.

Метрика у цьому просторі визначається наступним чином:

$$\rho(\varphi, \psi) = \max_{x,i} |\varphi_i - \psi_i|$$

Цей метричний простір є повним. До того ж, наступний оператор є оператором стиску в цьому просторі:

$$A(\varphi_1(x) \dots \varphi_m(x)) : \left\{ y_{0,i} + \int_{x_0}^x f_i(\tau, \varphi_1(\tau), \dots, \varphi_m(\tau)) d\tau; i = \overline{1, m} \right\} \quad (2)$$

А отже, згідно принципу стискаючих відображень, можемо побудувати ітераційну процедуру для знаходження наближеного розв'язку $\varphi_i^{(n)}(x), i = \overline{1, m}$ задачі Коші (1):

$$\varphi_i^{(j)}(x) = y_{0,i} + \int_{x_0}^x f_i(\tau, \varphi_1^{(j-1)}(\tau), \dots, \varphi_m^{(j-1)}(\tau)) d\tau$$

Тут j – це крок ітерації.

Відзначимо, що в якості $(\varphi_1^{(0)}, \dots, \varphi_m^{(0)})$ можна брати будь-які неперервні в D функції.

Видно, що, у випадку, коли підінтегральні функції $f_i(\tau, \varphi_1(\tau), \dots, \varphi_m(\tau))$ є неполіноміальними, ітераційна процедура може привести до дуже складних квадратур, які можуть навіть не виражаються у елементарних функціях. Це значно ускладнює застосування цього методу на практиці, зокрема на ЕОМ. Саме тому, було б доцільно модифікувати метод так, щоб уникнути інтегрування таких функцій, в чому і полягає наша задача.

Модифікація методу Пікара

Як вже було сказано, ми зацікавлені у тому, щоб позбутися багаторазового інтегрування складних функцій у ітераційній процедурі. Одним з рішень такої проблеми може бути наближення підінтегральних функцій поліномами. Для цього, ми можемо застосувати до $f_i(\tau, \varphi_1(\tau), \dots, \varphi_m(\tau))$ формулу Тейлора для функцій однієї змінної, розклавши їх до порядку k у точці a (для визначеності, нехай розглядається інтервал $[a, b]$) та відкинемо залишковий член. Відповідно припускаємо, що функції $f_i(\tau, \varphi_1(\tau), \dots, \varphi_m(\tau))$ мають необхідну кількість частинних похідних по всім змінним. В якості нульового наближення також беремо достатньо гладкі функції. Таким чином:

$$f_i(\tau, \varphi_1(\tau), \dots, \varphi_m(\tau)) \approx \sum_{n=0}^k \frac{f_i^{(n)}(a, \varphi_1(a), \dots, \varphi_m(a))}{n!} (\tau - a)^n$$

Відмітимо, що тут, під $f_i^{(n)}$ розуміється повна n -та похідна функції f_i , як складної функції аргументу τ , тобто наприклад

$$f_{i,\tau}'(\tau, \varphi(\tau)) = \frac{\partial f_i}{\partial \tau} + \frac{\partial f_i}{\partial \varphi_1} \frac{d\varphi_1}{d\tau} + \dots + \frac{\partial f_i}{\partial \varphi_m} \frac{d\varphi_m}{d\tau}$$

Тоді оператор (2), при заміні в ньому підінтегральних функцій їх частковими сумами ряду Тейлора (для уникнення плутанини надалі позначатимемо цей оператор $\tilde{A}\varphi(x)$.) приймає наступний вигляд:

$$\begin{aligned} \tilde{A}(\varphi_1(x), \dots, \varphi_m(x)) &: \{y_{0,i} + \int_a^x \sum_{n=0}^k \frac{f_i^{(n)}(a, \varphi_1(a), \dots, \varphi_m(a))}{n!} (\tau - a)^n d\tau ; i = \overline{1, m}\} = \\ &= \left\{ y_{0,i} + \sum_{n=0}^k \frac{f_i^{(n)}(a, \varphi_1(a), \dots, \varphi_m(a))}{(n+1)!} (x - a)^{n+1} ; i = \overline{1, m} \right\} \end{aligned} \quad (3)$$

Звичайно, постає питання збіжності методу при такій заміні підінтегральної функції, а отже і за нового вигляду оператора. Саме це питання ми і розглянемо у наступному розділі.

Доведення збіжності

Для доведення збіжності методу при такій заміні, покажемо, що оператор (3) – це оператор стиску.

Дійсно, розглянемо відстань між $\tilde{A}\varphi^1$ та $\tilde{A}\varphi^2$:

$$\begin{aligned} \rho(\tilde{A}\varphi^2, \tilde{A}\varphi^1) &= \\ &= \max_{x,i} \left| \sum_{n=0}^k \frac{f_i^{(n)}(a, \varphi_1^2(a), \dots, \varphi_m^2(a)) - f_i^{(n)}(a, \varphi_1^1(a), \dots, \varphi_m^1(a))}{(n+1)!} (x - a)^{n+1} \right| \leq \\ &\leq (*) \end{aligned}$$

Зафіксуємо n , та розглянемо різницю під знаком суми.

Оскільки $f_i^{(n)}(a, \varphi_1(a), \dots, \varphi_m(a))$ є, по суті, функціями багатьох змінних, то застосуємо до них теорему Лагранжа про скінченні прирости для функцій багатьох змінних (для зручності запису позначимо $f^{(n)}(x, y_1, \dots, y_m) \equiv F(x, y_1, \dots, y_m)$):

$$\begin{aligned} & \left| f_i^{(n)}(a, \varphi_1^2(a), \dots, \varphi_m^2(a)) - f_i^{(n)}(a, \varphi_1^1(a), \dots, \varphi_m^1(a)) \right| \equiv \\ & \equiv |F(a, \varphi_1^2(a), \dots, \varphi_m^2(a)) - F(a, \varphi_1^1(a), \dots, \varphi_m^1(a))| = \\ & = |0 + F'_{y_1}(\zeta, \xi_1, \dots, \xi_m)(\varphi_1^2(a) - \varphi_1^1(a)) + \dots + \end{aligned}$$

$$\begin{aligned}
& + F'_{y_m}(\zeta, \xi_1, \dots, \xi_m)(\varphi_m^2(a) - \varphi_m^1(a))| \leq \\
& \leq \left| \text{Позначимо } M_i = \max_k |F'_{y_k}(\zeta, \xi_1, \dots, \xi_m)| \right| \leq \\
& \leq M_i \cdot m \cdot \max_{x,i} (\varphi_i^2(a) - \varphi_i^1(a)) = \\
& = M_i \cdot m \cdot \rho(\varphi^2, \varphi^1)
\end{aligned}$$

Відмітимо, що M_i – завжди існує, як максимум неперервної на відріжку функції.

Далі, підберемо кінець інтервалу інтегрування b так, щоб виконувалась така умова:

$$\begin{cases} k \cdot M_i \cdot m \cdot |b - a|^{k+1} < 1, \text{ якщо } b > a + 1 \\ \text{або } k \cdot M_i \cdot m \cdot |b - a|^{\square} < 1, \text{ якщо } b < a + 1 \end{cases}$$

І матимемо, нарешті :

$$\begin{aligned}
(*) & \leq \max_{x,i} \left| \sum_{n=0}^k M_i \cdot m \cdot \rho(\varphi^2, \varphi^1) \cdot \frac{(x-a)^{n+1}}{(n+1)!} \right| \leq \\
& \leq \left| \text{Для визначеності нехай, } b < a + 1; \text{ і позначимо } M = \max_i(M_i) \right| \leq \\
& \leq k \cdot M \cdot m \cdot (b-a)^{\square} \cdot \rho(\varphi^2, \varphi^1) \leq \alpha \cdot \rho(\varphi^2, \varphi^1)
\end{aligned}$$

Варто відзначити, що b , тобто кінець інтервалу інтегрування, ми можемо вибирати власноруч, що дозволяє зробити $\alpha < 1$.

Тобто маємо, що:

$$\rho(\tilde{A}\varphi^2, \tilde{A}\varphi^1) \leq \alpha \cdot \rho(\varphi^2, \varphi^1)$$

Де $\alpha < 1$, що і доводить той факт, що оператор \tilde{A} є оператором стиску.

Висновки

Отже, ми довели, що метод Пікара при заміні підінтегральної функції частковою сумою Тейлора у ряд Тейлора буде дійсно збіжним. При чому чим до вищого порядку розкласти підінтегральну функцію, тим точнішим стає наближення розв'язку задачі Коші на усьому інтервалі.

Разом з тим, при доведенні нам довелося накласти додаткові умови диференційованості підінтегральних функцій. До того ж, збіжність модифікованого методу залежить від вибору кінця інтервалу інтегрування, що й не дивно, адже формула Тейлора дає хороше наближення тільки локально.

Важливою ремаркою є також той факт, що коефіцієнти Тейлора не обов'язково шукати точними, символічними методами, адже можливе застосування різницевих схем для знаходження їх наближених значень. Тим не менш, збіжність методу у такому разі не гарантується. Критерії збіжності у цьому випадку потребують подальших досліджень.

Перспективи подальших досліджень

Окреслимо такі напрямки подальшого розвитку дослідження :

- Вивід явного виразу похибки цієї модифікації методу
- Збіжність цієї модифікації методу Пікара при застосуванні формули Тейлора з коефіцієнтами, знайденими чисельно
- Визначення алгоритмічної складності цього методу та його порівняння з іншими чисельними методами
- Послаблення вимог до підінтегральних функцій для збіжності методу

Література

1. Колмогоров А.Н. Елементи теорії функцій і функціонального аналізу :Підручник / А.Н. Колмогоров, С.В. Фомін. – Київ: Видавниче об'єднання «Вища школа», 1974. – 456 с.
2. Самойленко А.М. Диференціальні рівняння :Підручник / А.М. Самойленко, М.О. Перестюк, І.О. Парасюк. – Київ: Либідь, 2003. – 2-ге вид., перероб. і доп. – 600 с. – ISBN 966-06-0249-9.

УДК 519.688

**Канд., техн., наук, ст., наук, співр. Маслянюк П.П.,
студент Мірко С.С.**

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ПОШУКУ ПРЕЦЕДЕНТІВ В СУДОЧИНСТВІ УКРАЇНИ

Abstract

Pavlo Maslianko, assoc. prof., PhD; Serhii Mirko, student;
Mathematical and software provision of the search system for precedents in the judicial system of Ukraine

This article deals with the task of finding precedents in the judiciary of Ukraine. Existing services and methods of information retrieval and text analysis are studied and discussed. The new architecture of the system of searching for precedents in the judiciary of Ukraine is proposed and described. A comparative analysis of the proposed system and existing ones is presented. Ways for further research and modernization are also suggested.

Вступ

На сьогодні для інформатизації бізнес-процесів діяльності зацікавлених осіб у судочинстві України існують інформаційно-пошукові системи (ІПС) для пошуку прецедентів серед вироків, рішень, ухвал та інших документів. Частина таких ІПС має відомчий характер, а частина відкриті для громадськості, зокрема Єдиний державний реєстр судових рішень, Закон онлайн, База правових позицій верховного суду, та ін.

Функціональність таких ІПС у частині пошуку прецедентів має досить обмежені можливості інтерфейсу користувача, вибору пошукових метрик, кількості та точності отриманих в результаті пошуку документів. Окрім цього, виникає проблема організації пошукових алгоритмів при роботі з декількома базами даних та доступу до них зацікавлених осіб з різним рівнем юридичної підготовки і необхідними знаннями інформаційних технологій.

Тому, наразі, актуальною є проблема розробки науково обґрунтованої концептуальної моделі ІПС пошуку прецедентів у судочинстві України та її імплементація на множині відкритих баз даних

для забезпечення пошуку прецедентів серед законодавчо встановлених видів результатів судочинства України.

Основна ідея реалізації такої концептуальної моделі ІПС полягає у застосуванні методології системної інженерії та вдосконаленого бізнес-профіля Еріксона–Пенкера [1]

Постановка задачі

Об'єкт дослідження. Системи пошуку документів, системи пошуку документів у судочинстві України, пошукові системи, інформаційно-пошукова система. Методи, моделі, алгоритми пошуку документів, обробка природної мови, тематична модель тексту, виділення ключових слів, лінгвістичні статистичні моделі.

Предмет дослідження. Системна інженерія та модель ІПС на основі бізнес-профіля Еріксона–Пенкера, методи: теорії системного аналізу, системної інженерії, моделювання та проектування систем пошуку текстів, обробки природної мови, тематичного моделювання текстів, виділення ключових слів, візуалізації даних.

Мета дослідження. Розробка математичного, програмного, інформаційного та методичного забезпечення для реалізації інформаційно-пошукової системи пошуку прецедентів в судочинстві України на множині відкритих баз даних для автоматизації діяльності зацікавлених осіб.

Кінцевий результат дослідження. ІПС на основі бізнес-профіля Еріксона–Пенкера, математичне і програмне забезпечення системи.

Вимоги до пошукового інтерфейсу користувача системи:

- Пошуковий рядок з операторами – надає користувачу можливість скласти складний запит за допомогою операторів: AND - ставиться між словами, якщо обидва мають бути в документі; OR - ставиться між словами, якщо хоча б одне має бути в документі; NOT- якщо слова не має бути в документі; () – якщо слова мають бути в зазначеному порядку та разом.
- Допомога в формуванні запиту – надає користувачу допомогу в пошуку шляхом підказки слів, які слід додати або виключити з запиту, щоб отримати релевантні документи на основі оцінки релевантності документів користувачем.
- Виділення ключових слів тексту – надає користувачу можливість виділити бажану кількість слів в тексті на основі їх статистичної важливості.
- Пошук за числовими сутностями – надає користувачу можливість пошуку за числовими сутностями. Числові сутності включають цифри та їх контекст, наприклад номер статті кодексу та його назва. Наприклад, якщо

користувачу потрібна стаття 124 кримінального кодексу, то в такому випадку буде знайдено документи, які містять цю сутність, а не документи, в яких є токени «124», «кримінальний», «кодекс», «стаття».

- Пошук статей кодексів – пошук релевантних статей кодексів для уточнення запиту пошуку судових документів.
- Доступ до системи надається через web-інтерфейс.

Компонентна модель ІПС пошуку прецедентів в судочинстві України

Структурне представлення системи має відповідати поставленим вимогам та мати необхідні функціональні властивості для пошуку судових прецедентів .

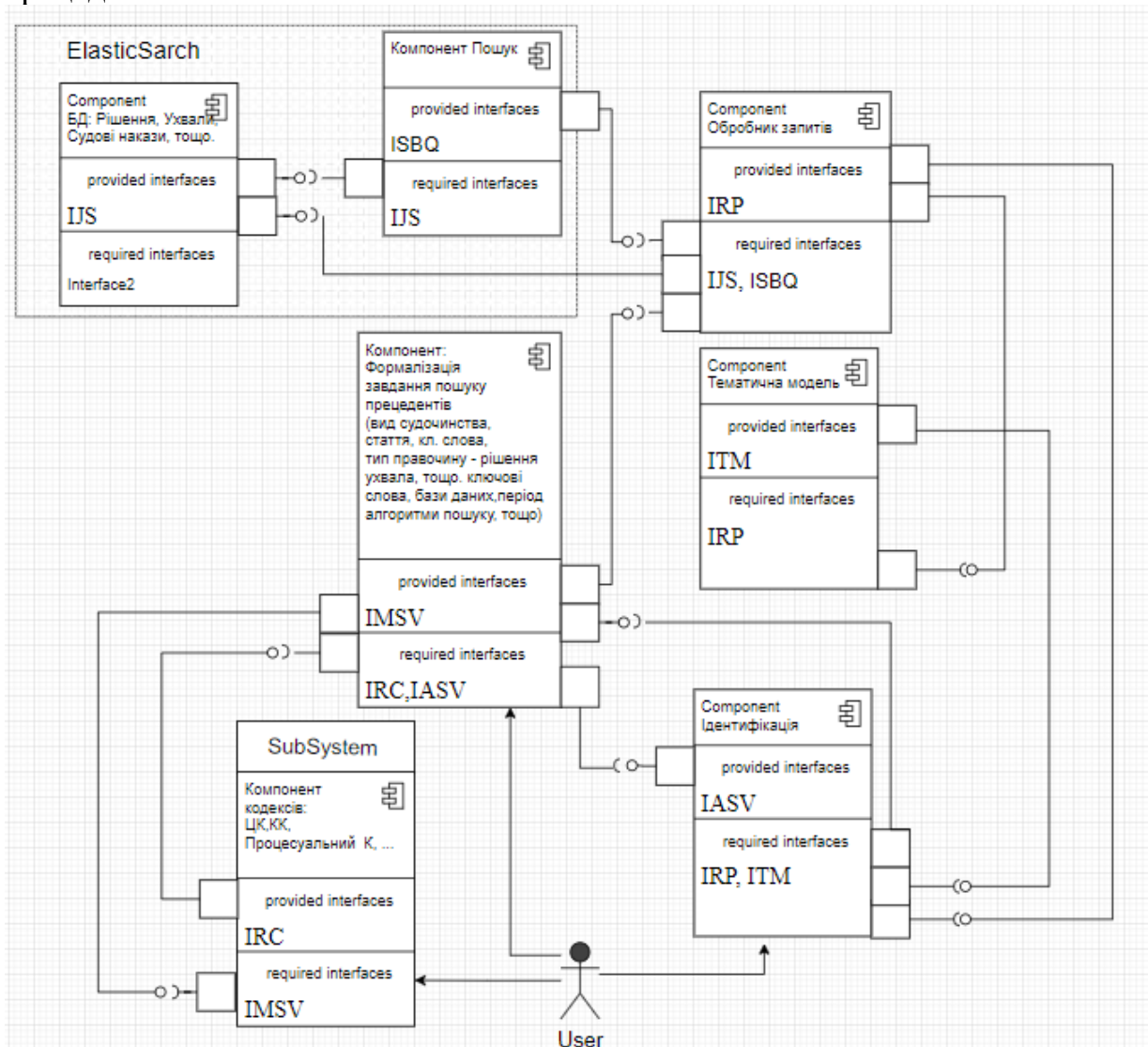


Рис. 1. Модель інформаційно-пошукової системи. Діаграма компонентів у нотації UML

Основні user case-и системи.

Сценарій для кейсу «Запит»

Актори	Користувач, система
Мета	Формалізація та обмеження запиту
Успішний сценарій: <ol style="list-style-type: none">1. Користувач переходить до вкладки пошуку2. Користувач вибирає грубі фільтри (тип судочинства, справи, проміжок часу, номери кодексів і т.д)3. Користувач вводить ключову фразу для пошуку документа4. Користувач натискає «Пошук»5. Система виводить знайдені документи кейс «Результат»	
Результат	Запит успішно сформовано
Розширення: 4а. Система не знайшла документи. Результат: Система виводить повідомлення «Документи не знайдено»	

Сценарій для кейсу «Результат»

Актори	Користувач, система
Мета	Ознайомитися з знайденими документами
Успішний сценарій: <ol style="list-style-type: none">1. Система виводить картки знайдених документів разом з їх метаданими та змодельованими темами2. При натисканні на картку документа відкривається нова сторінка в браузері, на якій відображається текст документа3. Робота з текстом / виділення слів<ol style="list-style-type: none">a. Користувач обирає кількість слів, яку хоче виділитиb. Користувач обирає, яку тему треба виділити в документі4. При натисканні «Виділити» система повертає документ зі словами, виділеними іншим кольором5. Оцінка релевантності документа, перехід до кейсу «Допомога в пошуку»	
Результат	Користувач ознайомився зі знайденими документами

Сценарій для кейсу «Допомога в пошуку»

Актори	Користувач, система
Мета	Знайти додаткові ключові слова
Успішний сценарій: <ol style="list-style-type: none">1. Користувач розмітив 5 чи більше документів2. Користувач натискає «Допомога в пошуку»3. Система порівнює статистику релевантних та не релевантних документів та надає бали словам в межах $(-\infty, +\infty)$.4. Система відкриває нову сторінку в браузері, на якій відображаються слова та їх оцінки. Чим нижча оцінка, тим більше слово рекомендовано як штрафне, чим вище оцінка, тим більше слово рекомендовано як ключове.5. Користувач обирає уточнюючі слова для запиту. Перехід до кейсу «Пошук»	

Результат	Користувач знайшов уточнюючі слов
Розширення: 1а. Користувач оцінив менше 5-ти документів Результат: Система виводить повідомлення, скільки ще документів треба оцінити	

Компонентна модель ІПС, що реалізує вказані функції складається з таких компонентів (рис.1):

- Компонент кодексів: зберігання, фільтрація/очищення, токенизація, індексація, вилучення та зберігання корисної структурованої інформації, що включає числа з документа. Пошук релевантних статей кодексів.

- Компонент судових прецедентів: зберігання, фільтрація/очищення, токенизація, індексація, вилучення та зберігання корисної структурованої інформації, що включає числа.

- Компонент формалізації: формування пошукового вектора, вид судочинства, тип справи, ключеві слова, опис, період тощо

- Компонент дерево пошуку: алгоритм відповідності з завданням пошуку Що? Як? Де? Скільки? шукаємо.

- Компонент обробник запитів: задачею компоненту бути посередником між Elasticsearch[2] та іншими компонентами системи. Тобто відправляти запити до Elasticsearch і повертати результат запиту.

- Компонент тематична модель: відповідає за підготовку та обробку даних для навчання тематичної моделі, навчання тематичної моделі та надання доступу до результатів навчання.

- Компонент ідентифікація: Відповідає за візуалізацію результатів та допомогу користувачу: виведення тексту документів, виділення слів, рекомендація слів. Результатом взаємодії з компонентом користувач за потреби уточнює пошуковий запит.

Далі наводимо список усіх інтерфейсів ІПС:

- Interface Judgment Storage (IJS) – інтерфейс надання доступу до бази судових прецедентів та статистики їх токенів;

- Interface Search By Query (ISBQ) – інтерфейс надання доступу до пошуку документів за запитом;

- Interface Request Processing (IRP) – інтерфейс надання доступу до пошуку документів за запитом;

- Interface Main Search Vector (IMSV) – інтерфейс передавання базового вектора запиту;

- Interface Additional Search Vector (IASV) - інтерфейс передавання додаткового вектора запиту;

- Interface Relevant Codex (IRC) – інтерфейс передавання знайдених

релевантних кодексів, знайдених за запитом;

- Interface Topic Modeling(ITM) – інтерфейс надання доступу до навченої тематичної моделі судових прецедентів, розподілу тем документів, і які токени утворюють кожену тему.

Модель системи формалізовано на основі мови графічного моделювання UML, що дозволяє формалізувати відношення компонентів, їх внутрішню реалізацію.

На рис. 2 наведена модель інформаційно-пошукової системи у вигляді деталізованої діаграми діяльності в нотації UML.

Компонент судових прецедентів складається з двох підкомпонентів.

1. Конвеєр обробки судових прецедентів - відповідає за формування JSON-документа судового рішення, який включає в себе текст судового документа та його метадані (суд, суддя, вид судочинства і т. ін). Окрім цього аналізувати текст та зберігати структуровані сутності з числами, такі як номери статей кодексів, посилання на інші рішення тощо.

2. Індекс ElasticSearch (аналог таблиці в реляційних базах даних) - його задачею є збереження та індексація сформованого JSON(JavaScript Object Notation)-документа за раніше заданими налаштуваннями, відображенням JSON-документа.

Компонент пошуку: це компонент структури ElasticSearch, який відповідає за пошук у кластері та функції розрахунку відповідності документа до заданих ключових слів.

Компонент формалізації завдання пошуку: це компонент з інтерфейсом користувача, який дозволяє обрати характеристики судового рішення та ввести ключеві слова. Потім компонент зчитує дані, введені користувачем, формує запит та відправляє його до компонента пошуку.

Компонент обробник запитів: це компонент, який відповідає за зв'язок системи та ElasticSearch. Він використовує Query DSL ElasticSearch для відправки запитів та повернення їх результатів.

Компонент тематична модель: за допомогою алгоритмів моделювання тем в певній вже відомій категорії документів можна виділити додаткові підтеми. Такі алгоритми дозволяють оцінювати ймовірність того, що документ був згенерований якоюсь темою та які слова в якому ступені до неї відносяться.

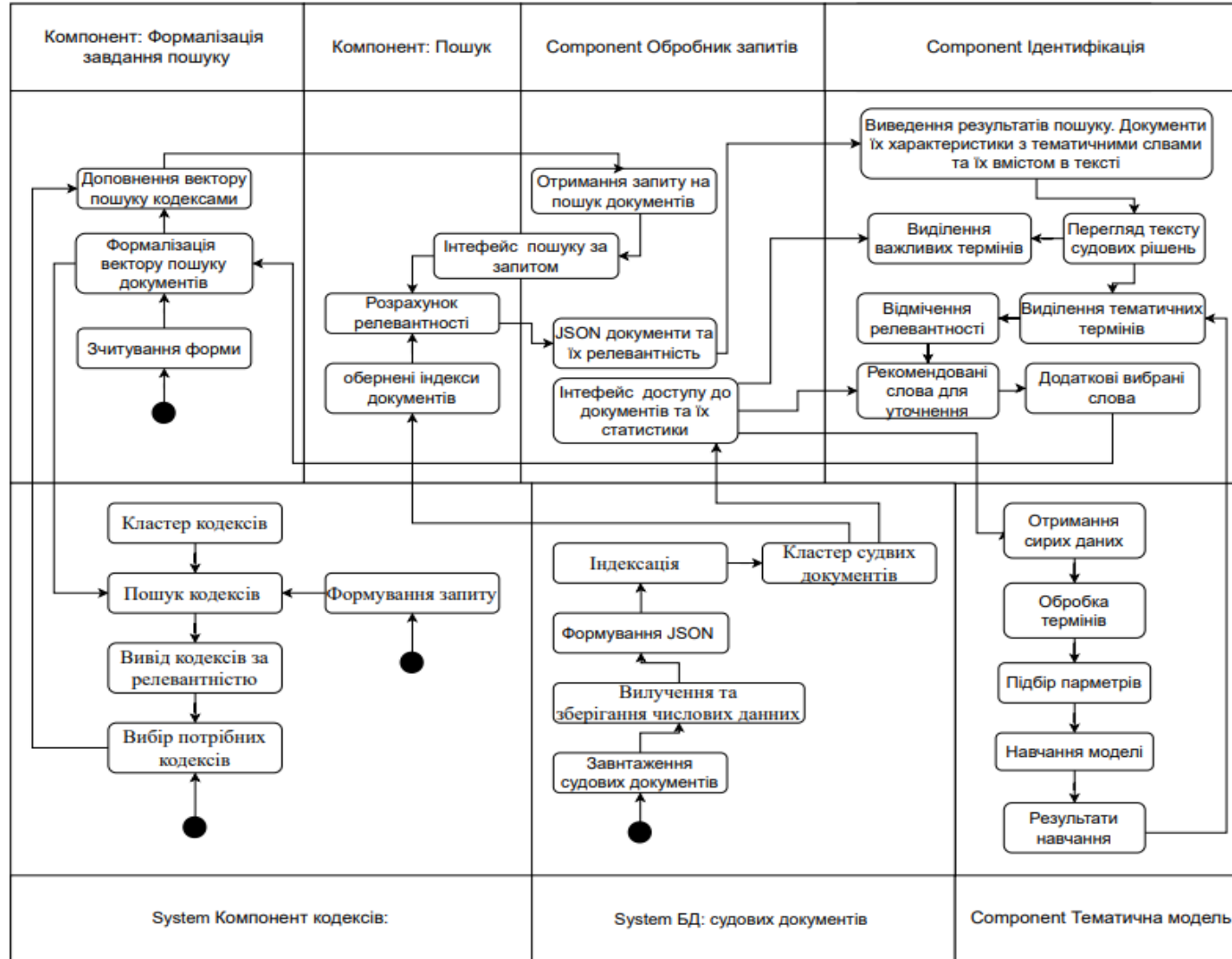


Рис. 2. Модель інформаційно-пошукової системи. Деталізована діаграма процесів у нотації UML

Таким чином, ми можемо сформувати список репрезентативних слів теми, щоб користувач міг здогадатися, чого саме стосується згенерована тема і показувати її відсотковий вміст в певному тексті. Компонент має наступні задачі:

1. Завантаження «сирих» даних, статистики токенів по документах.
2. Обробка токенів, виключення дуже частих та рідких токенів.
3. Формування матриці частот токенів в документах.
4. Підбір параметрів моделі.
5. Навчання моделі зі знайденими параметрами
6. Для кожного документа, якщо це можливо, визначення розподілу його тем та зберігання цієї інформації для подальшого використання.
7. Формування репрезентативних слів для кожної теми.
8. Надання доступу до матриці оцінок відповідності токенів до певної теми.

Компонент ідентифікація: головною задачею цього компоненту є допомога користувачу для пошуку документа. Цей компонент має наступний функціонал:

1. Повернення результатів пошуку за запитом. Користувачу виводяться картки з загальною інформацією по судовому документу (метадані) з вкладом кожної визначеної теми на етапі моделювання тем та репрезентативних слів теми в порядку їх релевантності до теми. Користувач, натискаючи на картку, може переглянути текст документа.

2. Виділення ключових слів. За допомогою статистики слів з індексу компонент розраховує важливість слова в тексті та ранжує їх у порядку статистичної важливості. Коли користувач вибирає кількість слів (n), яку хоче виділити, система бере перші n токенів зі списку важливих токенів та виділяє їх іншим кольором.

3. Допомога в пошуку. Користувач в процесі пошуку потрібного документа має змогу оцінити документ на релевантність. Після розмічення 5-ти документів система дозволить йому запросити рекомендації слів, які слід додати до пошукового запиту або виключити з нього. Коли користувач запитує систему про допомогу в пошуку, система порівнює статистику слів релевантних документів із статистикою не релевантних і на основі цих даних оцінює токени. Оцінка лежить в межах $(-\infty, +\infty)$. Слова з оцінкою менше 0 рекомендовані до включення до запиту, але як штрафні слова (чим менша оцінка, тим слово більш рекомендовано до включення як штрафне слово). Слова з оцінкою більше 0 рекомендовані до включення до запиту (чим більша оцінка, тим слово більш рекомендовано до включення).

4. Під час перегляду тексту документу користувач має змогу виділити в тексті слова які відповідають обраній темі. Для цього компонент запитує список усіх токенів документа і їх позиції в документі та за допомогою матриці оцінок відповідності токенів до певної теми надає кожному токену, якщо він є в матриці, мітку теми. Коли користувач обирає, яку тему треба виділити, компонент виділяє слова з відповідними мітками.

Компонент кодексів - це підсистема, яка має схожу функціональність та структуру вищеописаної системи, але її головною

задачею є пошук статей кодексів, які відповідають запиту користувача для уточнення запиту пошуку судових прецедентів.

Верифікація ІПС

На даний момент вже існують системи пошуку прецедентів в судочинстві України. В таблиці представлено результати порівняння трьох основних публічних ресурсів та розробленої системи ІПС.

	Єдиний державний реєстр судових рішень	Закон онлайн	База правових позицій верховного суду	Розроблена система ІПС
Пошуковий рядок з операторами	+	+	-	+
Правові позиції та пошук по ним	-	Створюється людиною	Створюється людиною	-
Допомога в формуванні запиту	-	-	-	+
Виділення ключових слів тексту	-	Виділяються лише пошукові слова	-	+
Пошук по числовим сутностям	-	-	-	+
Пошук статей кодексів	-	-	-	+

Висновки

1. Досліджено існуючі системи пошуку прецедентів у судочинстві України та встановлено їх недоліки, якими є не використання статистичної інформації текстів, що наявна в системах для забезпечення алгоритмів пошуку документів та слабка взаємодія з користувачем. Також досліджено алгоритми інтелектуального аналізу тексту і запропоновано алгоритми їх використання.

2. В роботі розроблено нову модель інформаційно-пошукової системи, головною перевагою якої є тісна взаємодія з користувачем шляхом надання статистичної інформації щодо токенів у документах у зрозумілому вигляді, за допомогою алгоритмів інтелектуального аналізу тексту.

3. Система має такі основні функції: пошуковий рядок з операторами, допомога в пошуку, виділення важливих або тематичних слів, пошук релевантних статей кодексів для уточнення запиту, пошук за числовими сутностями

4. Перспективами подальших досліджень є вдосконалення існуючих моделей інтелектуального аналізу тексту та впровадження нових для покращення та нарощення функціональності.

Література

1. Маслянюк П.П., Сельський Є.П. Метод системної інженерії систем нейронного машинного перекладу. *KPI Science News*, 2021, № 2. с. 46 – 55
2. Технічна документація ElasticSearch. Режим доступу: <https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html>

УДК 519.688

Канд., техн., наук, ст., наук., співр. Маслянко П.П,
студент Коломієць С.С.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ПРОГНОЗУВАННЯ ТА АНАЛІЗУ ПОКАЗНИКІВ ЕФЕКТИВНОСТІ ІМПОРТУ ТА ЕКСПОРТУ ЗОВНІШНЬОЇ ТОРГІВЛІ УКРАЇНИ

Abstract

Pavlo Maslianko, assoc. prof., PhD; Serhii S. Kolomiets, student

Mathematical and software support of the system of forecasting and analysis of indicators of the efficiency of import and export of foreign trade of Ukraine

The article discusses the results of the development of a system of forecasting and analysis of indicators of the effectiveness of foreign trade of Ukraine for all groups of goods and for individual ones, based on methods of system engineering and methods of Data Science. ARIMA model, correlation analysis and clustering are used for forecasting and analysis of foreign trade performance indicators. The design and development of the system will benefit importers, exporters, individual ministries that issue recommendations on the sale of various groups of goods, and the customs service that regulates imports and exports.

Вступ

Наразі для експортерів, імпортерів, а також для окремих міністерств в залежності від групи товарів, насамперед дуже важливо бути в курсі усіх подій, що відбуваються, впливають та можуть вплинути у перспективі на показники ефективності імпорту та експорту зовнішньої торгівлі України, особливо зараз, коли війна розв'язана Росією 24 лютого впливає на всі види бізнесу та на інші ринки праці у нашій країні і на динаміку зміни показників ефективності імпорту та експорту зовнішньої торгівлі в тому числі. Застосовано модель ARIMA, для прогнозування на основі історичних даних, кореляційний аналіз, кластеризацію для аналізу показників ефективності - визначення типів зв'язку між показниками ефективності, віднесення товару до певного кластеру у вибрані періоди часу.

Постановка задачі

Метою цієї статті є застосування бізнес-профіля Еріксона-Пенкера [1,2] для моделювання системи, необхідної для прогнозування та аналізу

показників ефективності імпорту та експорту.

Предметом дослідження є математичне та програмне забезпечення системи для прогнозування, кластеризації та аналізу показників ефективності імпорту та експорту зовнішньої торгівлі України на основі моделей аналізу даних, моделей DataScience.

Для розв'язання поставленої задачі було оглянуто методи прогнозування та аналізу зовнішньої торгівлі у вигляді методів прогнозування часових рядів. Моделі прогнозування, аналізу та кластеризації показників ефективності імпорту та експорту на основі моделей DataScience: декомпозиції часових рядів, регресійні моделі, експоненційне згладжування, моделі ARIMA для прогнозування, кореляційного аналізу, метод кластеризації k-means, метод DBSCAN. Математичні інструменти: оцінки точності прогнозування та аналізу, кореляції Пірсона. Моделі ефективності діяльності зовнішньої торгівлі України: індекси фізичного обсягу Ласпейреса (експорту/імпорту), індекси цін Пааше (експорту/імпорту), індекс умов торгівлі (кількісний, ціновий).

Огляд існуючих систем прогнозування та аналізу: AmazonForecast (AF), OracleSiebelForecasting, ForecastPro.

Вимоги до системи:

- доступ до системи надається через платформу-застосунок.
- користувач системи може зробити вибір, або зробити прогнозування по усім групам товарів, або по окремим
- користувач системи може виконати кластеризацію в окремі періоди часу, з прив'язкою до дат надзвичайних подій визначених системою – віднести групу товарів до певного кластеру за ознаками найбільшого та найменшого впливу.
- користувач системи може дослідити кореляцію між окремими показниками з прив'язкою до дат надзвичайних подій визначених системою.

Розробка моделі системи прогнозування, аналізу та кластеризації

Компонентна модель системи прогнозування та аналізу показників ефективності імпорту та експорту, що реалізує задачі цієї системи складається з таких компонентів (рис. 1): Сховище даних показників ефективності імпорту та експорту, Обробка даних показників ефективності імпорту та експорту, Кластеризація даних за показниками ефективності імпорту та експорту зовнішньої торгівлі, Дослідження кореляції між показниками ефективності зовнішньої торгівлі, Прогнозування на основі моделі ARIMA, Платформа застосунок.

Опис компонентів моделі системи прогнозування, аналізу та кластеризації

1. Компонент Сховище даних показників ефективності імпорту та експорту. Сховище даних – необроблений набір даних за групами товарів, яких 95 варіацій – по кожній групі товару представлено помісячні показники ефективності діяльності зовнішньої торгівлі : індекс фізичного обсягу(імпорту/експорту) , індекс цін(імпорту/експорту), індекс умов торгівлі(кількісний/ціновий). Цей набір даних – датасет, саме у цьому компоненті завантажується та зчитується для подальшого виконання задачі кластеризації, прогнозування із застосуванням моделі ARIMA та аналізу даних за допомогою кореляційного аналізу.

2. Компонент Обробка даних показників ефективності імпорту та експорту. Призначений для первинної очистки даних. Забезпечує обробку набору даних про показники ефективності зовнішньої торгівлі: індекс фізичного обсягу(імпорту/експорту), індекс цін(імпорту/експорту), індекс умов торгівлі(кількісний/ціновий). Компонент реалізує алгоритм, розбиття набору даних на тренувальний, тестовий та валідаційний, розбиття великої таблиці на кілька маленьких (нормалізація, групування, зменшення надмірності та залежності).

3. Компонент Кластеризація даних за показниками ефективності імпорту та експорту зовнішньої торгівлі. Компонент реалізує алгоритм k-means для кластеризації при використанні даних про показники ефективності діяльності зовнішньої торгівлі за окремими групами товарів, у періоди часу що пов'язані з екстремальними подіями та розбиває їх на кластери за встановленими ознаками: перелік груп товарів на які найбільше впливають непередбачувані події та на які найменше впливають.

4. Компонент Прогнозування на основі моделі ARIMA. Компонент реалізує модель ARIMA для прогнозування часового ряду. У цьому компоненті протягом кожної ітерації часовий нестационарний ряд, зводиться до стаціонарного ряду за допомогою коефіцієнтів, що описують модель - p, q, r , котрі підбираються протягом кожної ітерації та підбираються таким чином, щоб похибки були мінімальні. Опісля того, як було знайдено оптимальні p, q та r – на основі цих параметрів, котрі підставляються у функцію і прогноуються майбутні значення у інші часові проміжки.

5. Компонент Дослідження кореляції між показниками ефективності. Компонент призначений для кореляційного аналізу – встановлення статистичної залежності між двома випадковими величинами, котра носить імовірнісний характер. Обчислюється коефіцієнт кореляції двох величин, який і характеризує тип зв'язку.

6 Компонент Платформа-застосунок. Компонент, що має такий інтерфейс та такий функціонал:

- зчитування файлу, для подальших операцій, визначених системою – прогнозування, аналізу, класифікації та кластеризації: зчитування файлу передається шляхом інтерфейсу IIDR на вхід функції обробки даних датасету – даних про показники ефективності імпорту та експорту зовнішньої торгівлі.

- вибір, та зчитування окремих груп товарів по яким робиться прогнозування від (1-95), якщо користувач вводить “0”, то робиться прогнозування по усім групам товарів.

- вибір досліджуваного проміжку для кластеризації, та аналізу даних та його зчитування.

- введення тих показників ефективності між якими треба дослідити кореляційний зв’язок

- перегляду результатів прогнозування, аналізу, класифікації та кластеризації (отриманий через інтерфейс IPA) – функція виведення результатів.

На (рис. 2) показано внутрішню структуру компонентів системи, представляє собою елементи діаграми компонентів, компоненти якої розмежовані водними доріжками, котрі не дозволяють їм вийти за межі батьківського компонента.

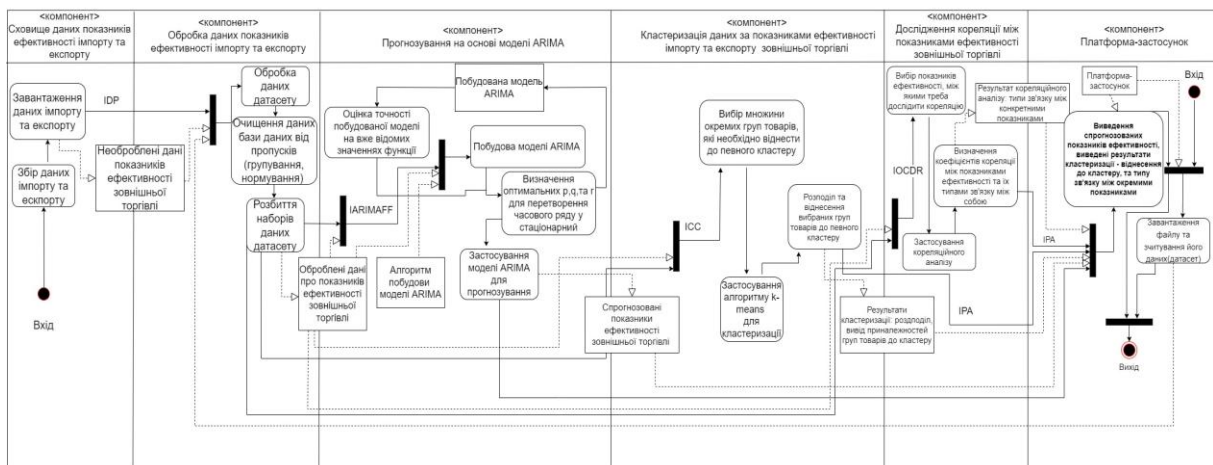


Рис. 2. Модель системи прогнозування, аналізу та кластеризації.
Деталізована діаграма діяльності в нотації UML

Імплементація системи

Для демонстрації окремих функцій системи використовується датасет сформований із сайту ukrstat.gov.ua [3] «Індекси фізичного обсягу, індекси цін та умови торгівлі».

У якості тестового прикладу виконується завдання прогнозування показників ефективності, на основі історичних даних про показники ефективності за допомогою моделі ARIMA. Дані за період з січня 2019 року помісячно по липень 2022 року – це і буде нашим тестовим прикладом за показниками ефективності, що стосуються експорту та кількісного оцінювання умов торгівлі для прогнозування. Є 3 показники, кожен з яких вираховується помісячно. Таким чином у нашому тестовому прикладі 126 значень показників ефективності, кожні 3 із яких прив’язані до певного року та місяця, таким чином на основі цих даних і буде робитись прогнозування. Кожен рядок містить інформацію про код та назву групи товарів, індекси фізичного обсягу експорту, індекси цін експорту та індекси умов торгівлі кількісний помісячно, що охарактеризовано такими елементами набору даних:

`time_2` – місяць та рік, протягом якого вираховані показники ефективності

`ind_cnt` – показники індексів обсягів експорту

`ind_price` – показники індексів цін обсягів експорту

`ind_quant` – показники індексів кількісних умов торгівлі

На (рис. 3) зображені перші 5 рядків тестового прикладу.

	<code>time_2</code>	<code>ind_cnt</code>	<code>ind_price</code>	<code>ind_quant</code>
0	Jan-2019	106.6	102.4	111.3
1	Feb-2019	104.7	101.8	90.0
2	Mar-2019	107.4	99.3	100.9
3	Apr-2019	102.1	99.9	89.8
4	May-2019	113.2	99.8	105.0

Рис.3 Візуалізація таблиці набору даних контрольного прикладу про ефективності діяльності зовнішньої торгівлі

Для того, щоб оцінити точність побудованої моделі ARIMA для прогнозування, необхідно розділити вибірку на тестову та тренувальну, таким чином, щоб тренувальний набір даних це були перші 80% значень, а тестова вибірка – це останні 20% значень. На (рис. 4) зображено графік динаміки індексів цін експорту – візуалізовано динаміку 100% вибірки тестового прикладу.

Опісля розподілу вибірки необхідно зробити прогноз на основі цих 80% даних, про те, якими стануть 20% даних, підбираються параметри ARIMA для того, щоб зробити зі стаціонарного ряду нестационарний, та робиться прогноз.

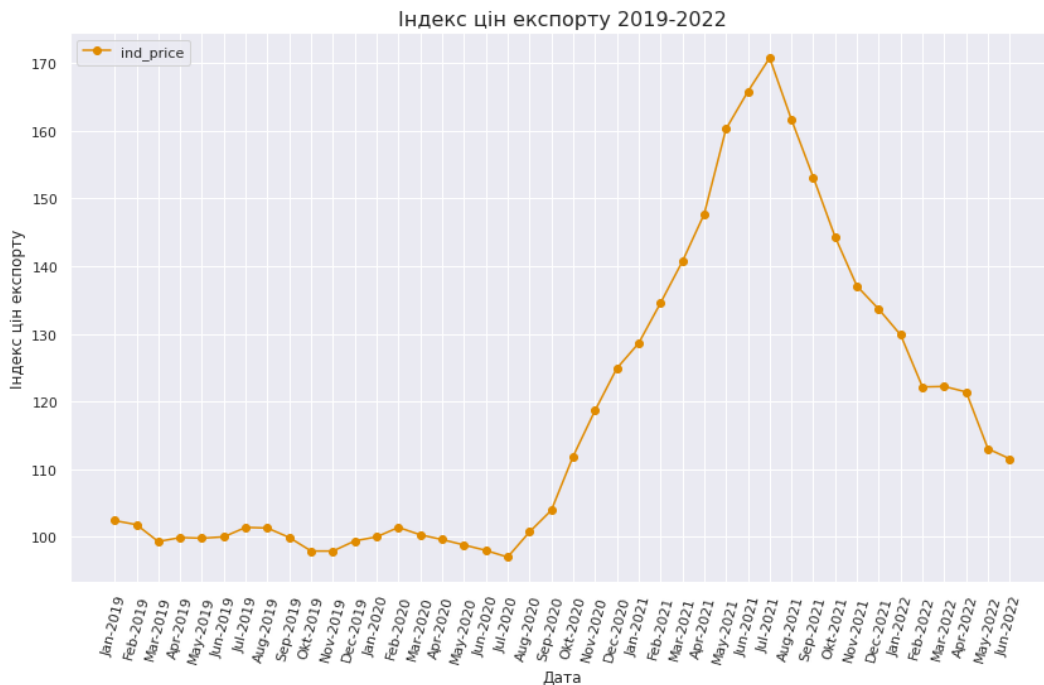


Рис.4 Графік динаміки індексів цін експорту

Після підбору параметрів, було визначено, що для прогнозування індексу цін, задовільними є параметри моделі ARIMA (0,1,5), будемо її. На (рис. 5) зображені сумарні результати моделі ARIMA.

```

SARIMAX Results
=====
Dep. Variable:      ind_price      No. Observations:      33
Model:             ARIMA(0, 1, 5)  Log Likelihood         -76.194
Date:              Sun, 16 Oct 2022 AIC                     168.388
Time:              11:51:49       BIC                     180.114
Sample:            0               HQIC                    172.275
                  - 33
Covariance Type:   opg
=====
              coef  std err      z      P>|z|    [0.025    0.975]
-----
ind_cnt      0.0096   0.058     0.167   0.868    -0.103    0.123
ind_quant   -0.0066   0.055    -0.121   0.904    -0.114    0.101
ma.L1       0.8657   0.247     3.512   0.000     0.382    1.349
ma.L2      1.4149   3.375     0.419   0.675    -5.201    8.030
ma.L3       0.7213   2.992     0.241   0.810    -5.144    6.586
ma.L4       0.4496   1.451     0.310   0.757    -2.395    3.294
ma.L5      -0.1535   0.786    -0.195   0.845    -1.695    1.388
sigma2      5.6096   20.298     0.276   0.782   -34.173   45.393
=====
Ljung-Box (L1) (Q):      0.00  Jarque-Bera (JB):      3.62
Prob(Q):                 0.96  Prob(JB):              0.16
Heteroskedasticity (H):  6.32  Skew:                  -0.12
Prob(H) (two-sided):    0.00  Kurtosis:              4.63
=====

```

Рис.5 Сумарні результати роботи моделі ARIMA

На (рис. 6) можна візуально оцінити наскільки спрогнозовані дані індексів цін відрізняються від фактичних. Синя лінія на графіку, train –

80% даних використаних для прогнозування, зелена лінія на графіку, test - 20% даних використаних для перевірки точності нашого прогнозу та для оптимізації моделі ARIMA, червона лінія на графіку, forecast – прогнозування на основі 80% даних – тренувальна вибірка та отримання спрогнозованих 20% даних, які і порівнюються з фактичними.

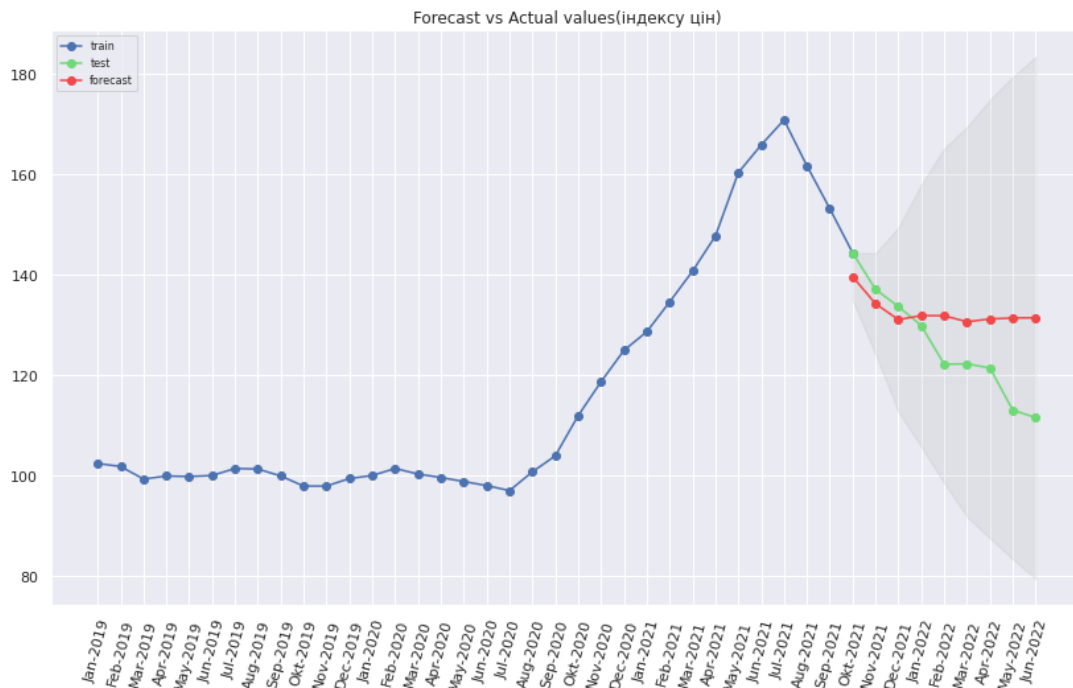


Рис.6 Візуалізація порівняння спрогнозованих індексів цін з фактичними індексами цін

Такий самий порядок дій пророблений і для прогнозування інших показників ефективності, які можна побачити на наступних (рис. 7)-(рис. 10)

Окрім гарних результатів прогнозування є і погані, оскільки коефіцієнти p, q та r підбирались вручну. В подальшому, буде реалізовано алгоритм автоматичного підбору параметрів для кожного часового ряду, оскільки часових рядів по кожній групі товарів багато.

Верифікація

Верифікація, перевірка— підтвердження на основі надання об’єктивних свідчень того, що встановлені вимоги були виконані. Виходячи із цього визначення, було реалізовано - проєкт системи прогнозування та аналізу показників ефективності імпорту та експорту. Протестовано реалізовану частину проєкту на основі тестового прикладу та визначено, що система відповідає поставленим вимогам.

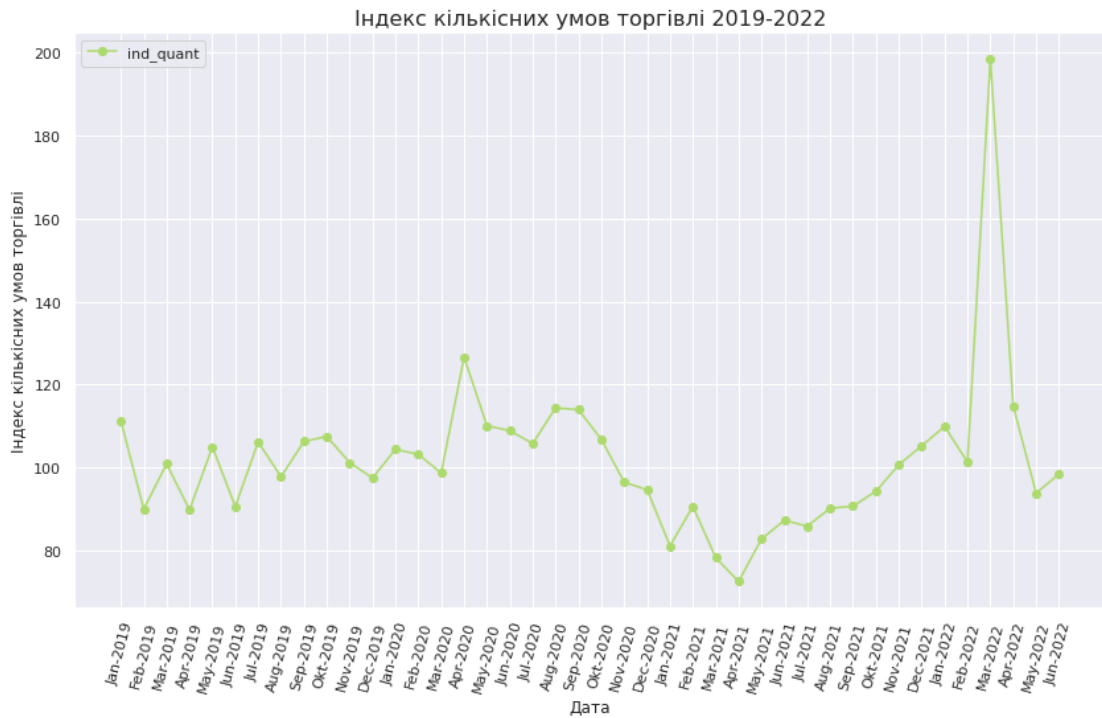


Рис.7 Графік динаміки кількісних умов торгівлі

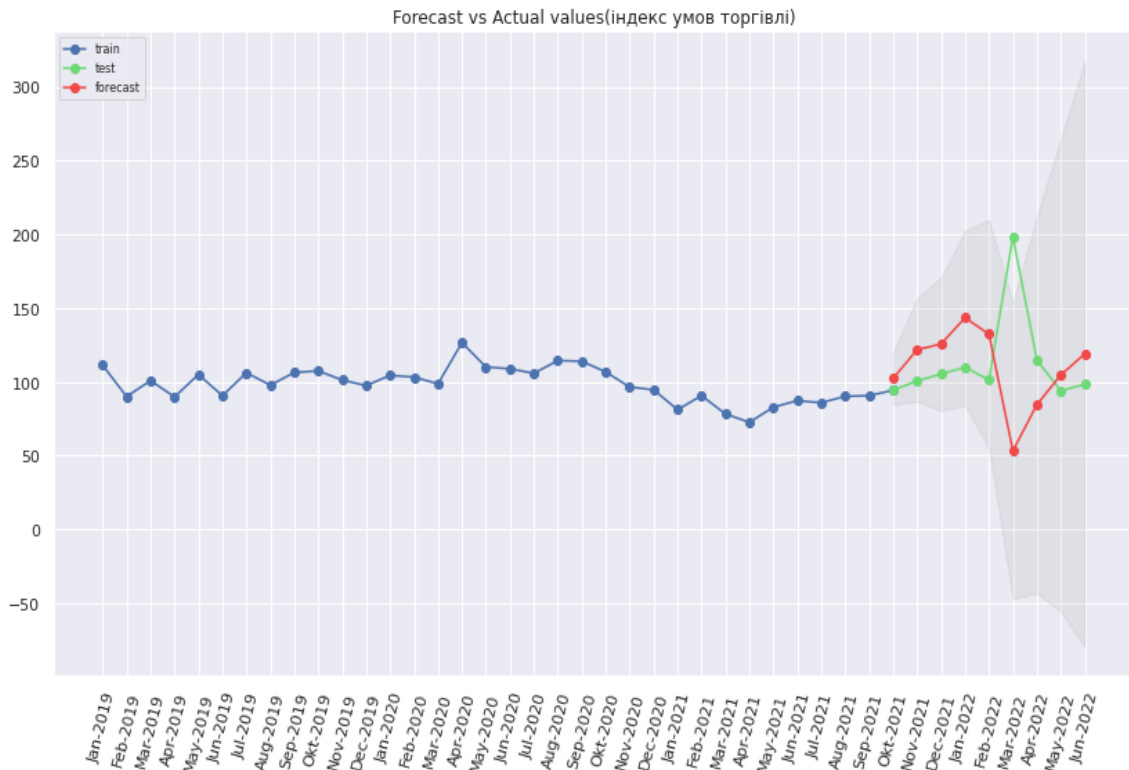


Рис.8 Візуалізація порівняння спрогнозованих індексів кількісних умов торгівлі з фактичними індексами кількісних умов торгівлі

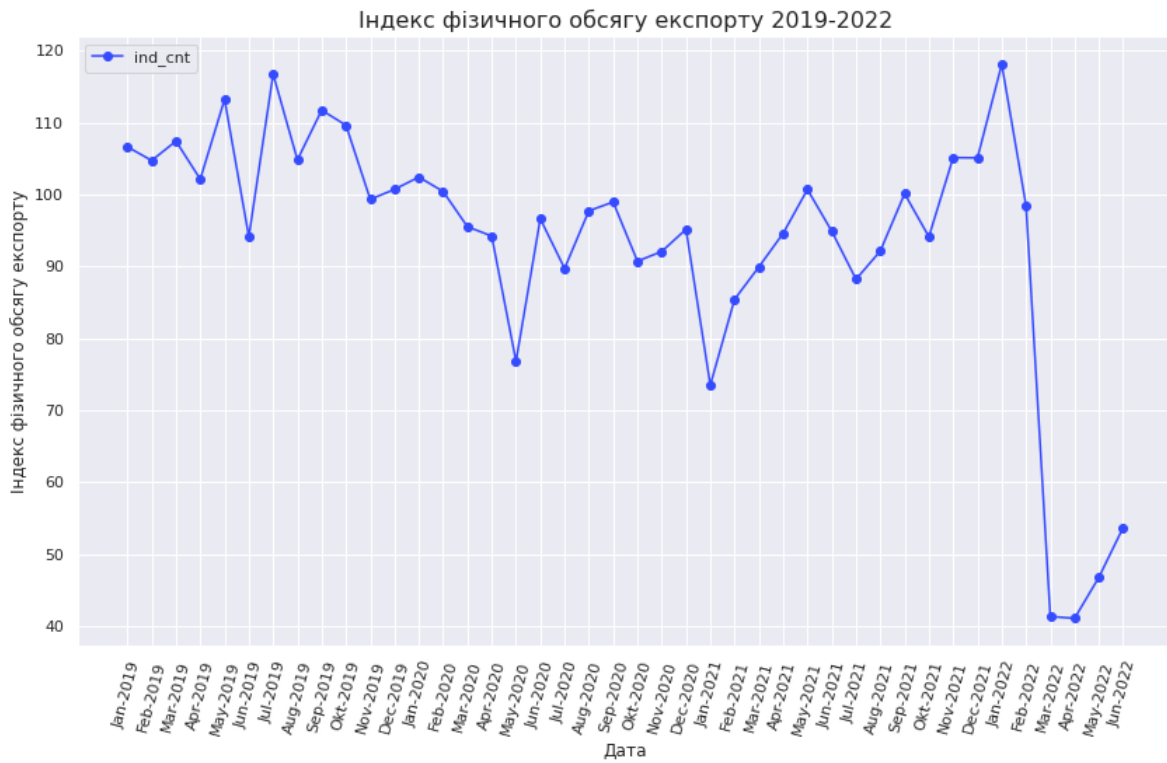


Рис.9 Графік динаміки індексу фізичного обсягу

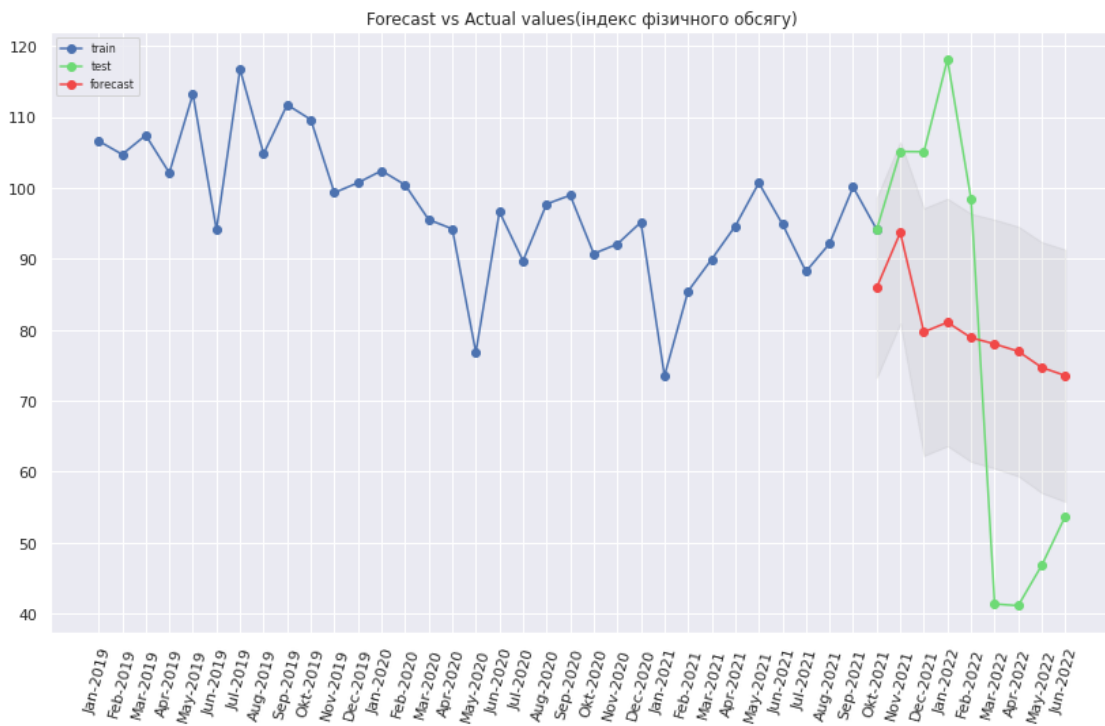


Рис.10 Візуалізація порівняння спрогнозованих індексів фізичного обсягу експорту з фактичними індексів фізичного обсягу експорту

Валідація

Валідація (validation — надання законної сили)— підтвердження на основі надання об'єктивних свідчень того, що встановлені вимоги, призначені для конкретного використання і застосування, виконані.

Для прогнозування динаміки показників ефективності імпорту та експорту застосовується модель ARIMA – на основі множини даних за минулі часові проміжки про динаміку зміни показників ефективності зовнішньої торгівлі. Для оцінки точності цього прогнозу розраховуються середня абсолютна відсоткова похибка (MAPE)– (MeanAbsolutePercentageError) [4], та середньоквадратична похибка (MSE) – (MeanSquaredError) [5] – обчислюється за формулами (1)та (2):

$$MAPE = \frac{100\%}{L} \sum_{i=1}^L \left| \frac{y_i - \bar{y}_i}{y_i} \right| \quad (1)$$

$$MSE = \frac{1}{L} \sum_{i=1}^L (y_i - \bar{y}_i)^2 \quad (2)$$

y_i - фактичне відоме значення із датасету

\bar{y}_i - спрогнозоване значення на основі моделі ARIMA

L – інтервал прогнозу

MSE – мінімізується протягом кожної ітерації, MAPE – оцінка побудованої моделі, на основі цих мір можна зробити наступну оцінку точності і на основі цієї оцінки точності у таблиці 1 наведено інтервали, що відповідають якісним характеристикам прогнозу.

Таблиця 1

Якісна оцінка точності прогнозу на основі MAPE

MAPE	Точність прогнозу
Менше 10%	Висока
10% - 20%	Добра
20% - 40%	Задовільна
40% - 50%	Погана
Більше 50%	Незадовільна

На (рис. 11)-(рис. 13) зображені розраховані фактичні значення оцінки точності прогнозування показників ефективності зовнішньої торгівлі, що вийшли –MAPE та MSE для того, щоб не тільки візуально запевнитись у точності побудованої моделі ARIMA, а підкріпити це у числовому вигляді.

MAPE test: 0.07
MSE test: 114.52

Рис.11 Візуалізація обчислених похибок моделі ARIMA, для прогнозування індексів цін

MAPE test: 0.18
MSE test: 954.08

Рис.12 Візуалізація обчислених похибок моделі ARIMA, для прогнозування індексів умов торгівлі

MAPE test: 0.41
MSE test: 729.01

Рис.13 Візуалізація обчислених похибок моделі ARIMA, для прогнозування індексів обсягу продажів

На основі даних про обчислення похибки MAPE при застосування моделі ARIMA на (рис.11) - (рис. 13) для прогнозування показників ефективності, та таблиці 1, що свідчить про оцінки точності прогнозу, у таблиці 2 наведені якісні характеристики нашого прогнозу для трьох спрогнозованих показників ефективності.

Таблиця 2

Якісна оцінка точності прогнозу розробленої системи ARIMA за показниками ефективності зовнішньої торгівлі на основі MAPE

Показники ефективності	Значення середньої абсолютної відсоткової похибки - MAPE	Якісна оцінка прогнозу моделі ARIMA
Індекс цін експорту	7%	Висока точність прогнозу
Індекс кількісних умов торгівлі	18%	Добра точність прогнозу
Індекс фізичного обсягу	41%	Погана точність прогнозу

Для оцінки кореляції Пірсона, щоб дослідити зв'язки між показниками ефективності необхідно скористатись формулою (3) [6]:

$$R_{xy} = \frac{\sum_{i=0}^{n-1} (x_i - \bar{x}) * (y_i - \bar{y})}{\sqrt{\sum_{i=0}^{n-1} (x_i - \bar{x})^2 * \sum_{i=0}^{n-1} (y_i - \bar{y})^2}} \quad (3)$$

x_i та y_i – значення випадкових величин

n – кількість спостережень

На (рис. 14) зображено зв'язки між показниками ефективності зовнішньої торгівлі за період з січня 2019 року помісячно по липень 2022 року та представлено у вигляді матриці кореляцій, що свідчить про характер взаємодії між показниками.

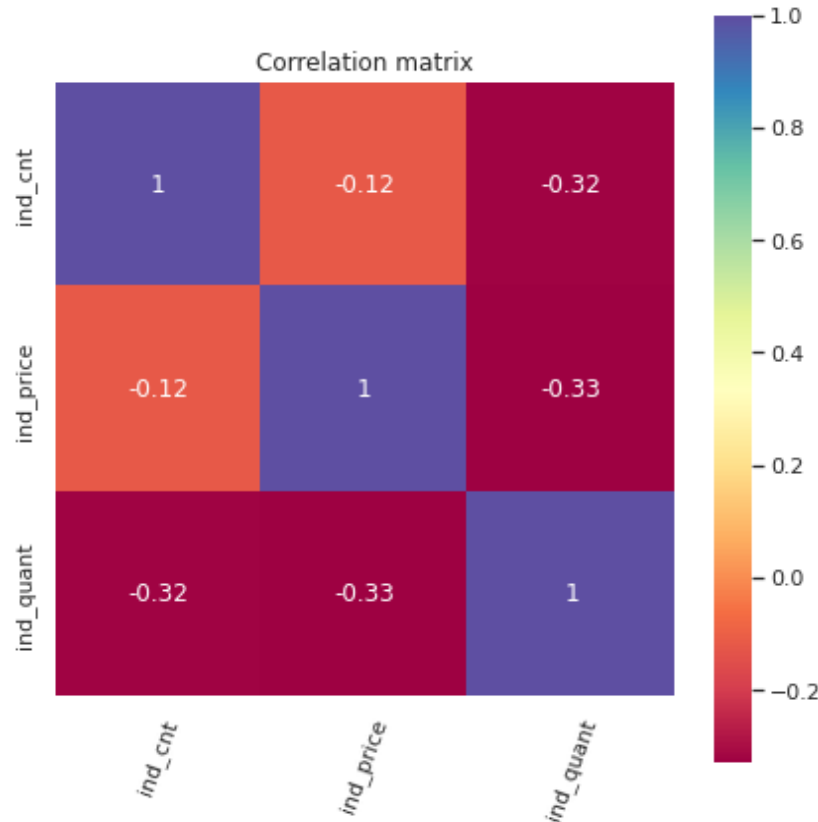


Рис.14 Візуалізація коефіцієнтів кореляції між показниками ефективності

Для того, щоб дати якісну характеристику типів зв'язку між показниками ефективності, необхідно скористатись шкалою Чеддока на таблиці 3.

Таблиця 3

Якісна оцінка коефіцієнта кореляції за шкалою Чеддока

Тіснота зв'язку	Значення коефіцієнта кореляції при наявності	
	Прямий зв'язок	Зворотній зв'язок
Слабка	0,1-0,3	-(0,3)-(0,1)
Помірна	0,3-0,5	-(0,5)-(0,3)
Помітна	0,5-0,7	-(0,7)-(0,5)
Висока	0,7-0,9	-(0,9)-(0,7)
Дуже висока	0,9-1	-(1)-(0,9)

На основі якісної оцінки коефіцієнта кореляції за шкалою Чеддока та візуалізації коефіцієнтів кореляції між показниками ефективності, на таблиці 4 наведені якісні характеристики типів зв'язку між показниками ефективності.

Таблиця 4

Матриця якісних характеристик типів зв'язку між показниками ефективності

Показник ефективності	Індекс фізичного обсягу експорту	Індекс цін експорту	Індекс кількісних умов торгівлі
Індекс фізичного обсягу експорту	Прямий зв'язок із високої тіснотою (+1)	Зворотній зв'язок із слабкою тіснотою (-0,12)	Зворотній зв'язок із помірною тіснотою (-0,32)
Індекс цін експорту	Зворотній зв'язок із слабкою тіснотою (-0,12)	Прямий зв'язок із високої тіснотою (+1)	Зворотній зв'язок із помірною тіснотою (-0,33)
Індекс кількісних умов торгівлі	Зворотній зв'язок із помірною тіснотою (-0,32)	Зворотній зв'язок із помірною тіснотою (-0,33)	Прямий зв'язок із високої тіснотою (+1)

Висновки

1. У роботі досліджено існуючі методи, моделі та системи прогнозування та аналізу, а також датасети міністерства статистики, що стосуються оцінювання показників ефективності зовнішньої торгівлі України. Встановлено, що для таких розрахунків застосовуються такі методи: декомпозиція часових рядів, регресійні моделі, експоненційне згладжування, модель ARIMA для прогнозування, кореляційний аналіз, метод кластеризації k-means, метод кластеризації DBSCAN, а також встановлено що існуючі системи є дуже дорогими, громіздкими і не мають функціональної реалізації окремих розрахунків.

2. Розроблено систему прогнозування та аналізу показників ефективності зовнішньої торгівлі, що дає можливість розрахувати динаміку поведінки цих показників на заданому інтервалі часу, на тестовому прикладі, сформованого на основі датасету із сайту ukrstat.gov.ua [3] «Індекси фізичного обсягу, індекси цін та умови торгівлі».

3. Перевагами системи є її дешевизна, простота застосування, а також наявності функцій первинної обробки даних у порівнянні з існуючими системами, оскільки, існуючі системи мають високу повну вартість володіння та складність в експлуатації.

4. Перспективи подальших досліджень полягають у нарощуванні функціональності – кластеризація по групам товарів, що представлені на імпорт та експорт - віднесення їх до певного кластеру товарів, на які

сильно вплинули екстремальні умови і на які найменше вплинули екстремальні умови; кореляційний аналіз, для оцінювання прихованих зв'язків між групами товарів імпорту та експорту.

Література

1. Eriksson, H. E., & Penker, M. (2000). Business modeling with UML. New York, 1-12 - Режим доступу: [ModeOf Access](#).

2. Маслянюк П.П., Майстренко О.С. Системна інженерія проектів інформатизації організаційних систем // Наукові вісті НТУУ "КПІ". – 2008. - № 6. - С.34-42.

3. Ukrstat.gov.ua [Електронний ресурс]. [ModeOf Access](#).

4. Indeed.com [Електронний ресурс]. [ModeOf Access](#).

5. Statisticshowto.com [Електронний ресурс]. [ModeOf Access](#).

6. Medium.com [Електронний ресурс]. [ModeofAccess](#).

УДК 519.688

Доцент Маслянюк П.П., студентка Питайло І.Ю.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ОЦІНЮВАННЯ ТА АНАЛІЗУ ТРИВОЖНОСТІ СУСПІЛЬСТВА

Abstract

Pavlo Maslianko, docent; Iryna Pytailo, student

Mathematical and software system for social anxiety assessment and analysis

This paper discusses the task of anxiety assessment in society. It proposes the architecture for such a system and reflects on the implications for further research. It discusses the current advancements in anxiety prediction and suggests a way to view it from a more detailed perspective.

Вступ

Депресія є одним із захворювань, які часто зустрічаються в людей, незалежно від віку та статі. В основі депресії лежить тривожність. В Україні спостерігається надзвичайно високий рівень депресії як в довоєнний час, так і особливо зараз. Часто люди з депресією не звертаються до лікаря, а інколи такий стан призводить до самогубства [1]. Наразі запропоновано багато моделей, які прогнозують наявність тривожності та депресії за текстом. Наприклад, дає гарні результати розбивка депресії на декілька груп і визначення кожної з них [2], використання лінгвістичних маркерів депресії [3, 4]. Це збільшує точність моделі і робить її більш зрозумілою.

Однак тривожність та депресія не завжди прив'язані виключно до однієї людини, вони не завжди мають сприйматись в ізоляції від суспільства. Зростання рівня депресії та тривожності в певному регіоні або серед певної групи людей може свідчити про зміну в середовищі, у якому вони мешкають і про певні явища, що в ньому відбуваються. Наприклад, погіршення економічного стану та збільшення рівня злочинності. Такі причинно-наслідкові зв'язки можуть стати корисними для подальшої роботи державних органів влади.

Ця стаття пропонує архітектуру системи оцінювання та аналізу тривожності суспільства на основі дописів у соціальних мережах.

Постановка задачі

Об'єктом дослідження є моделі класифікації, метод опорних векторів та рекурентні нейронні мережі, методи LDA та LSA, методи кластеризації.

Предметом дослідження є математичне та програмне забезпечення системи оцінювання тривожності суспільства на основі текстових даних, визначення тематики тексту і подальша кластеризація таких текстів за рівнем тривожності та тематикою, на основі машинного навчання.

Метою дослідження є моніторинг стану тривожності суспільства та визначення факторів, які його обумовлюють.

Кінцевий результат – математичне, програмне та лінгвістичне забезпечення системи оцінювання та аналізу рівня тривожності в суспільстві.

Вимоги до системи:

- система повинна визначати рівень тривожності за шкалою від 0 до 1 включно;
- система повинна визначати теми тексту;
- система повинна об'єднувати тексти зі схожою тематикою і рівнем тривожності у кластери;
- доступ до системи повинен надаватись через веб-інтерфейс.

Компонентна модель системи ідентифікації оціночних суджень

Система має виконувати такі основні функції:

- завантаження даних до системи;
- зберігання даних у сховищі;
- обробка даних;
- навчання, валідація та тестування моделі;
- оцінка тривожності тексту;
- визначення тематики тексту;
- кластеризація текстів;
- візуалізація результатів;
- надання веб-інтерфейсу для користувача.

На рис. 1 представлена модель системи оцінювання та аналізу рівня тривожності в суспільстві в нотації UML.

Модель складається з шести функціональних компонентів і інтерфейсів між ними. Внутрішня структура компонентів призначена для реалізації конкретних функцій кожного компонента і показана на рис. 2.

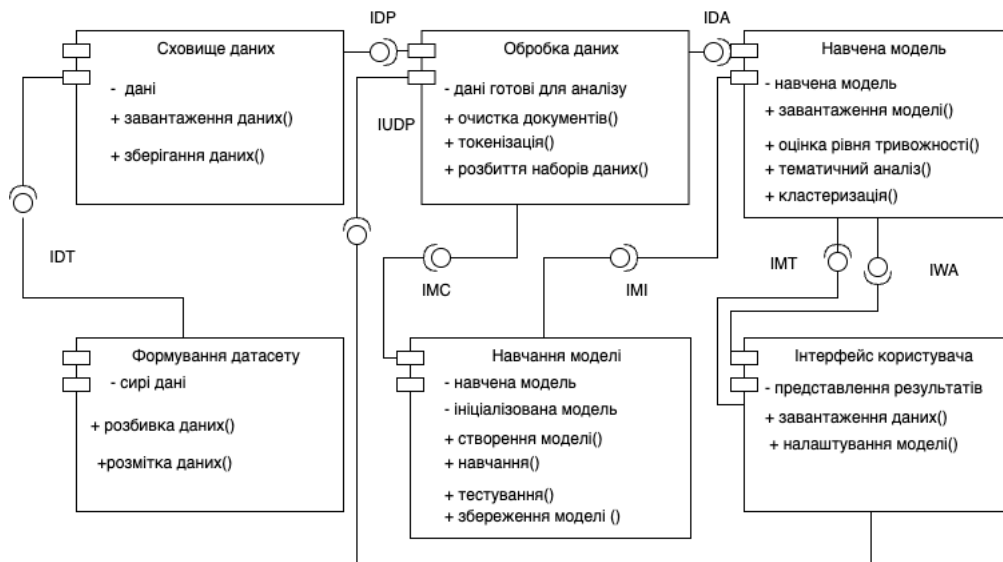


Рис. 1. Модель системи оцінювання та аналізу рівня тривожності в суспільстві.
 Діаграма компонентів в нотатції UML

Інтерфейси – це реалізація механізму взаємодії між компонентами системи.

Система містить такі інтерфейси:

- IDT (Interface Data Transfer) — інтерфейс для перенесення даних до сховища;
- IDP (Interface Data Processing) — інтерфейс обробки даних для тренування моделі;
- IUDP (Interface User Data Processing) — інтерфейс завантаження даних користувача;
- IMC (Interface Model Creation) — інтерфейс навчання моделі;
- IMI (Interface Model Initiation) — інтерфейс ініціалізації моделі для використання системою;
- IDA (Interface Data Analysis) — інтерфейс оцінювання рівня тривожності, визначення тематики тексту та кластеризації даних;
- IMT (Interface Model Tuning) — інтерфейс налаштування моделі користувачем;
- IWA (Interface Web Application) — інтерфейс для виведення результатів і функціонування веб-застосунку.

Модель діяльності системи оцінювання та аналізу рівня тривожності в суспільстві наведена на рис. 2.

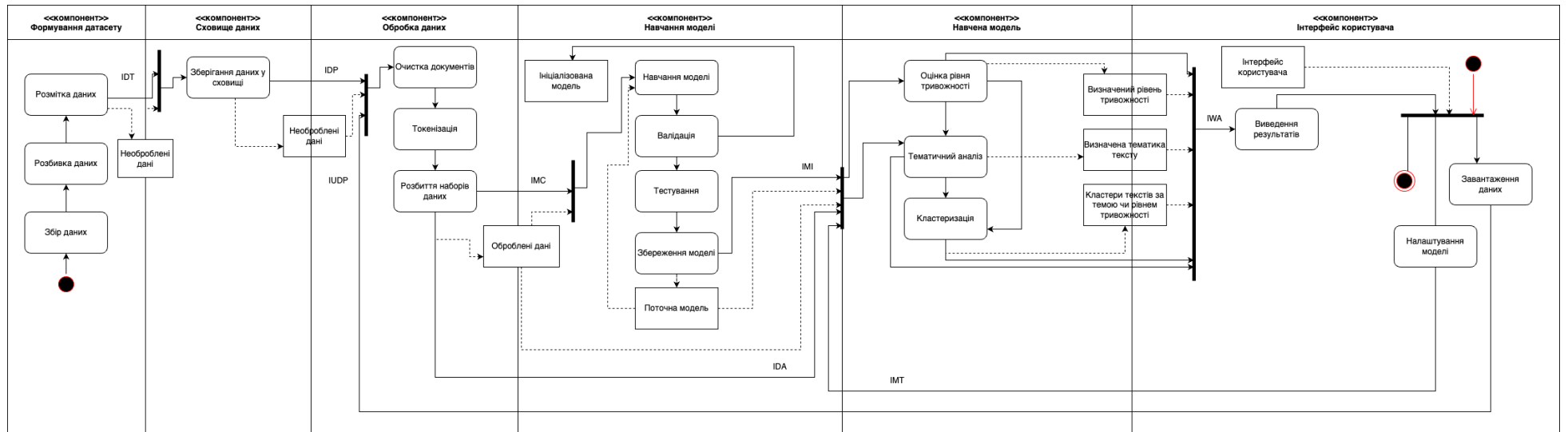


Рис. 2. Модель системи оцінювання та аналізу рівня тривожності в суспільстві. Діаграма діяльності в нотації UML

Математичне забезпечення системи, не лише визначає рівень тривожності, але і шукає його причини.

Математичні моделі реалізації компоненту «Навчання моделі»

Для визначення рівня депресії було використано декілька підходів: дерева рішень, метод опорних векторів, логістична регресія, рекурентні нейронні мережі.

Для тематичного аналізу використовуватимуться методи LSA та LDA.

Для виділення груп текстів у кластери використовуватиметься ієрархічна кластеризація та кластеризація методом KNN.

Опис датасету

Датасет взятий із відкритого сайту kaggle.com. Він містить близько 8000 розмічених твітів англійською мовою [5].

Обробка датасету

Із датасету були вилучені веб-посилання, стоп-слова та хештеги. Для виділення лінгвістичних характеристик тексту було використано програму LIWC-22. Крім цього, було додано такі ознаки:

- середня кількість символів у слові;
- частка унікальних слів;
- частка прийменників першої особи серед всіх прийменників;
- частка прийменників першої особи однини серед всіх прийменників;
- частка дієслів минулого часу серед всіх дієслів;
- частка дієслів минулого часу першої особи однини серед всіх дієслів;
- частка інфінітивів серед дієслів;
- частка слів «однак», «проте».

Результати

Результати визначення депресії представлені в Табл. 1.

Таблиця 1

Результати

Метод	Accuracy	Precision	Recall	F1 score
Логістична регресія	0.60	0.59	0.58	0.58
Дерева рішень (індекс Джині)	0.80	0.80	0.80	0.8
Метод опорних векторів	0.83	0.80	0.75	0.77
Рекурентна нейронна мережа	0.86	0.87	0.86	0.86

Висновки

1. Досліджено існуючі рішення в області аналізу рівня тривожності та депресії та встановлено, що їм бракує аналітики.
2. Запропоновано модель системи аналізу тривожності в суспільстві з визначенням причини та залежності від часу, локації людини. У подальшому така модель може застосовуватись для аналізу стану суспільства.
3. Побудовано моделі для передбачення рівня тривожності та досліджено їх основні показники. Визначено ефективні моделі для прогнозування рівня тривожності.
4. Перспективи подальших досліджень полягають у нарощуванні функціональності системи і виділенні нових лінгвістичних змінних, щоб зробити модель інформативною і не використовувати нейронні мережі.

Література

1. William, D., Suhartono, D. "Text-based depression detection on social media posts: A systematic literature review," *Procedia Computer Science*, vol.179, pp.582–589, 2021.
2. Navigerová, J. M., Naviger, J., Kučera, D., Hoffmannová, P. "Text-based detection of the risk of depression," *Frontiers in Psychology*, vol.10, 2019.
3. O’Dea, B., Boonstra, T. W., Larsen, M. E., Nguyen, T., Venkatesh, S., Christensen, H. "The relationship between linguistic expression in blog content and symptoms of depression, anxiety, and Suicidal thoughts: A longitudinal study," *PLOS ONE*, vol.16, 2021.
4. Klaisingto, P. "Linguistic indicators of negative-perspective language in women’s depressive disorder," *Humanities, Arts and Social Sciences Studies*, vol.21, pp.467-476, 2021.
5. Cho, H.K. *Twitter Depression Dataset*, 2021. [Електронний ресурс]. - Режим доступу: <https://www.kaggle.com/datasets/hyunkic/twitter-depression-dataset>

УДК 007: 658.5

Канд., техн., наук, ст., наук., співр. Маслянко П.П.,
студент Савчук І. В.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

МАТЕМАТИЧНЕ І ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ DEVOPS-СИСТЕМИ ПРОДУКУВАННЯ СИСТЕМ ТИПУ QUESTION-ANSWERING

Abstract

Pavlo Maslianko, assoc. prof., PhD; Ivan Savchuk, student.
*Mathematical and software of the DevOps-system for the production of
Question-Answering systems*

This paper concerns the task of engineering DevOps Question-Answering system based on latest insights in the field of Question-Answering. Existed approaches of system engineering were discussed. The new architecture for DevOps Question-Answering system is proposed and discussed. The analysis of system accuracy is presented. The ways for further research and upgrades are proposed as well.

Вступ

На сьогодні існує безліч різноманітних систем пошуку відповідей на запитання (Question-Answering systems, QA systems). Для забезпечення якості та швидкодії пошуку історично застосовувались методи, які ґрунтуються на порівнянні частоти входження слів які входять у запит користувача, та у документи кандидати. Однак, нещодавно з'явилися підходи які дозволяють застосовувати нейронні мережі для продукування векторного представлення документів, і використовувати методи грубої сили, такі як алгоритм пошуку найближчих сусідів, з деякими евристичними.

На даний момент, область застосування QA систем дуже широка, починаючи від інтернет-магазинів, закінчуючи великими пошуковими гігантами. Більшість систем керуються алгоритмами, які були розроблені досить давно, і ґрунтуються на статистичних показниках тексту, що не завжди враховує контекст та зміст слів у документах. Останні перспективні дослідження в області нейронних мереж трансформерів, показують значний

конкурентний потенціал, що спонукає до розробки та оновлення існуючих QA систем.

Одночасно з цим у світі активно розвиваються підходи та методи для продукування програмних продуктів. Однією з найпопулярніших, є методологія DevOps.

У цій статті ми спробуємо знайти та показати один з раціональних підходів для інженерії таких систем спираючись на методологію DevOps, а також покажемо на прикладі конкретної задачі, як найкраще продукувати QA системи для пошуку відповідей за конкретною тематикою.

Постановка задачі

Об'єкт дослідження. Класи, підкласи QA систем. Методи, моделі, системи, процеси їх продукування і підтримки за допомогою DevOps. Концепт DevOps. DevOps-система.

Предмет дослідження. Математичне та програмне забезпечення системи пошуку відповідей на запитання (QA-системи) на основі нейронних мереж трансформерів та алгоритмів пошуку найближчих сусідів.

Мета дослідження. є створення такої системи пошуку відповідей на запитання (QA-системи) за медико-біологічною тематикою, яка би могла максимально точно знаходити бажані відповіді.

Кінцевий результат. Модель DevOps системи та її імплементація для інженерії QA системи медичного напрямку. Реалізоване математичне та програмне забезпечення QA системи у вигляді застосунку. Визначені найбільш ефективні підходи пошуку текстових документів, та порівняно отримані результати зі вже існуючими системами..

Призначення. Покращення роботи систем пошуку відповідей на запитання за медико-біологічною тематикою та підвищення ефективності роботи служби підтримки подібних систем.

Для того аби регламентувати вимоги до бажаного результату, скористаємось наступним переліком бізнес-правил:

- BR1: Система має відповідати, тільки, на питання медичної тематики;
- BR2: Отримана система, має відповідати на популярні запитання перш за все amatorів, а не професіоналів;
- BR3: Довжина введеного запиту в систему не має перевищувати двісті символів;
- BR4: Пошук відповіді на запит має займати не довше п'яти секунд;

- BR5: Точність отриманої системи, має бути не нижчою за 50%, за метрикою Precision@10, а за метрикою MRR@10 (Mean Reciprocal Rank) не нижче 80%.

Теоретичні засади інженерії системи

Система відповіді на питання (Question Answering, QA система) – це система автоматизації процесів отримання бажаних відповідей на запитання, які задає людина природною мовою. Фундаментальна ідея системи QA системи – полягає в тому, щоб допомогти людино-машинній взаємодії [1].

В нашій попередній статті [2], ми провели велике дослідження не тільки DevOps як концепту, а того, що є важливим для розробки реальних програмних продуктів та систем. Матеріали статті [2], на численних прикладах - показують, що DevOps при впровадженні для вже існуючих систем призводить до покращення бажаних організаціями, метрик та результатів. З цього ми можемо зробити висновок, що для продукування сучасної та конкурентної системи ми маємо керуватися методологією DevOps.

Означення. Система DevOps - це [2] множина сутностей і відношень між ними, необхідна і достатня для забезпечення колаборації сутностей інженерії та підтримки життєдіяльності продуктів, товарів та послуг на всіх стадіях їх життєвого циклу за заданих умов і обмежень.

Розвинувши ідею вище наведеного визначення, ми можемо зробити висновок, що при продукуванні [2] системи-DevOps керуватися тим аби налагодити максимальну взаємодію між всіма компонентами системи, для забезпечення максимальної гнучкості при розробці, та швидкості реакції на події, що можуть призводити до модифікації системи аби усунути несправності, та забезпечувати конкурентну спроможність системи.

Враховуючи всі наведені особливості та вимоги до систем-DevOps, виконаємо планування власної QA системи.

Компоненти QA DevOps-системи

Опираючись на результати статті [2], а саме на представлення класів сутностей концепту DevOps у вигляді діаграми Венна, наведемо компонентне представлення QA DevOps-системи (рис. 1). На даному високому рівні, ми маємо три компоненти:

- Plan – планування та управління при розробці даної системи;
- Development – реалізація математичного та програмного забезпечення;
- Operations and Support – забезпечення взаємодії та комунікації.

За методологією DevOps, всі компоненти системи мають працювати [2] отримуючи інформацію один від одного, за допомогою чого здійснювати ефективну колаборацію.

На (рис. 1) міжкомпонентний зв'язок забезпечується за допомогою шести інтерфейсів:

- Plan-to-Dev – інтерфейс передавання інформації про задачі, вимоги, умови для створення продукту. Окрім цього слугує для виділення ресурсів;
- Dev-to-Plan – інтерфейс передавання інформації про прогрес в розробці, результати, артефакти, метрики продукту;

DevOps-система

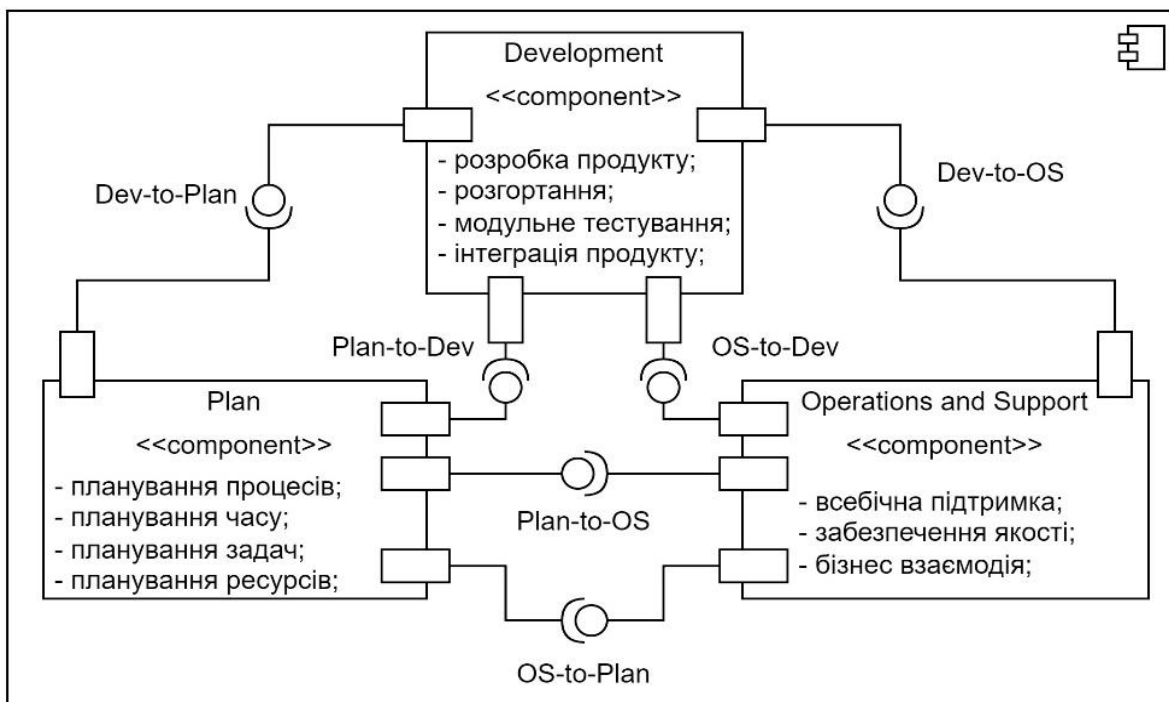


Рис. 1. Концептуальна модель QA DevOps-системи.
Діаграма компонентів у нотації UML

- Dev-to-OS – інтерфейс передавання інформації про версії продукту, забезпечення виконання задача, дотримання поставлених вимог.
- OS-to-Dev – інтерфейс передавання інформації про якість та відповідність поставленим задачам, вимогам, метрикам. Надаються коментарі.
- OS-to-Plan – інтерфейс передавання інформації про прогрес реалізації системи, виконані задачі, вимоги, та метрики для зацікавлених осіб;

- Plan-to-OS – інтерфейс уточнення та контролю можливостей, задач, вимог та метрик до системи зацікавленими особами.

Динамічне представлення DevOps показано у вигляді діаграми діяльності (рис. 2) у відповідності з переліком та внутрішньою структурою компонентної моделі (рис.1).

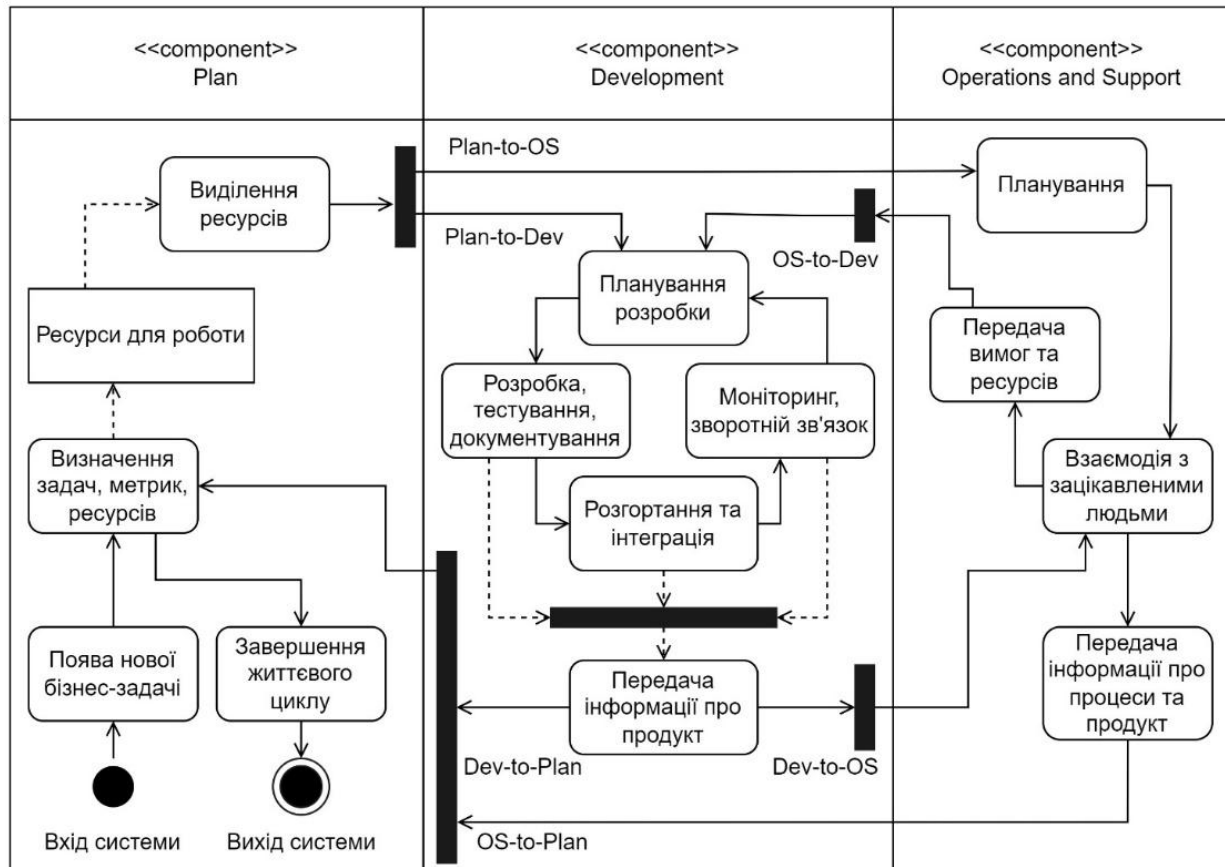


Рис. 2. Концептуальна модель QA DevOps-системи.
Діаграма діяльності у нотації UML

Таким чином концептуальні моделі представлені на (рис. 1, 2) повністю розкривають суть QA DevOps-системи на концептуальному рівні.

Далі в цій роботі, перш за все, розкриємо суть компоненту Development, оскільки, на початковому етапі, він є найбільш важливим для функціонування системи. Зробимо це на основі діаграми діяльності (рис. 3) та набору субкомпонентів.

Назви субкомпонентів та процеси в середині діаграми (рис. 3), є повністю репрезентативними. Розкриємо суть інтерфейсів зображених на діаграмі (рис. 3):

- IUQP (Interface User Query Processing) – інтерфейс забезпечення обробки запитів користувача;
- IML (Interface Model Loading) – інтерфейс завантаження найефективнішої версії алгоритмів необхідних для пошуку відповідей на запити користувачів;

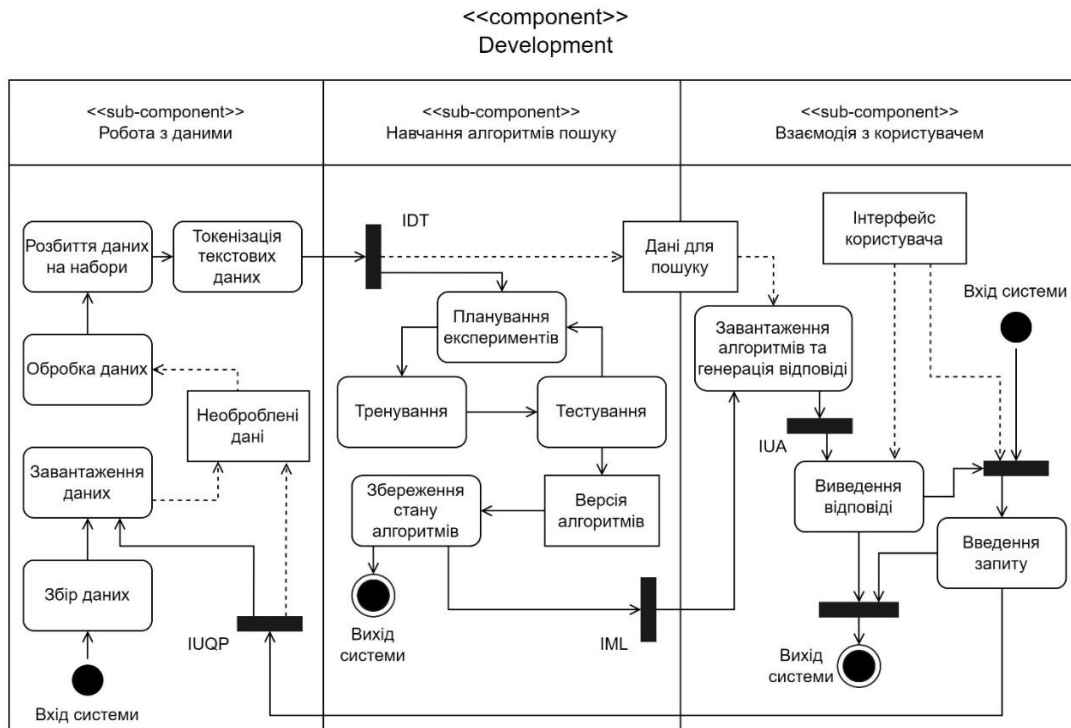


Рис. 3. Модель компонента Development.
Діаграма діяльності другого рівня в нотатції UML

- IDT (Interface Data Transfer) – інтерфейс передачі даних за допомогою яких проводиться пошук відповідей на запит користувача;
- IUA (Interface User Application) – інтерфейс передачі відповідей на виведення їх для користувача.

Математичне забезпечення навчання алгоритмів пошуку

Для реалізації пошуку в колекції документів, в даній роботі використовується метод [3] пошуку найближчих сусідів (KNN, K-Nearest Neighbors). Такий вибір обумовлений невеликим розміром колекції документів.

Для того, аби представити текстові документи було перевірено декілька типів моделей трансформерів [4], а саме моделі DistillBERT [5], MiniLM [6]. Цей тип моделей був обраний з огляду на їх невеликий розмір, та можливості їх використання в реальних системах. В деяких версіях системи модель MiniLM, також була використана для додаткового уточнення результатів пошуку використовуючи архітектуру мереж близнюків [7].

Верифікація та валідація

В результаті роботи, менш ніж за два місяці, розроблено першу версію QA системи «Medical Search». Отримана система, за результатами порівняння метрик [8] Precision@10, MRR@10 показала конкурентні результати на датасеті MEDIQA [9], та дещо перевершує результати вже існуючої системи ChiQA [10] яка також була випробувана на цьому ж датасеті, результати наведені в (табл. 1).

Набір даних MEDIQA [9] містить в собі пари документів питання-відповідь, та від початку був розбитий на три частини. Набір для тренування містить 208 запитань та 1701 відповідь до них, набір для тестування містить 150 запитань та 1107 відповідей до них, а валідаційний містить всього 25 запитань та 234 відповідей до них. Набір даних сформований англійською мовою, та містить документи лише на медичну тематику.

Таблиця 1

Результати випробування системи

Системи	Модель	Precision@10	MRR@10
Medical Search	MiniLM-L6 + KNN	0.543	0.874
	MiniLM-L12 + KNN	0.544	0.861
	DistillBERT + KNN	0.567	0.902
	DistillBERT +KNN +MiniLM-L12	0.574	0.913
ChiQA	Group of models [10]	0.517	0.895
<i>Best result on MEDIQA</i>	<i>Unknown Ensemble [9]</i>	<i>0.819</i>	<i>0.937</i>

Сама система «Medical Search» та програмний код розміщено на платформі Huggingface, і доступна за таким посиланням: <https://huggingface.co/spaces/ivan-savchuk/medical-search>. За цим же посиланням доступний як веб-застосунок, який можна використовувати для тестування системи, так і репозиторій з кодом.

Висновки

1. В роботі досліджено теоретичні підходи до інженерії та продукування QA систем. Встановлено, що для продукування QA системи на наборі MEDIQA [9], застосування системної інженерії інженерії DevOps, дає змогу отримати конкурентну систему у порівнянні з існуючими аналогами.

2. Запропоновано спосіб застосування системної інженерії DevOps-систем для продукування QA систем, що дозволяє організаціям різного виду діяльності розробляти конкурентні ІТ продукти в будь-якій предметній області.

3. В результаті порівняльного аналізу застосування DevOps-системи для продукування конкретного класу QA систем було встановлено, що розроблена система за показниками $Precision@10 = 0.574$, $MRR@10 = 0.913$ не поступається перед існуючою системою ChiQA у якої дані показники сягають всього $Precision@10 = 0.517$, $MRR@10 = 0.895$.

4. Перспективи подальших досліджень спрямовані на розробку методу системної інженерії DevOps-систем призначених для продукування ІТ-систем для будь-якої стадії їх життєвого циклу.

Література

1. Dwivedi, S.K., & Singh, V. (2013). Research and reviews in question answering system. *Procedia Technology*, 10, 417-424.
2. Маслянюк П.П., Савчук І.В. DevOps – концепт і структурне представлення. *KPI Science News*, 2021, № 4. с. 39 – 51.
3. Johnson, Jeff & Douze, Matthijs & Jégou, Hervé. (2017). Billion-Scale Similarity Search with GPUs. *IEEE Transactions on Big Data*. 7. 10.1109/TBDATA.2019.2921572
4. Vaswani, A., Shazeer, N.M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., & Polosukhin, I. (2017). Attention is All you Need. *ArXiv*, abs/1706.03762.
5. Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

6. Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., & Zhou, M. (2020). MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. ArXiv, abs/2002.10957.
7. Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. ArXiv, abs/1908.10084.
8. Zhai, C., & Massung, S. (2016). Text Data Management and Analysis: A Practical Introduction to Information Retrieval and Text Mining.
9. Abacha, A.B., Shivade, C.P., & Demner-Fushman, D. (2019). Overview of the MEDIQA 2019 Shared Task on Textual Inference, Question Entailment and Question Answering. BioNLP@ACL.
10. Demner-Fushman, Dina & Mrabet, Yassine & Ben Abacha, Asma. (2019). Consumer health information and question answering: helping consumers find answers to their health-related information needs. Journal of the American Medical Informatics Association : JAMIA. 27. 10.1093/jamia/ocz152.

Д.ф-м.н., Норкін Б.В., студентка Пашенко К.М.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ КЛАСИФІКАЦІЇ ЦИФРОВИХ ЗОБРАЖЕНЬ ТЕКСТОВИХ ДОКУМЕНТІВ

Abstract

Bohdan Norkin, Sc.D.; Kateryna M. Pashchenko, student
Mathematical and software of Document Image Classification system

This article describes the approach to solving the problem of document image classification, which is important in is an important step in office automation, digital libraries, and other document image analysis applications. Models of neural network was used to solve this problem. Information about super-resolution, structural context, information about intra-class variance and inter-class variance were taken into account to rich optimal decisions.

Вступ

Велика кількість текстових документів, таблиць, платежів створюються, обробляються та зберігаються у формі цифрових зображень. Зображення документа може бути отримане в результаті сканування, відправлення по факсу або шляхом конвертації електронного документу в зображення формату JPEG або TIFF. Процес класифікації зображень текстового документа до відповідних груп відіграє важливу роль у завданнях машинного навчання про розпізнавання тексту чи зображень, створенні цифрової бібліотеки, або вилучення інформації з документу. Вище наведені приклади завдань з документами можуть бути спрощені, якщо попередньо відомо, до якого типу документів відноситься той чи інший об'єкт дослідження. Класифікація цифрових зображень документів тісно переплітаються в сучасних проектах про розпізнавання тексту та описуються декількома парадигмами [1].

Методи класифікації цифрових зображень документів в загальному можна розділити на дві категорії - методи, що базуються на аналізі структури документу та методи, що базуються на основі аналізу вмісту документу (оптичне розпізнавання тексту (англ. optical character recognition)). Оскільки перший метод працює напряму з цифровою картинкою ділового документу, тому він і буде розглянутий в доповіді. Рішення задачі класифікації, запропонований в даній статті, побудоване на основі нейронних мереж, що дозволяє збільшити ефективність та точність класифікації.

Постановка задачі

Метою роботи є підвищення ефективності процесу обробки ділових документів в застосунках для аналізу документів шляхом розробки математичного та програмного забезпечення для класифікації цифрових зображень документів.

Предметом дослідження є засоби автоматизації класифікації цифрових зображення на основі згорткової нейронної мережі.

Задачею роботи є розробка спеціалізованої системи на основі математичного та програмного забезпечення для класифікації цифрових зображень документів.

Для вирішення поставленої задачі розглянуто моделі згорткової нейронної мережі для класифікації зображень, моделювання та проектування систем Data Science, метрики оцінки ефективності роботи алгоритму класифікації, методи web-розробки.

Вимоги до системи:

- здатність системи обробляти сирі дані та трансформувати зображення в вектор ознак сталої довжини;
- наявність алгоритму машинного навчання, що вирішує задачу класифікації цифрових зображень текстових документів;
- здатність системи класифікувати зображення за наступними класами: «лист», «меморандум», «реклама», «email», «структура папки» «бланк», «рукописний документ», «рахунок-фактура», «реклама», «бюджет», «стаття в журналі», «презентація», «наукова публікація», «анкета», «резюме», «науковий звіт» і «документ -опис»;
- розроблений програмний продукт в кінцевому вигляді повинен бути представлений у виді вікна з можливістю завантаження зображень документів;
- здатність системи проводити процедуру класифікації зображень документів з точністю не нижче 80%;
- доступ до системи надавати через web-інтерфейс.

Термінологія

Document image classification – завдання визначення односторінкового документу до одного з попередньо заданих класів документів. Класифікація базується на аналізі фізичної структури документу (англ. document layout analysis).

Document Layout Analysis – процес, що застосовується для визначення фізичної структури документу та його компонентів. Прикладом компонентів документу можуть бути основний текст, ілюстрації, зображення, таблиці.

Document class – множина документів, що характеризується подібною фізичною структурою, формою, стилем.

Convolutional Neural Network (CNN) – різновид глибоких нейронних мереж, що використовується для розпізнавання двовимірних або тривимірних даних[2].

Опис набору даних

The RVL-CDIP Dataset Набір даних RVL-CDIP (Ryerson Vision Lab Complex Document Information Processing) [3] складається з 400 000 зображень у відтінках сірого в 16 класах, по 25 000 зображень на клас. Існує 320 000 навчальних зображень, 40 000 зображень для перевірки та 40 000 тестових зображень. Зображення мають такий розмір, щоб їх найбільший розмір не перевищував 1000 пікселів. В наборі даних представлено 16 категорій: «лист», «меморандум», «реклама», «email», «структура папки» «бланк», «рукописний документ», «рахунок-фактура», «реклама», «бюджет», «стаття в журналі», «презентація», «наукова публікація», «анкета», «резюме», «науковий звіт» і «документ-опис». Вибір категорій керувався за діапазоном категорій, присутніх у Small-CDIP.

Компонентна модель системи класифікації цифрових зображень документів

Головна ідея роботи полягає в побудові та тренуванні нелінійного класифікатора, функцією якого є визначати різні структури документів та відносити зображення документу до попередньо визначеного класу. Модель системи класифікації цифрових зображень документів складається з таких компонентів (рис 1):

- блок збереження даних;
- блок обробки даних;
- блок навчання класифікатора;
- блок оцінки ефективності роботи класифікатора;
- збереження класифікатора;
- Web застосунок.

Збереження та обробка даних

Постановка задачі для моделі класифікації зображень документів охоплює оцінку простору документів та перелік можливих класів документів. Простір документів – це множина всіх документів, які подаються на вхід класифікатора для обробки, іншими словами це вхідні дані моделі. Звідси визначаються марковані екземпляри для навчання та тренування класифікатора.

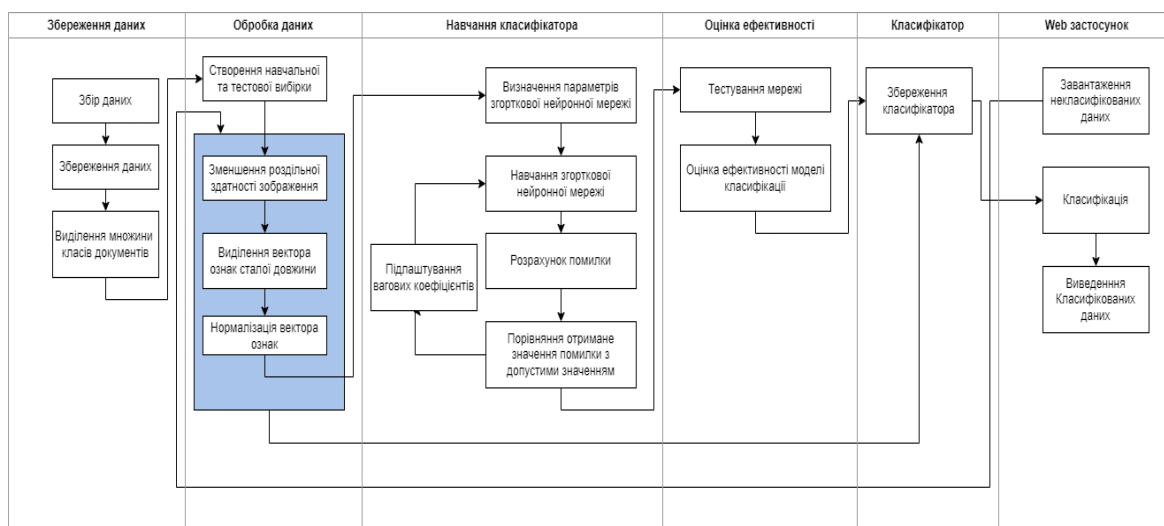


Рис. 1. Модель системи класифікації цифрових зображень документів

Множина класів зображень документів визначає як розділений простір документів, а назва класу документу є результатом роботи. Можливо декілька варіантів розділення зображень документів на класи: зображення документу належить лише одному класу або зображення документу належить відразу декільком класам. В дані роботі розглядається перший варіант розділення простору документів, оскільки це дозволяє уникнути спірних ситуацій та збільшити ефективність роботи.

В компоненті обробки даних визначається структура вхідних зображень за допомогою процесу попередньої обробки даних. Роздільність цифрових зображень документів є досить великою та може досягати 1000 пікселів. Така роздільність є громіздким вхідним параметром для згорткової нейронної мережі, та може призвести до перенавчання моделі. Оскільки модель класифікації працює з структурою документу, а не його змістом, то роздільність зображень можна зменшувати до тих пір, доки

структура самого документу ще є зрозумілою. Одним з методів для зміни дискретизації є білінійна інтерполяція[3]. Таким чином роздільна здатність зображення може бути зменшена та приведена до єдиного розміру (рис 2), щоб потім бути поданою на вхід класифікатора.

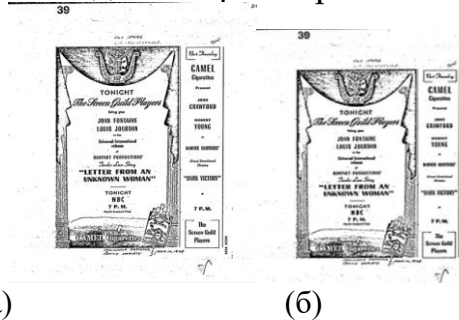


Рис. 2. (а) - початкове зображення; (б) - зображення після зменшення роздільної здатності.

Опис класифікатора

На практиці досліджено багато статичних методів класифікації зображень, наприклад метод дерева рішень, прихована марківська модель чи модель нейронної мережі. Такі класифікатори працюють з даними, що на вхід подаються як вектор ознак фіксованої довжини. Зображення текстового документу може бути стиснуте до оптимального розміру та представлено як вектор ознак фіксованої довжини.

Модель згорткової нейронної мережі, що є різновидом нейронних мереж, відзначається особливою швидкістю прийняття рішень та забезпечує високу ефективність класифікації зображень. Саме згорткову нейронну мережу можна розглянути для досягнення поставленої цілі. Навчання моделі проводиться на маркованих даних, які використовуються для ітеративного налаштування ваг. Мережа вимагає ручного налаштування топології моделі [4].

Існують додаткові програмні засоби та фрейморки, що дозволяють автоматизувати та прискорити налаштування нейронних мереж.

Фреймворк Keras – є високорівневий фреймворк, написаний на Python. Keras надає підтримку для побудови моделей глибокого навчання, включаючи нейронні мережі. Keras підтримує бібліотеки нижчого рівня, такі як TensorFlow, Theano і Microsoft Cognitive Toolkit (CNTK), що також можуть використовуватися для побудови моделей машинного навчання.

Аналіз шарів згорткової нейронної мережі

При побудові якісного класифікатора варто звернути увагу на складові згорткової нейронної мережі, оскільки це впливає на якісь та швидкість процесу навчання класифікатора. Варто виділити основні блоки згорткової нейронної мережі – вхідний шар, вихідний шар, та приховані шари. Приховані шари згорткової нейронної мережі складаються з згорткових шарів, об'єднання шарів, повністю пов'язаних шарів і шарів нормалізації (ReLU) [5].

Згортковий шар є основним компонентом згорткової нейронної мережі, який складається з набору ядер. В цьому шарі вхідні дані подаються на вхід до згорткової функції, яка повертає матрицю скалярних добуток між вхідними даними та параметрами фільтрів(1).

$$z^l = h^{l-1} \times W^l \quad (1)$$

Після проходження даного шару виділяється саме високорівневих характеристик, що є вхідним параметром наступного шару (рис 3).

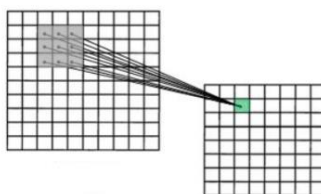


Рис. 3. Графічне представлення роботи згорткового шару

Результат подається на вхід до шару агрегування, де відбувається процес зменшення кількості параметрів. Даний процес не лише сприяє зменшенню об'єму обчислень мережі, а й попереджає перенавчання. Як приклад процесу є максимальне агрегування (2), що відокремлює максимальне значення в підмножині нейронів попереднього шару.

$$h_{x,y}^l = \max_{i,j} \{h_{x+1,y+1}^{l-1}\}, i = 0, \dots, s, j = 0, \dots, s \quad (2)$$

Для посилення ефективності роботи мережі в класифікації зображень запропоновано використати шар зрізаних лінійних вузлів (англ. Rectified Linear Units layer), де застосовується передавальна функція активації ReLU(3).

$$f(x) = \max(0, x) \quad (3)$$

Саме функція ReLU вказує до якого класу належить подане на вхід зображення документу. На відміну від звичайної сигмоїдальної функції чи гіперболічного тангенсу, ReLU дозволяє навчити згорткову мережу у декілька разів швидше та підвищити нелінійні властивості шару. За приклад структури згорткової нейронної мережі можна розглянути схему на рис.4.



Рис. 4. Приклад архітектури згорткової нейронної мережі

Методи оцінки ефективності роботи класифікатора

Невід'ємна частина класифікатора є процес оцінки ефективності, так як він показує залежність ефективності моделі від вхідних параметрів, стандартизованих чи нестандартизованих, кількості класів дослідження, топології моделі та інших характеристик. В роботі проводиться дослідження класифікатора, що працює з множиною класів, то запропоновано метрики для оцінки багатокласової моделі.

Метриками для визначення якості отриманого результату:

- точність (precision);
- повнота (recall);
- F-міра (F-score).

Значення вказаних вище метрик розраховується за допомогою параметрів true positive (TP), true negative (TN), false positive (FP) і false negative (FN). В даному випадку вище вказані метрики, разом з матрицею помилок буде представлена для багатокласового класифікатора.

Висновки

1. В ході роботи було досліджено актуальні методи та застосунки класифікації зображень текстових документів та виявлено можливі сфери застосування запропонованої системи; виокремлено методи класифікації текстових документів на основі фізичної структури документу.
2. Представлено вимоги до запропонованої моделі класифікації, оцінено простір вхідних даних та типи можливих класів. Виокремлено основні компоненти цілісної системи класифікації зображень документів. Розглянуто методи попередньої обробки даних, зокрема методи зменшення роздільної здатності зображення.
3. Запропоновано модель системи класифікації зображень документів, що базується на розробці та налаштуванні згорткової нейронної мережі. Розглянуто метрики для оцінки адекватності системи багатокласової класифікації.

Література

1. Nawei C. A survey of document image classification: problem statement, classifier architecture and performance evaluation [Електронний ресурс] / C. Nawei, B. Dorothea // Springer-Verlag 2006. – 2006. – Режим доступу до ресурсу: <https://www2.cs.sfu.ca/CourseCentral/414/li/material/refs/Document-survey-07.pdf>.
2. James Atwood and Don Towsley. Search-Convolutional Neural Networks. CoRR, abs/1511.02136, 2015.
3. RVL-CDIP-I Dataset [Електронний ресурс] – Режим доступу до ресурсу: <https://www.kaggle.com/datasets/nbhativp/first-half-training>.
4. D. C. Cirean, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber. Flexible, high performance convolutional neural networks for image classification. In International Joint Conference on Artificial Intelligence, pages 1237–1242, 2011.
5. D. C. Cirean, U. Meier, L. M. Gambardella, and J. Schmidhuber. Convolutional neural network committees for handwritten character classification. In International Conference on Document Analysis and Recognition, pages 1250–1254.

К.т.н., доцент Олєфір О.С., студент Буловацький М.С.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ПРИЙНЯТТЯ РІШЕНЬ У КОАЛІЦІЙНІЙ ГРІ З НЕПОВНОЮ ІНФОРМАЦІЄЮ

Abstract

Oleksandr S. Olefir, assoc. prof., PhD; Maksym Bulovatskyi, student
Mathematical and software of decision making system in cooperative games with incomplete information

This paper concerns the problem of creating autonomous agents for cooperative games with incomplete information. Computer players surpass humans in games of different genres, starting with strategies such as chess and GO, and including more dynamic one as Dota2 and Starcraft. However games that require negotiations and cooperating still cause problems for computer agents. Ways to overcome this frontier are proposed in that paper.

Вступ

Стратегічні комп'ютерні та настільні ігри є не лише сферою розваг, а й цікавим перетином теорії прийняття рішень із моделюванням економічних перемовин та кооперації автономних агентів.

Комп'ютерні агенти вже подолали такі рубежі як стародавні шахи, більш стратегічно глибоку Го, сучасні та динамічні Dota 2 та Starcraft на рівні перевищуючи людські можливості у швидкості реакції та далекоглядності планування.

Жанр в якому комп'ютерна програма досі не може наздогнати людину відносяться до соціальної взаємодії - вміння кооперуватися, знаходити компроміс, брехати чи блефувати.

У цій статті розглядаються моделі, алгоритми та підходи до архітектури, для реалізації системи автономних агентів на прикладі коаліційної гри з неповною інформацією Diplomacy, використовуючи синтез методів прийняття рішень та моделей перемовин у сфері економіки.

Постановка задачі

Метою роботи є проектування та створення системи прийняття рішень, що може у якості автономного агента взаємодіяти із іншими агентами або з людиною, у стратегічній грі з акцентом на перемовини та дипломатію.

Предметом дослідження є гібридні ігри, ігри з неповною інформацією; моделі перемовин, а саме моделі довіри, репутації, аргументації; методи навчання з підкріпленням.

Задачею є розробка математичного забезпечення системи прийняття рішень у іграх, що можуть містити кооперацію, неповну інформацію, асиметричність. Та розробка програмного забезпечення, у якому можна перевірити отриману систему на прикладі гри Diplomacy.

Термінологія

Потік перемовин - це послідовність пропозицій та контрпропозицій між агентами.

Тактика - лінійна комбінація простих функцій, на основі яких генеруються пропозиції та контрпропозиції. Тактика генерує пропозицію на окремий компонент об'єкту перемовин (тобто проблему що обговорюється - наприклад ціна, якість, кількість товару), засновуючись на одному критерії - часу до дедлайну, ресурсів що залишилися, попередній поведінці опонента, тощо. Якщо необхідно враховувати різні критерії - застосовується лінійна комбінація різних тактик із вагами залежно від важливості критерію.

Стратегія - функція зміни ваги тактик від часу чи ресурсів.

Компонентна модель системи

Головна ідея роботи полягає в побудові системи для тренування автономних агентів, функцією яких є вибір стратегій поведінки та прийняття рішень у перемовинах з іншими агентами або людиною. Модель системи класифікації цифрових зображень документів складається з таких компонентів (рис 1):

1. Блок вхідних даних;
2. Блок взаємодії гравців;
3. Система прийняття рішень;
4. Навчання агентів;
5. Візуалізація (інтерпретація) результатів.

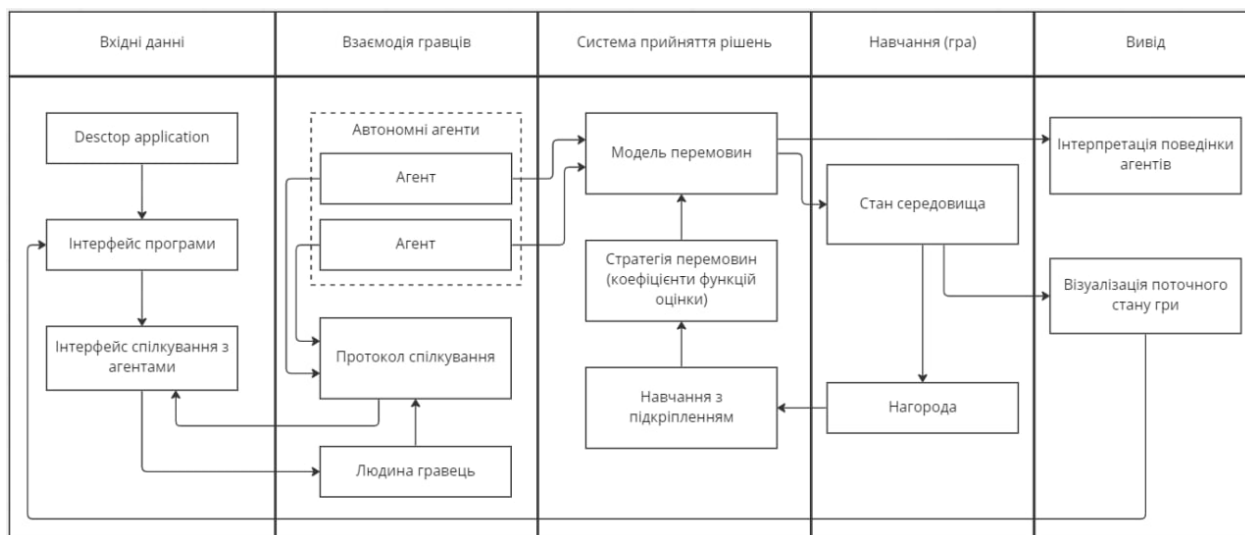


Рис 1. Структура системи

Протокол спілкування агентів

Гра Diplomasy, на якій пропонується досліджувати дипломатичні здібності агентів вже була запропонована у якості тестового майданчика для подібних задач у публікації [1], і хоча не вдалося знайти задовільні результати цих випробувань, у статті повною мірою розкриваються питання архітектури розглядуваної системи. Протоколи комунікації розбито на багато рівнів від низькорівневих - прикладних, до більш високорівневих - абстрактних:

1. Обговорення пропозицій;
2. Обмін прямою інформацією (щодо ігрових дій);
3. Обмін непрямою інформацією (обмін предикатами пункту 2);
4. Емоційний контекст повідомлень;
5. Врахування часу;
6. Аргументація;
7. Спростування.

Подібна архітектура дозволяє експериментувати з різним рівнем абстракції у виборі стратегій, від класичних алгоритмів перебору можливих ходів (min-max пошук, альфа-бета відсіч) до алгоритмів досягнення більш абстрактних цілей (дерева Монте-Карло, алгоритми deep learning, чий результати важче інтерпретувати), і потенційно навіть до алгоритмів, що використовують психологічний портрет опонентів при прийнятті рішень (актуально для дипломатичних чи бізнес переговорів).

Опис моделі

Оскільки у грі присутні багато сторін із власними інтересами, використовується мульти-агентна, багато-проблемна модель[2].

Нехай $i \in \{a, b\}$ це агент (учасник перемовин).

$j \in \{1, \dots, n\}$ перелік проблем що обговорюються. Їх набір як і в реальному світі завжди скінченний.

Нехай $x_j \in [\min_j^i, \max_j^i]$ це значення для проблеми j прийнятний для агента i . Для спрощення розрахунків кожен агент має функцію нормалізації $V_j^i: [\min_j^i, \max_j^i] \rightarrow [0,1]$.

Крім цього, агент присвоює кожній проблемі вагу w залежно від її важливості. $\sum_{j=1}^n w_j^i = 1, \text{ for } \forall i$. В результаті чого отримуємо зважену

оцінку пропозиції $x = (x_1, \dots, x_n)$:

$$V^i(x) = \sum_{j=1}^n w_j^i V_j^i(x_j) \quad (1)$$

Потік перемовин між агентами, в момент часу t_n , що записується як $X^{t_n}_{a \leftrightarrow b}$, це скінченна послідовність довжини n пропозицій та контрпропозицій $(x^{t_1}_{a \rightarrow b}, x^{t_2}_{b \rightarrow a}, x^{t_3}_{a \rightarrow b}, \dots)$, де $t_1 \leq \dots \leq t_n$.

Агенти пропонують лише прийнятні для них пропозиції:

$$\forall j, x^i_{a \rightarrow b}[j] \in [\min_j^a, \max_j^a], x^{i+1}_{b \rightarrow a}[j] \in [\min_j^b, \max_j^b]$$

Останній елемент потоку може бути завершаючою дією: прийняти або відхилити передостанню пропозицію. Потік вважається активним, якщо останній елемент

$$(x^{t_n}_{a \leftrightarrow b}) \notin \{\text{прийняти, відхилити}\}.$$

Коли агент a отримує пропозицію від агента b в момент часу t , він оцінює його згідно своєї функції оцінки (1). Якщо оцінка пропозиції є кращою, за оцінку контрпропозиції, що агент a збирався виставити на наступному кроці - вона приймається, в іншому випадку висувається контрпропозиція, та потік продовжується до закінчення виділеного на перемовини часу t_{max} .

Запишемо цей алгоритм більш формально. $I^a(t^{i+1}, x^t_{b \rightarrow a})$ дія агенту

- Якщо $t_{i+1} > t_{max}$ - відхилити пропозицію;
- Якщо $V^a(x^t_{b \rightarrow a}) \geq V^a(x^{t_{i+1}}_{a \rightarrow b})$ - прийняти;
- В інших випадках - $x^{t_{i+1}}_{a \rightarrow b}$.

Тактики в свою чергу можна поділити на різні класи, основні з яких: залежні від часу, залежні від ресурсів, залежні від попередньої поведінки опонента.

Дослідження останнього класу особливо цікаве в контексті ігор, залежних від таких навичок як блеф, брехня, довіра, тощо, що відсилає нас

до моделей довіри [3] та репутації [4]. Дані моделі окрім сухого обміну пропозиціями також додають предикати обіцянки щодо майбутніх пропозицій, обміну між двома агентами інформації щодо третього агента, емоційного окрасу реакції агента на пропозицію, тощо.

Наприклад таку характеристику як честь пропонується інтерпретувати як відносну ентропію того, що агенти очікують один від одного, залежно від того яка доля їх обіцянок в минулому виконувалась.

Стратегії перемовин

Зважена лінійна комбінація різних тактик дозволяє керувати пріоритетами агента для досягнення цілі, яка задається глобальною стратегією, що в свою чергу може змінювати вагу тактик із плином перемовин.

Вважаємо що в момент t агент a має інформацію про середовище (ресурси, час), свої переконання щодо інших агентів та висновки з попередніх взаємодій із ними (моделі довіри та репутації), інші атрибути (свої цілі, обмеження, зобов'язання, тощо), і все це представлено деяким станом, яким в літературі прийнято називати ментальним станом (MS_a^t) на області значень $D = D_1 \times \dots \times D_p$. У зваженій композиції з m різних тактик, кожна окрема вага залежить від MS^t , тобто ваги можна представити матрицею з m на p елементів.

Оптимізацію цієї матриці пропонується виконувати алгоритмами навчання з підкріпленням, зокрема ProximalPolicyOptimization (PPO), який показав свою ефективність при швидкодії, що дозволяє використовувати його у іграх в реальному часі.

Висновки

1. Було розглянуто алгоритми з різних сфер, синтез яких може допомогти у проектуванні автономних агентів для коаліційної гри, проаналізовано архітектуру для їх програмної реалізації.
2. Запропоновано використовувати мультиагентну, багато-проблемну модель, дотримання якої двома учасниками перемовин дає можливість знаходження оптимального контракту.
3. Для коригування вагів довгострокової стратегії пропонується використовувати ProximalPolicyOptimization алгоритм навчання з підкріпленням, ефективний у задачах реального часу.
4. У подальшому передбачається додати протокол спілкування людського гравця з комп'ютерним опонентом для виконання свого роду теста Тьюрінга з навченими агентами.

Література

1. *A.Fabregues*. DipGame: A challenging negotiation testbed. // *A.Fabregues, C.Sierra, /Engineering Applications of Artificial Intelligence, –2011–P.1137–1146*
2. *Faratin P*. Negotiation decision functions for autonomous agents / *C. Sierra, N.Jennings* // *Robotics and Autonomous System. –1998. – P.159–182.*
3. *C.Sierra*, Trust and honour in information-based agency. / *C. Sierra, J.K.Debenham, //The 5th International Joint Conference on Autonomous Agents and Multi-agent Systems, –2006.– P.1225–1232.*
4. *I.Pinyol*, Pragmatic strategic reputation based decisions in multiagents. // *J.Sabater Mir, I.Pinyol /The 8th International Joint Conference on Autonomous Agents and Multiagent Systems, – 2009 – P. 1001–1008.*

УДК 519.688:004.89

К.т.н., доцент Олефір О.С., студент Галюк Є.І.,
асистент Ковальчук-Хіміук Л.О.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

ВПЛИВ ВЕЛИЧИНИ ПОПУЛЯЦІЇ ПРИ ВИКОРИСТАННІ МАШИННОГО НАВЧАННЯ З ПІДКРІПЛЕННЯМ НА ПОКАЗНИКИ ШВИДКОСТІ НАВЧАННЯ

Abstract

Oleksandr S. Olefir, assoc. prof., Ph.D; Yevhenii Haliuk, student;
Kovalchuk-Khymiuk, assistant.

Influence of reinforcement machine learning population size on learning rate

This article is about the problem of real-time decision-making using a reinforcement learning type neural network. This article evaluates the influence of the population on the learning rate, as well as the number of populations to achieve the goal.

Вступ

Навчання з підкріплення є одним із різновидів машинного навчання. Агент потрапляє у середовище, в якому потрібно приймати певні рішення для досягнення найкращого результату. Таке навчання спирається на виконанні дій, для отримання за них позитивних чи негативних відгуків для агента, а також передачі найкращих результатів новим поколінням. При цьому в агентів розвивається так звана “інтуїція”, яка допомагає як можна скоріше реагувати на отримані з навколишнього середовища чинників, та обирати правильну послідовність дій у тій чи іншій ситуації.

Одним із найкращих прикладів такої інтуїції у реальному житті є водій автомобіля. Оскільки визначається що час середньої реакції водія на зовнішні чинники при кермуванні автомобільним засобом являється одна секунда. То на великих швидкостях, особливо при пересуванні за містом її може бути недостатньо і долі секунди можуть допомогти водію уникнути дорожньо транспортної пригоди. Але, така реакція набувається не одразу, а через певний час, так само, як і у випадку навчання з підкріпленням, де агент вчиться реагувати на зовнішні чинники і час реакції останнього краще ніж у людини.

Постановка задачі

Метою дослідження є порівняння впливу величини популяції при використанні нейронної мережі на базі навчання з підкріпленням для невеликого набору даних.

Предметом дослідження є величина популяції при використанні навчання з підкріпленням для невеликого набору вхідних даних.

Задачею є порівняння витраченого часу на кожну популяцію, а також кількість необхідних популяцій для досягнення результату при різній величині популяцій.

Для розв'язку поставленої задачі використовувалися методи: теорії системного аналізу, системної інженерії, моделювання та проектування систем Data Science, Q-learning алгоритм машинного навчання, NEAT алгоритм машинного навчання, візуалізації даних. Для порівняння роботи алгоритмів використовують програмне забезпечення на основі системи автоматичного управління транспортом в умовах згенерованого двох вимірною зображення.

Вимоги до системи:

- система визначає границі дорожнього покриття;
- система керує транспортним засобом при максимально дозволений швидкості;
- система виводить на екран результати пересування транспортного засобу у 2D форматі;

Опис набору даних

На кожному кроці, тобто при кожному оновленні зображення результатів, алгоритм отримує дані з п'яти радарів, які рахують дальність від агента до краю полоси треку, при цьому радари зчитують дані, починаючи від -90 градусів, до 90 градусів із кроком 45 градусів. Це досить схоже на радіус огляду дорожньої ситуації керманіча при управлінні транспортним засобом. Також, додатково, алгоритм отримує поточну швидкість. Йому необхідно повернути коефіцієнти, які своєю чергою впливають на набір вжитих дій: напрям кермового управління (від -1 до 1), тобто зміна напрямку руху, а також зміну для коректування швидкості (від -1 до 1), тобто зменшення чи збільшення поточної швидкості. Ця дія потім передається в модуль обробки даних, який обробляє вхідні дані, і приймає рішення про зміщення агента у заданому напрямі та цикл повторюється [1].

Опис алгоритму

Для реалізації поставленої задачі використовується підготовлене віртуальне оточення, яке розроблено за допомогою бібліотеки pygame для

зору агента, Точно так, як відбувається при використанні водієм автомобіля, він бачить лише частину проїзної частини перед собою. Останнє ж значення це - швидкість автомобіля. На основі даних значень, нейронна мережа передає відповідний масив значень зі своїми коефіцієнтами, які своєю чергою обробляються програмним засобом, та впливають на подальші дії агента у віртуальному оточенні. В програму додано лічильники проведеного часу на навчання, перевірка на проходження повного кола підготовленого треку, та лічильник кількості популяцій, який дозволяє зробити висновки щодо доцільності використання різної кількості агентів для проведення навчання.

Результати застосування алгоритму

Для порівняння швидкості навчання, підготуємо декілька величин популяцій для алгоритму, починаючи з п'ятдесяти агентів, закінчуючи п'ятьма тисячами агентів в одній популяції. Метриками порівняння будуть виступати: кількість необхідних популяцій для досягнення мети, а також час затрачений на одну популяцію [6,7].

Таблиця 1

Порівняння величин популяцій

Величина популяції	Кількість популяцій для досягнення мети	Середній затрачений час на одну версію популяції (с)
50	12±4	2
100	8±2	4
250	6±1	10
500	4±2	20
1000	2±1	45
2500	3±1	100
5000	4±1	180

Висновки

1. Було досліджено вплив величини популяцій на швидкість навчання алгоритму нейронної мережі на базі навчання з підкріплення.

2. Встановлено, що збільшення величини популяції зменшує кількість необхідних популяцій для досягнення кінцевої мети. При цьому дуже великі обсяги популяції на невеликому наборі даних, який складається з п'яти векторів відстаней та швидкості конкретного агента, впливають негативно на результат, так як кількість необхідних популяцій, а отже і час на навчання збільшується, а оптимальність результату зменшується. Так, порушується стабільність навчання, а отже і оптимальність затраченого на навчання часу.
3. Використання невеликої чисельності популяцій тягне за собою збільшення необхідної кількості популяцій для досягнення мети. В той же час, при підрахуванні затраченого часу на кожен з результатів, можна зробити висновок, що менша чисельність популяції, хоча і потребує більшої кількості популяцій для навчання, займає менше часу на навчання.
4. Використання невеликих популяцій для невеликого набору вхідних даних як видно з таблиці 1 є оптимальним вибором для зменшення затраченого часу, а отже і машинних ресурсів для досягнення мети.
5. В подальшому планується використання невеликих популяцій агентів для використання методу навчання з підкріпленням.

Література

1. *Грессер Л., Кенг Ст.* Глибоке навчання з підкріпленням: теорія і практика на мові Python, 2022. – С. 25-81.
2. *Lott S.* Functional Python Programming / S. Lott; Packt Publishing, 2015. – С. 24-87.
3. *Omelianenko I.* Hands-On Neuroevolution with Python/ Omelianenko I; Packt Publishing, 2019. – С. 38-96.
4. *Beazley D.* Python Cookbook / D. Beazley, В. К. Jones – 3rd edition. – O`Reilly Media, 2013. – С. 217-329.
5. *Равичандиран С.* Глибоке навчання з підкріпленням на Python. OpenAI Gym і TensorFlow для профі, 2019. – С. 153-215.
6. *Максим Ланан.* Глибоке навчання з підкріпленням. AlphaGO і інші технології, 2020. – С. 133-166.
7. *Андреа Лонца.* - Алгоритм навчання з підкріпленням, 2020. – С. 75-93

УДК 519.688

К. т. н, доцент Олефір О.С., студент Козак А.В.,
асистент Ковальчук-Хімюк Л.О.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ КЛАСИФІКАЦІЇ КОМБІНАЦІЙ КАРТ ТА ОЦІНКИ ЙМОВІРНОСТІ ВИГРАШУ

Abstract

Oleksandr Olefir, assoc. prof., PhD; ArtemKozak, student;
Kovalchuk-Khymiuk, assistant.

Mathematical and software system for classification of card combinations and estimation of probability of winning

This paper concerns the task of classification with artificial intelligence of card combinations when playing "Texas Holdem" game. The existed classification services and methods is studied and discussed. The new architecture for predictive classification system is proposed and described. An analysis of the efficiency and accuracy of the system is presented. The ways for further research and upgrades are proposed as well.

Вступ

Світ сповнений задач, пов'язаних із взаємодією між декількома агентами. Зараз комп'ютерні агенти вирішують безліч завдань: від простих і безневинних як автоматичні телефонні системи, до критично важливих як управління безпекою і навіть управління автономним транспортом.

Проте такі агенти не можуть функціонувати без належних знань. Для їх отримання, таких агентів зазвичай тренують на вибірці, для якої вже відомий результат. Для успішного «навчання» агенту необхідний великий об'єм даних [1]. Тут і постає питання як можна отримати вибірку таких розмірів вже з відомим результатом для подальшої обробки.

У статті розглянута задача класифікації карт в грі «Техаський Холдем», що є однією з найбільш поширених типів гри «Покер». У грі перемагає той гравець, що має старшу комбінацію карт. Оскільки від точності класифікації залежить перемога гравця, то цьому було приділено максимальну увагу.

Задача класифікації набору карт за комбінаціями є досить складною оскільки вхідний набір має дві властивості, які роблять його особливо складним для алгоритмів класифікації – цевсі категоріальні ознаки та

надзвичайна незбалансованість набору. Категоріальні ознаки складні, тому що типові показники відстані не можуть бути природно застосовані до таких ознак як, наприклад, ранг і масть – обчислення евклідової відстані між «піками» та «червами» просто не має сенсу. Незбалансовані набори даних є складними, оскільки алгоритми машинного навчання припускають гарний баланс у вибірці [2].

У статті запропоновано рішення на основі нейронних мереж [3], що дозволяє суттєво підвищити точність класифікації та відрізняє його від доступних алгоритмів найвищою точністю класифікації.

Постановка задачі

Метою дослідження є класифікація комбінацій карт у грі «Техаський холдем» з досягненням абсолютної точності даної класифікації та оцінка ймовірності виграшу при кожній з комбінацій.

Предметом дослідження є математичне та програмне забезпечення системи класифікації комбінацій карт в грі «Техаський холдем» та оцінка ймовірності виграшу на основі нейронних мереж та попередньої обробки даних.

Задачею є розробка спеціалізованої системи із використанням математичного та програмного забезпечення для обробки, класифікації та подальшого аналізу комбінацій карт у грі «Техаський холдем» для оцінки ймовірності перемоги у грі при кожній з комбінацій.

Для розв'язку поставленої задачі використовувалися методи: теорії системного аналізу, системної інженерії, моделювання та проектування систем Data Science, класифікації, машинного навчання та візуалізації даних.

Вимоги до системи:

- система визначає клас відгуку, у проміжку від 0 до 9 включно;
- система має класифікувати набори гральних карт по комбінаціям;
- система надає абсолютну точність при класифікації комбінацій карт;
- система повинна бути здатна обробляти великі об'єми даних з розміром вибірки більшим за 10 мільйонів комбінацій;
- доступ до системи надається через запуск файлу з кодом;
- отримані результати записуються у систему.

Термінологія

Multilayerperceptron (MLP) - окремий випадок перцептронун Розенблата [4], в якому один алгоритм зворотного розповсюдження помилки навчає всі шари. Назва з історичних причин не відображає особливості даного виду перцептронун, тобто не пов'язана з тим, що в ньому є кілька шарів.

Особливістю є наявність більш ніж одного шару, що навчається (як правило — два або три).

Компонентна модель системи

Архітектура системи має відповідати поставленим вимогам та мати необхідні властивості для забезпечення класифікації комбінацій карт та подальшого їх аналізу для оцінки ймовірності виграшу. Задачею системи є досягнення абсолютної точності при класифікації комбінацій та отримання ймовірностей перемоги в грі при подальшому аналізі даних.

Компонентна модель системи класифікації комбінацій карт та оцінки ймовірності виграшу, складається з таких компонентів (рис.1): сховище навчальних даних, обробка числових даних, навчання моделі класифікації комбінацій карт, класифікатор, моделювання гри «Техаський Холдем» та аналіз отриманих результатів.

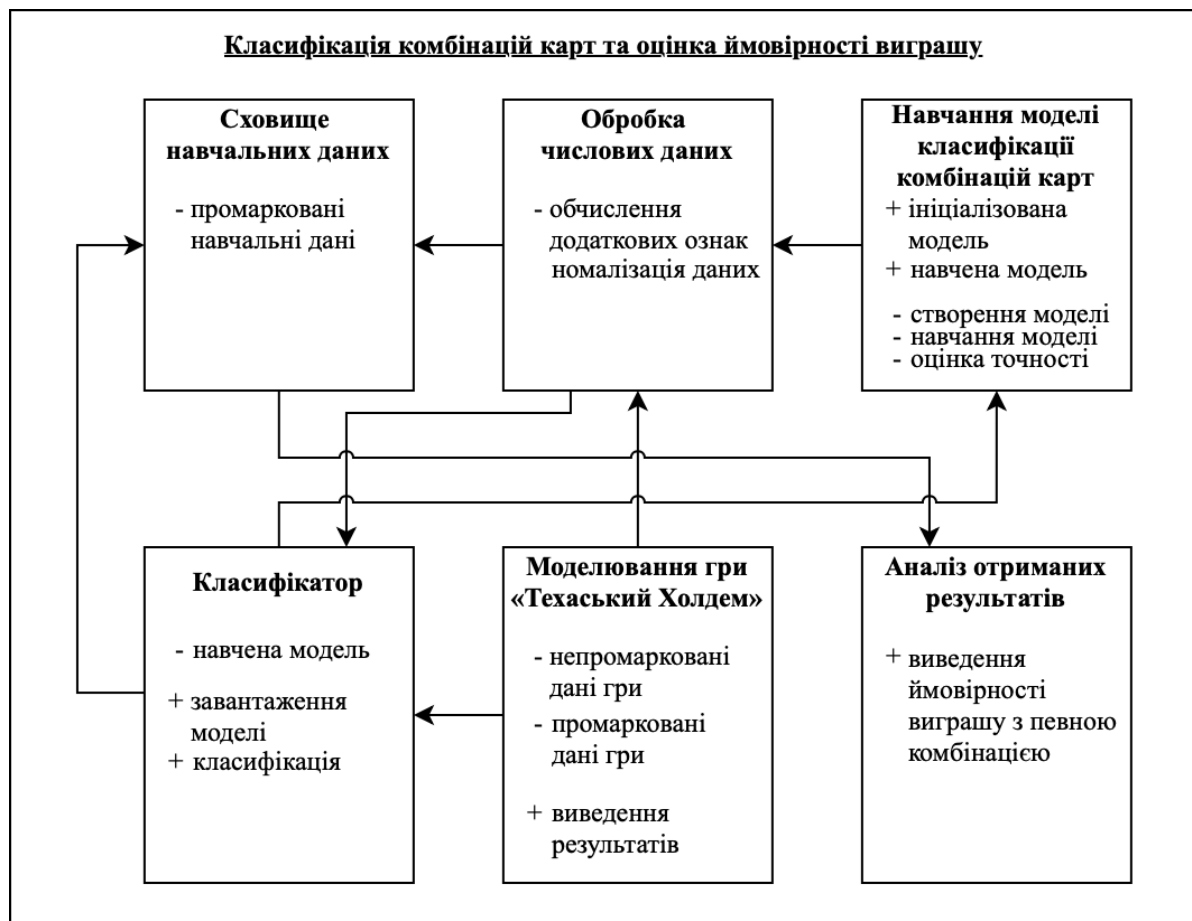


Рис. 1. Модель системи класифікації комбінацій карт та оцінки ймовірності виграшу

Опис набору даних

Для того, щоб натренувати модель розпізнавання комбінацій карт, необхідна вибірка з достатнім об'ємом даних. Для цього було розроблено алгоритм, що аналітичними методами визначає порядковий номер комбінації. Після реалізації даного алгоритму, можна згенерувати достатню кількість комбінацій карт для подальшого тренування моделі. Таким чином було згенеровано 6 мільйонів комбінацій карт з визначеним класом комбінації.

Кожна комбінація має вигляд

$[S1, C1, S2, C2, S3, C3, S4, C4, S5, C5, S6, C6, S7, C7, CLASS]$,

де $S1, S2 \dots, S7$ – порядкові номери масті карти, $C1, C2, \dots, C7$ – порядкові номери рангів карти, $CLASS$ – порядковий номер комбінації карт.

Цей набір даних надзвичайно незбалансований, адже перші три класи представляють собою близько 70% зразків як у навчальному, так і в тестовому наборі. Незбалансовані класифікації становлять проблему для прогнозного моделювання, оскільки більшість алгоритмів машинного навчання, що використовуються для класифікації, розроблено на основі припущення про однакову кількість прикладів для кожного класу.

Класифікатор, який навчиться правильно класифікувати ці три класи, але повністю не класифікує інші класи, все одно досягне приблизно 70% точності. Причина, по якій класифікатор усе ще отримує хорошу оцінку, полягає в тому, що враховується дисбаланс класів, тобто правильним прогнозам домінуючих класів надається вага, пропорційна кількості зразків. Низька точність прогнозування для нечастого класу затьмарюється кращими прогнозами з тих класів, де є багато зразків для навчання.

Для вирішення даної проблеми дисбалансу, було розроблено алгоритм. За ним, для кожної комбінації обраховано кількість однакових ознак в кожному наборі. Для цього створимо 4 нові значення ($suit1, suit2, \dots, suit4$) та за умовчанням присвоїмо їм значення 0. Дані значення будуть відповідати за кількість карт кожної з чотирьох мастей відповідно. Наступним кроком перебираючи карти $S1-S7$ будемо додавати по 1 до значення $suit1, suit2, \dots, suit4$ відповідно. Аналогічні дії виконуємо для підрахунку однакових номерів карт $C1-C7$ та занесемо їх у значення змінних $rank1, rank2, \dots, rank13$.

Крім того, розширено набір вхідних даних обраховуванням різниці між значеннями сусідніх карт по формулі:

$rank2 - rank1, rank3 - rank2, \dots, rank13 - rank12, rank13 - rank1$

Після того, як було отримано необхідний формат всіх вхідних, необхідно звести їх до єдиного виду. Для усіх ознак, що мають неперервний тип даних, необхідно зробити нормування.

Після всіх розрахунків в результаті отримано набір даних розміром 6 мільйонів рядків з 49 ознаками в кожному.

Опис алгоритму

При побудові алгоритму розпізнавання використаємо багатошаровий перцептрон з кількістю схованих шарів, що складає 150 на 150.

Першим кроком необхідно розширити та нормалізувати вхідну вибірку. Для цього обраховано кількість однакових мастей та рангів в кожному наборі карт відповідно. Далі обраховуємо різницю між значеннями сусідніх карт та стандартизуємо вибірку.

Потім відбувається тренування моделі на 4,800,000 комбінаціях, при цьому модель не перетреноується через встановлене обмеження на 10 ітерацій.

Наступним кроком оцінюємо точність прогнозування та за недосягнення 100% частково дотреноуємо модель.

Таким чином після всіх циклів тренування моделі, багатошаровий перцептрон на виході має точність 100%. Результат класифікації з вхідною вибіркою розміром у 6 мільйонів комбінацій карт (рис. 2).

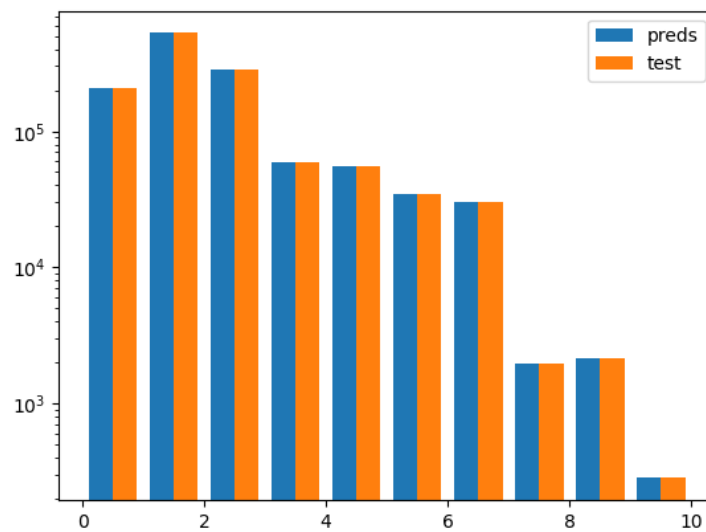


Рис. 2. Гістограма розподілу прогнозованих та справжніх класів комбінацій

Будуємо матрицю невідповідностей спрогнозованих та справжніх значень з тестової вибірки. (табл.1). З даної таблиці видно, що всі значення лежать на головній діагоналі, а всі інші паралелі до неї містять нульові значення. Це означає, що 1,200,000 комбінацій було правильно ідентифіковано та розподілено по відповідним класам комбінацій. Також добре помітно незбалансованість даного набору – 525397 комбінацій першого класу та лише 238 комбінацій дев'ятого.

Таблиця 1

Матриця невідповідностей прогнозованих значень до справжніх

Класкомбі-нації		Дійсні значення									
		0	1	2	3	4	5	6	7	8	9
Прогнозовані значення	0	20925 4	0	0	0	0	0	0	0	0	0
	1	0	52539 7	0	0	0	0	0	0	0	0
	2	0	0	28221 5	0	0	0	0	0	0	0
	3	0	0	0	5861 5	0	0	0	0	0	0
	4	0	0	0	0	5544 3	0	0	0	0	0
	5	0	0	0	0	0	3429 1	0	0	0	0
	6	0	0	0	0	0	0	3036 5	0	0	0
	7	0	0	0	0	0	0	0	197 7	0	0
	8	0	0	0	0	0	0	0	0	216 0	0
	9	0	0	0	0	0	0	0	0	0	28 3

Після досягнення абсолютної точності класифікації комбінацій карт, використовуємо натреновану модель для генерації та класифікації набору даних для подальшого аналізу ймовірності виграшу при кожному класі комбінації. Для цього моделюємо гру в «Техаський Холдем».

Було розроблено алгоритм, за яким у змодельованій грі приймає участь від 2 до 10 гравців, які отримують випадкові карти. В кінці кожної гри визначається переможець. Після моделювання 1 мільйону ігор, в результаті чого було згенеровано 12,003,213 комбінацій карт, передбачимо ймовірність перемоги для кожної з комбінацій (табл.2).

Таблиця 2

Ймовірність виграшу відповідно до типу комбінації

Класкомбі-нації	Назва комбінації	Ймовірність виграшу (%)
0	Nothing in hand	2.2752
1	One pair	21.1825
2	Two pairs	42.0745
3	Three of a kind	52.4582
4	Straight	85.4462
5	Flush	85.9971

6	Fullhouse	91.6273
7	Four of a kind	99.8419
8	Straightflush	99.0461
9	Royalflush	100

На даному прикладі можна пояснити, що перемога у грі буде з вірогідністю 100%, якщо у гравця комбінація «Royalflush», та лише 2,23%, при комбінації «Nothinginhand».

Висновки

1. Серед математичних методів, що існують для вирішення задачі підсистеми для класифікації, були розглянуті наступні: NeuralNetwork, DecisionTreeClassifier, MLP Classifier; LogisticRegression, SVM, RandomForestClassifier, GradientBoostingClassifier. Однією з найбільш підходящих є модель багатошарового перцептрону.
2. Вирішено проблему зниження точності класифікації, пов'язану з незбалансованістю вхідної вибірки.
3. Встановлено, що додаткове обчислення кількості схожих ознак та різниці між значеннями сусідніх карт збільшує точність алгоритму з 60-70% до необхідних 98-100%.
4. Запропоновано часткове дотренування моделі після повного проходження циклу основного тренування моделі, що забезпечує точність прогнозування 100%.
5. Досягнуто абсолютної точності у класифікації комбінацій карт в грі «Техаський холдем».

Література

1. Read J. et al. Classifier chains for multi-label classification // Machine Learning. – 2011. – V. 85. – №. 3. – P. 333-359.
2. Tenenboim-Chekina L., Rokach L., Shapira B. Identification of label dependencies for multi-label classification // Working Notes of the Second International Workshop on Learning from Multi-Label Data. – 2010. – P. 53-60.
3. W. McCulloch and W. Pitts. A logical calculus of ideas immanent in nervous activity. Bull. Math. Biophys., 5:115–133, 1943.
4. F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review, 65(6):386, 1958.

Д.т.н., професор Ориняк І.В., студент Жиляєв М.І.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

МОДЕЛЮВАННЯ ДЕФОРМУВАННЯ КОСОГО СТИКА ТРУБОПРОВОДУ ЯК ПРУЖНОЇ ОБОЛОНКИ ПІД ДІЄЮ ВНУТРІШНЬОГО ТИСКУ

Abstract

Igor V. Orynyak, prof., PhD; Zhylyayev Mykhailo, student

Stress state of pipe mitred bend as an cylindrical shell under inner pressure

The main goal of paper is application so called long and short approximate analytical solutions for cylindrical shell to analysis of the mitred bend. It starts from derivation of explicit expressions for all twelve components of stress strain field for each circumferential mode. The accuracy of them is demonstrated on example of action on the shell of two opposite concentrated radial forces. Then the boundary conditions at the plane of the oblique section are formulated which consist of four vector equations: 2 equations with respect to transverse forces in two directions, 1 - for axial displacement and 1 – for angular displacement. Each of vector equation is able to produce any scalar equations (by expansion into Fourier series) depending on the chosen number of circumferential modes. The results are compared with known experimental and numerical data and exhibit a satisfactory accuracy.

Вступ

Магістральні газо- та нафтопроводи становлять значні матеріальні активи людства. Разом з тим, трубопроводи, особливо їх надземні частини, змінюють траєкторію, щоб компенсувати поздовжні температурні деформації. Ця зміна напрямку трубопроводів відбувається за рахунок використання неперервних криволінійних згинів труби: виготовленого в заводських умовах коліна, чи в польових умовах відводу. Проте їх виготовлення є часто дорогим чи неможливим. В таких випадках на практиці застосовують секторні зварні коліна. Вони виготовляються за допомогою так званих косих стиків. Не дивлячись на широке застосування, практичні методики розрахунків косих стиків є застарілими, і до цього часу відомі стандарти [1] використовують формули створені ще в середині минулого сторіччя на основі формальних аналогій з згинами труб, що розраховуються як тороїдальні оболонки [2]. Таким чином, для косих стиків вимоги існуючих норм є неповними і, часто, нелогічними. Як відмічено в недавньому огляді літератури [3], теоретичні напрацювання

для косих стиків в рамках теорії пружних оболонок є вибірковими, неповними, і практично не використовуються.

Постановка задачі

Відсутність відчутного успіху в теоретичному моделюванні косоного стику є в першу чергу наслідком відсутності загальних ефективних рішень теорії циліндричних оболонок. Разом з тим, розвиток в останні декілька років теорії короткого і довгого рішень, запропонованої в роботах Ориняка І.В. з співробітниками [4-6], а особливо демонстрація точності зв'язаного [6] підходу до розв'язку фундаментальної задачі дії зосередженої сили – свідчить про створення такого ефективного інструментарію рішення задач теорії оболонок. Власне, мета роботи – це застосування вказаної теорії до рішення задачі косоного стику. Основна технічна трудність для неї полягає в формулюванні граничних умов для такої геометрії і створенні відповідного аналітичного і програмного забезпечення для їх врахування при розгляді дії внутрішнього тиску.

Загальні короткі та довгі рішення

За браком місця не будемо наводити постановочні диференціальні рівняння теорії оболонок, а будемо використовувати такі, що наведені в роботі [6]. Відмітимо, що теорія циліндричних оболонок оперує 12 наступними параметрами, що є функціями двох координат - (x, ϕ) , де x – осьовою координатою (по твірній), а ϕ – кутовою координатою. Вони такі: три переміщення - радіальне W , осьове U , та окружне V переміщення; два кути повороту радіальної осі - γ_x в напрямку осі x та γ_ϕ – в напрямку осі ϕ ; дві поперечні сили - Q_x , що діє в площині $x = const$, та Q_ϕ – в площині $\phi = const$; три мембранні сили - осьова N_x , окружна N_ϕ , та дотична L ; а також два згинальних момента - M_x , та M_ϕ . Для зручності чисельної алгоритмізації пронумеруємо їх наступним чином:

$$\begin{aligned} w = \Omega^1, \quad u = \Omega^2, \quad v = \Omega^3, \quad \gamma_x = \Omega^4, \quad \gamma_\phi = \Omega^5, \quad Q_x = \Omega^6, Q_\phi = \Omega^7, \\ N_x = \Omega^8, \quad N_\phi = \Omega^9, \quad L = \Omega^{10}, \quad M_x = \Omega^{11}, \quad M_\phi = \Omega^{12} \end{aligned} \quad (1a)$$

Як прийнято в літературі, представимо загальне рішення в вигляді розкладу в ряд Фур'є по коловій координаті, тоді всі 12 параметрів-функцій можуть бути представлені наступним чином:

$$\Omega^m(x, \phi) = \sum_{n=0}^N \bar{\Omega}_n^m(x) H_n^m(\phi) \quad (16)$$

Де $H_n^m(\phi)$ в залежності від виду навантаження та номеру функції представляється так:

$$H_n^m(\phi) = \begin{cases} (\cos n\phi), \text{ or } (\sin n\phi), & \text{при } m = 1, 2, 4, 6, 8, 9, 11, 12 \\ (\sin n\phi), \text{ or } (-\cos n\phi), & \text{при } m = 3, 5, 7, 10 \end{cases} \quad (1b)$$

Тобто, якщо, наприклад, параметр w вибирається пропорційним $\cos n\phi$, то параметр v залежить від $\sin n\phi$.

Наступною особливістю короткого та довгого рішень є те, що для кожного вибраного значення окружної моди n вони описуються подібними диференціальними рівняннями 4-го ступеня. Їх рішення для нескінченної оболонки в осьовому напрямку для додатніх x може бути формально представлене так:

$$\bar{\Omega}_n^m(x) = \left(\sum_{i=1}^2 B_{n,i} \Gamma_{n,S}^{m,i}(x)\right) + \left(\sum_{i=1}^2 C_{n,i} \Gamma_{n,L}^{m,i}(x)\right), \quad n \geq 0 \quad (2a)$$

Де відомі функції $\Gamma_{n,S}^{m,i}$ характеризують коротке (short), а $\Gamma_{n,L}^{m,i}$ – довге (long) рішення, а невідомі коефіцієнти $B_{n,i}$ та $C_{n,i}$ є метою рішення. Зауважимо наступне. Всі функції $\Gamma_{n,S}^{m,i}$ та $\Gamma_{n,L}^{m,i}$ виражаються через власні функції рішення відповідних диференціальних рівнянь і можуть бути представлені так:

$$\Gamma_{n,S}^{m,i} = \beta_{n,S}^{m,i,1} e^{-c_n x} \cos d_n x + \beta_{n,S}^{m,i,2} e^{-c_n x} \sin d_n x = \beta_{n,S}^{m,i,1} F_{n,S}^1 + \beta_{n,S}^{m,i,2} F_{n,S}^2 \quad (2б)$$

$$\Gamma_{n,L}^{m,i} = \beta_{n,L}^{m,i,1} e^{-a_n x} \cos b_n x + \beta_{n,L}^{m,i,2} e^{-a_n x} \sin b_n x = \beta_{n,L}^{m,i,1} F_{n,L}^1 + \beta_{n,L}^{m,i,2} F_{n,L}^2 \quad (2в)$$

Де всі коефіцієнти β та c_n , d_n та a_n , b_n є відомими константами, що залежать від розмірів та матеріалу оболонки, та отримані нами на основі залежностей приведених в [6]. Строго кажучи, довге рішення не існує при $n = 0$ та $n = 1$. Проте, для цих значень ми замість довгого рішення використовуємо набагато простіше балкове рішення. Воно дається такими формулами для поздовжнього осьового переміщення $u = \Omega^2$:

$$\bar{\Omega}_{n=0}^{m=2}(x, \phi) = (C_{0,1} + C_{0,2}x) \cos 0\phi \quad (3a)$$

для окружної моди $n = 0$ та

$$\bar{\Omega}_{n=1}^{m=2}(x, \phi) = (C_{1,1} + C_{1,2}x) (\cos 1\phi \text{ або } \sin 1\phi) \quad (3б)$$

для окружної моди $n = 1$. При такому переміщенні лише поздовжня сила $N_x = \Omega^8$, не дорівнює нулю, а всі інші компоненти рішення – нульові.

Верифікація для зосередженої сили

Верифікація отриманого рішення проводилася нами для прикладу, коли до нескінченної оболонки в січенні $x = 0$ прикладалися зосереджені радіальні зовнішні сили, що дорівнюють $P = \delta(x) \cos n\phi$, де $\delta(x)$ - функція Дірака. Граничні умови для знаходження коефіцієнтів $B_{n,1}$, $B_{n,2}$ та $C_{n,1}$, $C_{n,2}$ були взяті наступні:

$$Q_{x,n}(0) = \bar{\Omega}_n^6(0) = 1, \quad \bar{\Omega}_n^2(0) = 0, \quad \bar{\Omega}_n^4(0) = 0, \quad \bar{\Omega}_n^{10}(0) = 0 \quad (4)$$

Результати співставлення з даними роботи [4], наприклад при $n = 2$, наведені на Мал1. При інших n результати також були тотожними з [4].

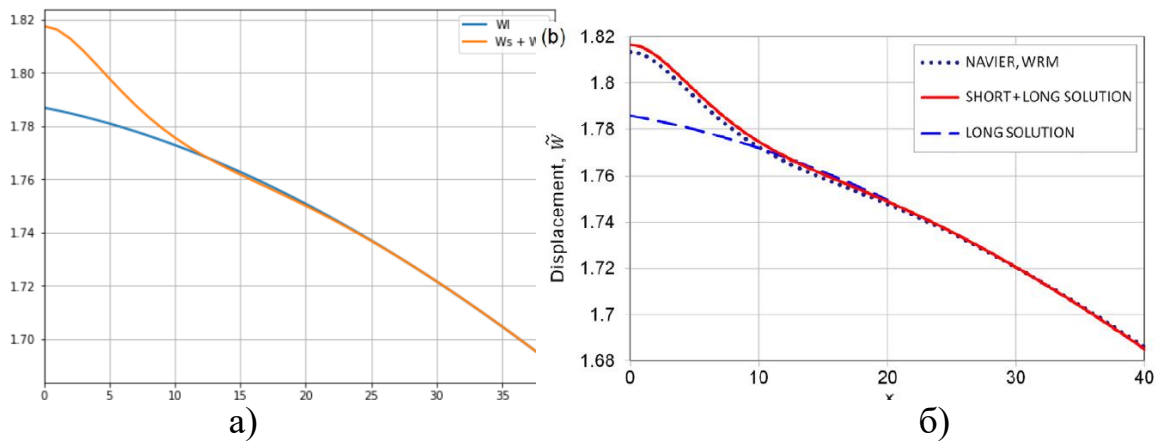


Рис. 1. Порівняння розподілу радіальних переміщень при дії зосередженої радіальної сили для окружної моди $n = 2$: а) отриманих за допомогою запропонованого формального представлення функцій параметрів; б) приведені в роботі [6].

Формулювання граничних умов для косого стика

Як і в розглянутій вище тестовій задачі про дію радіальних сил, нижче для косого стика розглядаються чотири умови спряження по лінії симетрії. Розглядається лише половина всієї конструкції. Проте дана задача не дозволяє незалежне застосування окремих рішень для кожної окружної моди, адже вісь симетрії задачі не співпадає з локальними осями оболонки. Треба почати з понять локальних та глобальних систем координат та відповідних одиничних векторів.

При розгляді задачі для косого стика (Мал 2) нам знадобляться три системи координат.

Розглянемо спочатку глобальні декартові координати. Вони пов'язані з площиною симетрії косого стика. Введемо трійку напрямків і координат. Координати (X, Y, Z) та відповідні їм одиничні вектори (напрямки) $(\vec{I}, \vec{J}, \vec{K})$ позначаємо прописними буквами (Мал 2). Координату X направляємо вправо, Y – вверх, а Z – виходить із площини симетрії і дивиться на нас, тобто вони утворюють праву трійку векторів.

Локальні вектори і напрямки циліндричної оболонки утворені поворотом початкової труби (вектори якої співпадали з глобальними) на кут θ навколо вектора $(-\vec{K})$. Для локальних координат і векторів введемо подібні позначення за допомогою рядкових букв, тоді маємо позначення координат (x, y, z) та одиничних векторів (напрямоків) $(\vec{i}, \vec{j}, \vec{k})$. Тоді локальні вектори визначаються таким чином через глобальні:

$$\vec{i} = \vec{I}\cos\theta - \vec{J}\sin\theta; \quad \vec{j} = \vec{J}\cos\theta + \vec{I}\sin\theta; \quad \vec{k} = \vec{K} \quad (5a)$$

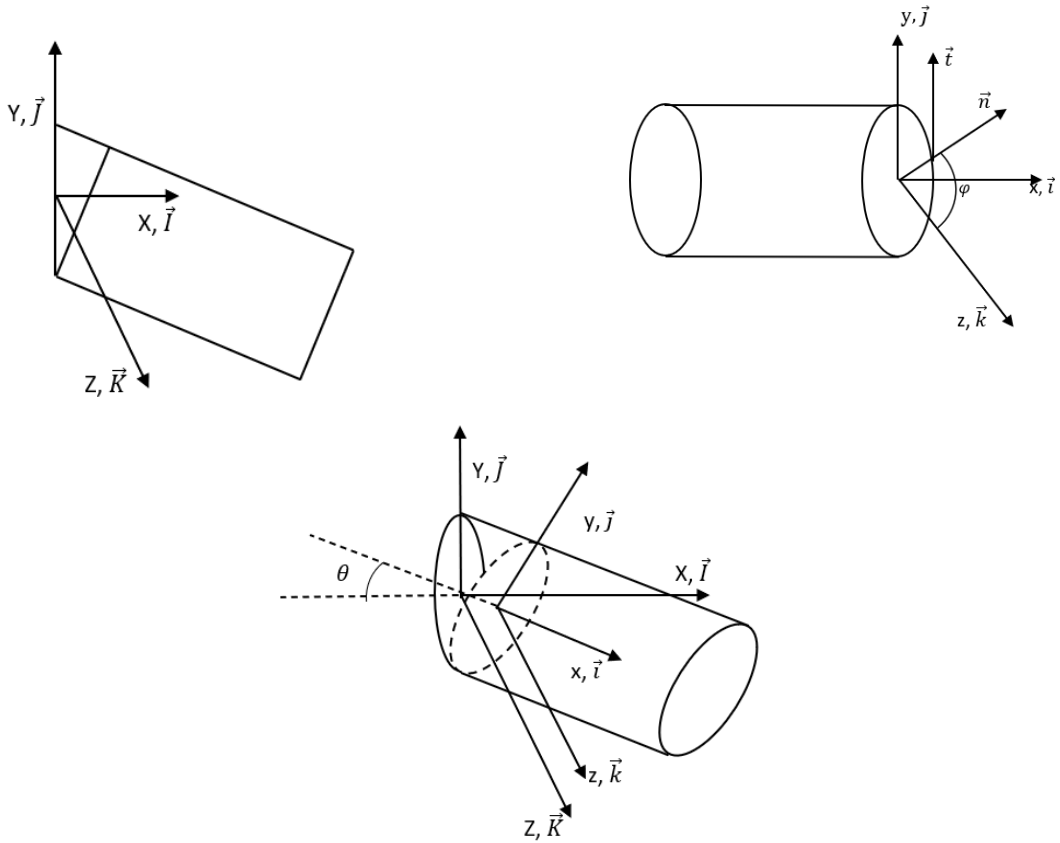


Рис. 2. Локальні та глобальні системи координат для симетричного косоного стику

Проте в оболонках ми працюємо в локальних циліндричних координатах, де кут φ відраховується від осі z до осі y , так що $\varphi = 90$ співпадає з локальним y (Мал2). Для нормального $\vec{n}(\varphi)$ та дотичний $\vec{t}(\varphi)$ до контуру векторів, маємо:

$$\vec{n}(\varphi) = \vec{k}\cos\varphi + \vec{j}\sin\varphi \quad \vec{t}(\varphi) = -\vec{k}\sin\varphi + \vec{j}\cos\varphi \quad (56)$$

Тепер можна легко виразити зв'язок глобальних координат з локальними циліндричними:

$$\begin{aligned} \vec{I} &= \vec{i}\cos\theta + \vec{t}\sin\theta\cos\varphi + \vec{n}\sin\theta\sin\varphi; \\ \vec{J} &= -\vec{i}\sin\theta + \vec{t}\cos\theta\cos\varphi + \vec{n}\cos\theta\sin\varphi; \quad \vec{K} = \vec{k} \end{aligned} \quad (5в)$$

З наведених вище рівнянь отримаємо зв'язок локальних циліндричних з глобальними векторами:

$$\begin{aligned} \vec{t} &= \vec{I}\sin\theta\cos\varphi + \vec{J}\cos\theta\cos\varphi - \vec{K}\sin\varphi; \\ \vec{n} &= \vec{I}\sin\theta\sin\varphi + \vec{J}\cos\theta\sin\varphi + \vec{K}\cos\varphi; \quad \vec{i} = \vec{I}\cos\theta - \vec{J}\sin\theta \end{aligned} \quad (5г)$$

Перейдемо до формулювання граничних умов. Проблема полягає в тому, що умови симетрії існують відносно глобальної площини $X = 0$, а всі аналітичні рішення [6] отримані нами для локальних циліндричних

координат. Для точок косоного стику (границі) значення локальної осьової координати (Мал 2), назвемо його $x_\theta(\varphi)$, такі:

$$x_\theta(\varphi) = -(y + y_c)tg\theta = -R(\sin\varphi + 1) \cdot tg\theta \quad (6a)$$

де θ – це кут (половина) стика, R -радіус труби (Мал 2). Таким чином, мають бути сформульовані граничні умови для кожної точки $x_\theta(\varphi)$. Тобто, всі умови симетрії (рівняння типу (4)) ми формулюємо для x_θ (6a). На цій овальній лінії змінюються одночасно координати x та φ . Якщо розглянути вздовж лінії контакту $s(\varphi)$ елемент довжиною по куту $\Delta\varphi$, то йому буде відповідати елемент довжиною по осі Δx :

$$\Delta x = -\Delta\varphi \cdot tg\theta \cdot R\cos\varphi \quad (6b)$$

Перейдемо до формулювання граничних умов для рівноваги сил. Розглянемо елементарний прямокутний трикутник, що розміщений справа від гіпотенузи Δs . Його катети такі: $\Delta\varphi \cdot R$ та Δx (Мал 3). По цим катетам діють наступні сили:

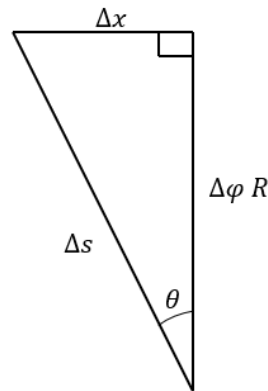


Рис 3. Елементарний прямокутний трикутник по катетам якого діють сили

$$dQ_x = \vec{n}R\Delta\varphi; \quad dQ_\varphi = \vec{n}\Delta x = \vec{n}\cos\varphi \cdot tg\theta R\Delta\varphi \quad (7a)$$

$$dN_x = \vec{i}R\Delta\varphi; \quad dN_\varphi = \vec{i}\Delta x = \vec{i}\cos\varphi \cdot tg\theta R\Delta\varphi \quad (7b)$$

$$dL_\varphi = \vec{t}R\Delta\varphi; \quad dL_x = \vec{t}\Delta x = \vec{t}\cos\varphi \cdot tg\theta R\Delta\varphi \quad (7b)$$

Де для зручності ми ввели поняття $L_x = L_\varphi = L$, оскільки дотична сила, на відміну від нормальних, діє на обох площадках.

Складемо рівняння рівнодіючих сил від двох площадок $\Delta\varphi \cdot R$ та Δx на дві різні проєкції. Спочатку розглянемо рівнодіючу всіх сил на глобальний напрямок \vec{k} на довжині елемента $R\Delta\varphi$. Відповідно до рівнянь (5) маємо:

$$\vec{n} \cdot \vec{k} = \cos\varphi \quad \vec{t} \cdot \vec{k} = -\sin\varphi \quad \vec{i} \cdot \vec{k} = 0 \quad (8a)$$

Тоді узагальнена проєкція всіх внутрішніх сил (7), назвемо її F_K , на глобальний напрямок \vec{k} є така:

$$\frac{F_K}{R\Delta\phi} = Q_x \cos\varphi + Q_\phi \cos\varphi \cos\varphi \operatorname{tg}\theta - N_\phi \sin\varphi \cos\varphi \cdot \operatorname{tg}\theta - L_\phi \sin\varphi \quad (8б)$$

Це перше основне рівняння для внутрішніх сил. В чисельній реалізації є зручнішим, якщо рівняння (8б) формально представляється так:

$$\frac{F_K}{R\Delta\phi} = \sum_{m=1}^{10} l_{m,1} \Omega^m(x_\theta(\varphi), \varphi) \quad (8в)$$

Тут використовуємо нижній індекс $z = 1$, як приналежність до першого рівняння граничної умови, де:

$$l_{1,1} = l_{2,1} = l_{3,1} = l_{4,1} = l_{5,1} = l_{8,1} = 0 \quad l_{6,1} = \cos\varphi, \quad l_{7,1} = \cos\varphi \cos\varphi \operatorname{tg}\theta \\ l_{9,1} = -\sin\varphi \cos\varphi \cdot \operatorname{tg}\theta, \quad l_{10,1} = -\sin\varphi \quad (8г)$$

Друге рівняння складається як рівнодіюча всіх внутрішніх сил на глобальний напрямок \vec{J} . Відповідно до рівнянь (5) проєкції \vec{n} , \vec{i} та \vec{t} на \vec{J} є такими:

$$\vec{n} \cdot \vec{J} = \cos\theta \sin\varphi \quad \vec{t} \cdot \vec{J} = \cos\theta \cos\varphi \quad \vec{i} \cdot \vec{J} = -\sin\theta \quad (9а)$$

Тоді сума всіх сил (7) на елемент довжини $R\Delta\phi$ дає друге рівняння:

$$\frac{F_J}{R\Delta\phi} = Q_x \cos\theta \sin\varphi + Q_\phi \sin\varphi \cos\varphi \sin\theta - N_x \sin\theta + N_\phi \cos\varphi \cos\varphi \cdot \sin\theta - \\ L_x \sin\theta \cos\varphi \cdot \operatorname{tg}\theta + L_\phi \cos\theta \cos\varphi \quad (9б)$$

Формально перепишемо його так:

$$\frac{F_J}{R\Delta\phi} = \sum_{m=1}^{10} l_{m,2} \Omega^m(x_\theta(\varphi), \varphi) \quad (9в)$$

Тут всюди використовуємо нижній індекс « $z = 2$ », як приналежність до другої граничної умови.

$$l_{1,2} = l_{2,2} = l_{3,2} = l_{4,2} = l_{5,2} = 0 \quad l_{6,2} = \cos\theta \sin\varphi, \quad l_{7,2} = \sin\varphi \cos\varphi \sin\theta, \\ l_{8,2} = -\sin\theta, \quad l_{9,2} = \cos\varphi \cos\varphi \sin\theta, \quad l_{10,2} = -\sin\theta \cos\varphi \operatorname{tg}\theta + \cos\theta \cos\varphi \quad (9г)$$

Розглянемо тепер два набори кінематичних граничних умов для переміщень і кутів. Очевидно, що в силу симетрії переміщення (внутрішні і зовнішні) по глобальному напрямку \vec{I} в силу симетрії мають бути нуль. Знайдемо суму всіх внутрішніх компонент. Представимо загальні внутрішні переміщення кожної точки так

$$\vec{\Pi}_{in}(\varphi, x) = (u(\varphi, x)\vec{i} + v(\varphi, x)\vec{t} + w(\varphi, x)\vec{n}) \quad (10а)$$

Найдемо їх проєкцію на горизонтальну глобальну вісь відповідно до рівнянь (5):

$$\vec{\Pi}_{in} \cdot \vec{I} = u(\varphi) \cos\theta + v(\varphi) \sin\theta \cos\varphi + w(\varphi) \sin\theta \sin\varphi \quad (10б)$$

Або в формальному вигляді

$$\vec{\Pi}_{in} \cdot \vec{I} = \sum_{m=1}^{10} l_{m,3} \Omega^m(x_\theta(\varphi), \varphi) = 0 \quad (10в)$$

де:

$$l_{1,3} = \sin\theta \sin\varphi, \quad l_{2,3} = \cos\theta, \quad l_{3,3} = \sin\theta \cos\varphi, \\ l_{4,3} = l_{5,3} = l_{6,3} = l_{7,3} = l_{8,3} = l_{9,3} = l_{10,3} = 0 \quad (10г)$$

Другий набір кінематичних рівнянь відноситься до кутів. Їх буде простіше зрозуміти, якщо залежні від радіальної координати осьові та окружні переміщення представити так:

$$u_r = -\gamma_x r \quad v_r = -\gamma_\varphi r \quad (11a)$$

Тоді будемо вимагати, щоб переміщення в напрямку глобального \vec{I} , були рівними нулю. Запишемо подібно до (10a) вираз для переміщень, що залежать від радіальної відстані від центра січення:

$$\vec{P}_r(\varphi, x)r = -r(\gamma_x(\varphi, x)\vec{i} + \gamma_\varphi(\varphi, x)\vec{t}) \quad (11б)$$

Тоді умова, що проекція всіх цих переміщень на вектор \vec{I} , дає з врахуванням (5):

$$\vec{P}_r(\varphi, x) \cdot \vec{I} = \sum_{m=1}^{10} l_{m,4} \Omega^m(x_\theta(\varphi), \varphi) \quad (11в)$$

де:

$$\begin{aligned} l_{1,4} &= 0, \quad l_{2,4} = -\cos\theta, \quad l_{3,4} = -\sin\theta\cos\varphi; \\ l_{4,4} &= l_{5,4} = l_{6,4} = l_{7,4} = l_{8,4} = l_{9,4} = l_{10,4} = 0 \end{aligned} \quad (11г)$$

Це четвертий набір граничних умов записаних з індексом $z = 4$.

Важливо відмітити, що 4 рівняння граничних умов складним чином залежать від величини кута φ . Для їх практичного застосування треба кожне з цих (векторних) рівнянь розкласти по певному набору тригонометричних функцій, тобто це приведе до конкретних скалярних рівнянь. Кількість цих тригонометричних функцій визначається числом врахованих окружних мод n .

Алгоритм зведення до матриці

Основним параметром-функцією, відносно якої визначаються всі інші параметри, є радіальне переміщення, W . Вибір структури його розкладу залежить від конкретної задачі. В розглядуваній задачі дії внутрішнього тиску структура і кількість членів N шуканого рішення представляється наступним чином:

$$W(x, \varphi) = \sum_{n=0}^{N=3} \bar{\Omega}_n^1 H_n^1 = \sum_{n=0}^{N=3} \left(\left(\sum_{i=1}^2 B_{n,i} \Gamma_{n,S}^{1,i} \right) + \left(\sum_{i=1}^2 C_{n,i} \Gamma_{n,L}^{1,i} \right) \right) H_n^1 \quad (12a)$$

Де $\bar{\Omega}_n^1$ -дається формулами (2a), а $H_n^1(\varphi)$ для заданої задачі вибирається так:

$$H_n^1(\varphi) = \begin{cases} \cos n\varphi, & \text{при } n = 0,2 \\ \sin n\varphi, & \text{при } n = 1,3 \end{cases} \quad (12б)$$

Інші функції окружної координати $H_n^m(\varphi)$ для інших параметрів функцій даються формулою (1в).

Для оптимізації розрахункового алгоритму введемо зручну нумерації невідомих і функцій. Використання загального представлення кожного параметру функції (2a) при кожному n є незручним, адже воно містить 4

невідомих і чотири різних функції. Введемо узагальнений індекс k і перенумеруємо індекси наступним чином:

$$k = \sum_{q=1}^2 \sum_{n=0}^{n=3} \sum_{i=1}^{i=2} ((q-1) \cdot 4 \cdot 2 + 2n + i) \quad (12в)$$

Де $q = 1$, якщо рішення є коротким, індекс "S"; та $q = 2$, якщо рішення є довгим, індекс "L". Відповідно невідомі коефіцієнти $B_{n,i}$ перепозначаються як X_k , де $1 \leq k \leq 8$; а коефіцієнти $C_{n,i}$ – як X_k , де $9 \leq k \leq 16$. При цьому всі функції $\Gamma_{n,S}^{m,i}$ та $\Gamma_{n,L}^{m,i}$ перепозначимо як Ω_k^m відповідно до нумерації (12в).

Таким чином, кожен з параметрів може бути представлений так:

$$\Omega^m = \sum_{k=1}^{16} X_k \Omega_k^m(x, \varphi) \quad (13а)$$

Тепер кожне з чотирьох визначальних рівнянь (граничних умов), а точніше їх права частина може бути записана у вигляді:

$$Z_z(\varphi) = \sum_{m=1}^{10} \sum_{k=1}^{16} X_k \Omega_k^m(x_\theta(\varphi), \varphi) l_{m,z}(\varphi); \quad 1 \leq z \leq 4 \quad (13б)$$

Як ми відмічали вище, кожне з (векторних) рівнянь (13б) має привести до N скалярних рівнянь. Вони формуються наступним чином. Для кожного рівняння вибирається набір початкових функцій Фур'є, відповідно до яких може бути розкладена функція (13б). Ці функції нами позначаються як $RU_{z,p}$, де z – номер векторного рівняння, а p – номер вибраної тригонометричної функції Фур'є. Для даної задачі вони такі. Для $z = 1$:

$$RU_{1,1} = \cos\varphi, \quad RU_{1,2} = \sin 2\varphi \quad RU_{1,3} = \cos 3\varphi \quad RU_{1,4} = \sin 4\varphi \quad (13в)$$

А для всіх інших z , тобто для $z = 2, 3, 4$

$$RU_{z,1} = \cos 0\varphi, \quad RU_{z,2} = \sin 1\varphi \quad RU_{z,3} = \cos 2\varphi \quad RU_{z,4} = \sin 3\varphi \quad (13г)$$

Підсумком складання алгоритму є отримання коефіцієнтів результуючої матриці. Для цього спочатку встановлюється номер, l , скалярного рівняння відповідно до наступного правила:

$$l = N(z-1) + p = 4(z-1) + p \quad (14а)$$

Це дає змогу отримати коефіцієнти матриці, a_l^k , в загальному рівнянні:

$$a_l^k X_k = b_l \quad (14б)$$

де k – номер по вертикалі (стовпчик); l – номер по горизонталі (рядок) матричного рівняння, числа вільного вектор-стовпчика, b_l , залежать від зовнішнього навантаження. Коефіцієнти a_l^k отримуються шляхом множення відповідного векторного граничного рівняння на конкретну функцію $RU_{z,p}$ і інтегрування по всій границі (тобто по $0 \leq \varphi \leq 2\pi$). Остання процедура відбувається чисельно, використовуючи дискретні точки, φ_j , з кроком $\Delta\varphi = 2\pi/360$. Ці точки визначаються по правилу:

$$\varphi_j = \frac{\Delta\varphi}{2} + (j-1)\Delta\varphi \quad (15а)$$

Для симетричних задач (тут ми розглядаємо таку), інтервал зменшується вдвоє. Таким чином, маємо для коефіцієнтів матриці рівняння:

$$a_{l,k} = \sum_{m=1}^{10} \sum_{j=1}^{180} \Omega_k^m(x_\theta(\varphi_j), \varphi_j) l_{m,z}(\varphi_j) RU_{z,p}(\varphi_j) \quad (156)$$

Переходимо до знаходження членів вільного стовпчика результуючого матричного рівняння, тобто значень b_l . Вони залежать від конкретного зовнішнього навантаження. У випадку дії внутрішнього тиску, P , в прямій трубі виникають наступні сили та переміщення, які в даному випадку можна розглядати як зовнішні фактори:

$$N_\varphi = -pR \quad N_x = -\frac{pR}{2} \quad (16a)$$

Відповідно до загальної методології рішення задач - внутрішні сили врівноважують зовнішні. Виходимо з принципу, що зовнішні сили мають врівноважувати внутрішні. Підставивши в праву частину (8б) значення N_φ та N_x , отримуємо праву частину рівняння (8в):

$$\sum_{m=1}^{10} l_{m,1} \Omega^m(x_\theta(\varphi), \varphi) = \sin\varphi \cos\varphi \cdot tg\theta \cdot pR = \frac{\sin 2\varphi}{2} tg\theta \cdot pR \quad (16б)$$

Множення правої частини (16б) на $RU_{1,p}$ і інтегрування по куту дають наступні значення b_l :

$$b_l = \sum_{j=1}^{180} pR \frac{\sin 2\varphi_j}{2} tg\theta \cdot RU_{1,p}(\varphi_j), \quad 1 \leq l \leq 4 \quad (16в)$$

Аналогічно підставивши в праву частину (9б) значення N_φ та N_x , отримуємо праву частину рівняння (9в):

$$\sum_{m=1}^{10} l_{m,2} \Omega^m(x_\theta(\varphi), \varphi) = -\cos 2\varphi \frac{\sin\theta \cdot pR}{2} \quad (17a)$$

Тому застосовуючи подібним чином $RU_{2,p}$:

$$b_l = \sum_{j=1}^{180} pR \frac{\sin 2\varphi_j}{2} tg\theta \cdot RU_{2,p}(\varphi_j), \quad 5 \leq l \leq 8 \quad (17б)$$

Перейдемо до третього векторного граничного рівняння, що стосується переміщення. Під діє внутрішнього тиску в прямій трубі виникають відомі радіальні переміщення:

$$W_p = \frac{pR^2}{hE} \left(1 - \frac{\mu}{2}\right) \quad (18a)$$

Оскільки сума внутрішніх (додаткових) і зовнішніх (від тиску) переміщень повинна бути рівною нулю, то вклад в рівняння (10б) від W_p приводить до виникнення в ньому правої частини. Рівняння (10в) прийме вигляд:

$$\sum_{m=1}^{10} l_{m,3} \Omega^m(x_\theta(\varphi), \varphi) = -\frac{pR^2}{hE} \left(1 - \frac{\mu}{2}\right) \sin\theta \sin\varphi \quad (18б)$$

Застосовуючи подібним чином $RU_{3,p}$, отримуємо коефіцієнти:

$$b_l = \sum_{j=1}^{180} -\frac{pR^2}{hE} \left(1 - \frac{\mu}{2}\right) \sin\theta \sin\varphi \cdot RU_{3,p}(\varphi_j), \quad 9 \leq l \leq 12 \quad (18в)$$

Що стосується інших коефіцієнтів b_l , то від внутрішнього тиску не виникає правої частини в векторному рівнянні (11в) для кутів. Тому маємо:

$$b_l = 0 \quad 13 \leq l \leq 16 \quad (18г)$$

Таким чином, всі постановочні рівняння для сформульовані, що дало змогу створити розрахункову програму і отримати конкретні рішення для невідомих X_k , а значить для всіх функцій-параметрів, що характеризують

напружено-деформований стан косоного стика при довільному куту θ , як елемента циліндричної оболонки.

Висновки

1. На основі векторних уявлень про всі силові та деформаційні параметри теорії оболонок в роботі вперше сформульовані загальні граничні умови для одиночного косоного стика. Вони представлені як складні функції від кутової координати. Для приведення їх до зручного вигляду ці умови чисельно розкладалися в ряд Фур'є, що дало змогу з кожної граничної умови отримати певну кількість визначальних рівнянь

2. Сформульовано перелік власних функцій і невідомих коефіцієнтів при них, знаходження яких і було метою аналізу при кожному конкретному випадку навантаження косоного стика. Короткі та довгі рішення доповнювалися простими рішеннями теорії балок – для розтягу та згинання.

3. Розроблений уніфікований алгоритм рішення задачі та розроблена відповідна розрахункова програма для аналізу деформування косоного стика при довільних його геометричних розмірів, кута стику θ та довільних характеристик матеріалу, що зводиться до розв'язання системи лінійних рівнянь.

4. Верифікація отриманих рішень приведена для аналізу поведінки прямої труби при дії зосереджених діаметрально прикладених сил, що є фундаментальним розв'язком для циліндричної оболонки, тобто дозволяє отримати всі інші можливі види навантажень.

Література

1. ASME B31.8-2016. Gas transmission and distribution piping systems. The American Society of Mechanical Engineers, USA 2016
2. Clark, R. A., Reissner. E. Bending of curved tube, *Advances in Applied Mechanics*. Academic Press, New York, 1951 V2, 95-122.
3. Wood, J. A review of literature for the structural assessment of mitred bends. *International Journal of Pressure Vessels and Piping*, 85.5 (2008): 275-294.
4. Orynyak, I, & Orynyak, A. "Efficient Solution for Cylindrical Shell Based on Short and Long (Enhanced Vlasov's) Solutions on Example of Concentrated Radial Force." *Proceedings of the ASME 2018 Pressure Vessels and Piping Conference*. Volume 3A: Design and Analysis. Prague, Czech Republic. July 15–20, 2018. V03AT03A033. ASME.
5. Orynyak, A., and Orynyak, I. (2020). "Application of Short and Long (Enhanced Vlasov's) Solutions for Cylindrical Shell on Example of

Concentrated Radial Force." ASME. J. Pressure Vessel Technol. February 2021; 143(1), 014501.

6. *Orynyak, Igor, and Yulia Bai.* "Coupled approximate long and short solutions versus exact Navier and Galerkin ones for cylindrical shell under radial load." *Thin-Walled Structures* 170 (2022)

Аспірант Рудник Т. П.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

МАТЕМАТИЧНИЙ МЕТОД АНАЛІЗУ БАГАТОМОВНИХ ПОВІДОМЛЕНЬ ІЗ СОЦІАЛЬНОЇ МЕРЕЖІ ТВІТТЕР

Abstract

Taras Rudnyk, PhD student

Mathematical method of multilingual analysis of posts on Twitter

This work concerns the task of natural language processing based on the Word2Vec methodology. I propose an approach to classify Twitter posts in Ukrainian, English, French, and Russian, as well as tweets that contain words in several languages. Our method was tested on political posts by Volodymyr Zelenskyy, Emmanuel Macron, and Marine Le Pen subscribers using F1 score. The work ends with ideas for further research.

Вступ

Задача обробки природної мови методами машинного навчання часто виникає в сучасному світі. Зі збільшенням популярності мережі Інтернет щорічно збільшується кількість текстів на сайтах та повідомлень у соціальних мережах.

Більшість сучасних методів аналізу природної мови зосереджені на англійській мові. Проте часто виникає потреба в аналізі текстів іншими мовами.

Існує багато методів аналізу природної мови. Вибір методу залежить від представлення інформації в наборі даних та мети аналізу повідомлень. Наприклад Word2Vec [1] добре підходить для аналізу тексту з врахуванням контексту слів. Doc2Vec [2] краще підходить для аналізу великих текстів, таких як параграфи чи документи. TF-IDF [3] зазвичай використовується коли потрібно визначити важливість слова у документі. Методи виділення ознак на основі структури [4] продемонстрували хороші результати при аналізі технічних статей. У даній роботі запропоновано модифікацію відомої методики опрацювання природної мови Word2Vec для аналізу політичних повідомлень у соціальній мережі Твіттер англійською, французькою, українською та російською мовами, а також повідомлень, які містять слова з різних мов одночасно. Така модифікація дає змогу точніше визначати зміст повідомлень та зробити висновки чи

вони відносяться до політичного діяча. А також позитивно чи негативно автор повідомлення відноситься до політика.

Постановка задачі

Задача полягає в модифікації існуючої методики обробки природної мови [1], щоб визначити ставлення авторів повідомлень у соціальній мережі Твіттер до українського та французьких політичних діячів — Президента України Володимира Зеленського, як найпопулярнішого політичного діяча України, рейтинг якого часто змінювався, Президента Франції Емманюеля Макрона та лідера політичної партії «Національне об'єднання» Марін Ле Пен, як основних кандидатів на перемогу у виборах Президента Франції в 2022 році. Модель повинна добре працювати з українською, англійською, французькою, та російською мовами, а також із повідомленнями, які містять слова з різних мов.

Термінологія

Стоп-слова — це слова, які не несуть смислового навантаження, тому їхня користь та роль для аналізу тексту не суттєва.

Лематизація — процес проведення слова в нормальну словникову форму, який включає в себе визначення частини мови.

Нейронні мережі — це алгоритм машинного навчання, який будується на принципі організації та функціонування біологічних нейронних мереж.

Word2Vec — методика опрацювання природної мови, яка використовує нейромережеву модель для навчання пов'язаностей слів із великого корпусу тексту. Відповідним способом навчена модель може виявляти слова-синоніми або підказувати додаткові слова для неповного речення. Як випливає з її назви, Word2Vec подає кожне слово як вектор дійсних чисел.

Вибір методів, підходів, моделей та інструментів розв'язання поставленої задачі

Алгоритм багатомовного аналізу текстових повідомлень із соціальної мережі Твіттер передбачає такі етапи:

1) Визначення мови. За допомогою словника слів визначається, до якої мови належить кожне слово в реченні. Для кожної мови використовується окрема навчена нейронна мережа. У випадку повідомлень із словами з декількох мов визначається відсоткове співвідношення використаних повідомленні мов та застосовується гібридна модель, яка працює з декількома мовами.

2) Приведення текстів до однорідного формату:

- a) усі букви переводять у нижній регістр;
- b) видалення знаків пунктуації та спеціальних символів;
- c) видалення стоп-слів;
- d) лематизація слів.

3) Вибір хештегів із твітів користувачів, які допомагають визначити, чи текст є на підтримку або проти політичного діяча.

4) Переведення слів у вектори та навчання моделі за допомогою алгоритму Word2Vec.

5) Переведення твітів у вектори. Для цього береться середнє значення вектору кожного слова з твіту.

6) Застосування алгоритму Extreme Gradient Boosting [5] для класифікації твітів.

7) Перевірка результатів за допомогою міри F1. Важливими компонентами цієї міри є:

- a) Істиннопозитивні значення – загальне число правильно класифікованих об'єктів;
- b) Хибнопозитивні значення – загальне число неправильно класифікованих негативних об'єктів;
- c) Хибнонегативні значення – загальне число неправильно класифікованих позитивних об'єктів.

Спочатку розраховується влучність за формулою

$$\text{Влучність} = \frac{\text{істиннопозитивні}}{\text{істиннопозитивні} + \text{хибнопозитивні}}$$

Далі розраховується повнота за формулою

$$\text{Повнота} = \frac{\text{істиннопозитивні}}{\text{істиннопозитивні} + \text{хибнонегативні}}$$

Базуючись на влучності та повноті, розраховується значення міри F1 за формулою

$$F1 = 2 * \frac{\text{Повнота} * \text{Влучність}}{\text{Повнота} + \text{Влучність}}$$

Розв'язання поставленої задачі

Для дослідження рівня підтримки політичного діяча в Україні було зібрано повідомлення підписників Володимира Зеленського в Твіттері з 1 січня 2019 року по 31 березня 2021 року. Більшість із них була українською (32%), англійською (14%), російською (43%) мовою та поєднанням української і російської мови (9%). У Табл. 1 наведено результати для всіх перерахованих мов.

Таблиця 1

Значення міри F1 для мов, які використовувались підписниками
Володимира Зеленського

Мова	Значення міри F1
українська	0.67
англійська	0.76
російська	0.69
поєднання української та російської	0.64

Для дослідження важливий загальний результат для набору даних, а не лише по кожній мові окремо. У Табл. 2 наведено результат при використанні трьох методів – відкидання багатомовних твітів, виявлення домінуючої мови в повідомленні і застосування моделі до неї, та запропонований у цій статті метод аналізу багатомовних повідомлень.

Таблиця 2

Значення міри F1 отримані при використанні різних методів для набору
даних про Володимира Зеленського

Метод	Значення міри F1
відкидання багатомовних твітів	0.61
застосування моделі до домінуючої мови в твіті	0.65
запропонований у статті метод	0.68

Для дослідження виборів у Франції було зібрано повідомлення підписників Емманюеля Макрона та Марін Ле Пен у Твіттері з 1 вересня 2021 року по 30 квітня 2022 року для порівняння результатів з соціологічними опитування, які почали публікуватись з кінця серпня 2021 року. Більшість повідомлень була французькою (49%) або англійською мовою (43%). У Табл. 3 наведено результати для всіх мов та двомовних твітів, які містять французькі та англійські слова (7%).

Таблиця 3

Значення міри F1 для мов, які використовувались підписниками
Емманюеля Макрона та Марін Ле Пен

Мова	Значення міри F1
французька	0.65
англійська	0.78
поєднання французької та англійської	0.58

У Табл. 4 наведено результат при використанні тих самих трьох методів, які використовувались при аналізі набору даних про Володимира Зеленського, до набору даних зібраного з повідомлень підписників Емманюеля Макрона та Марін Ле Пен.

Таблиця 4

Значення міри F1 отримані при використанні різних методів для набору даних про вибори у Франції

Метод	Значення міри F1
відкидання багатомовних твітів	0.63
застосування моделі до домінуючої мови в твіті	0.67
запропонований у статті метод	0.69

Висновки

Запропонований алгоритм дав змогу автоматизовано аналізувати неоднорідні набори даних, які містили англійську, українську, французьку та російську мови, а також повідомлення, що поєднують слова з різних мов. Отриманий результат в середньому має значення міри F1 на 0.02-0.07 вище від підходів, у яких відкидають багатомовні твіти або застосовують моделі до домінуючої в повідомленні мови.

Слід зазначити, що найкращий результат було отримано для англійської мови, що не дивно, оскільки дана мова легше піддається машинному аналізу за рахунок відсутності словозміни та чіткому порядку слів у речення. Для твітів, які поєднують французьку та англійську мову, результат виявився найгіршим. Це пояснюється тим, що в наборі даних таких повідомлень було менше всього.

У подальшому було б доцільно замість методу Word2Vec спробувати застосувати інші відомі методи, на кшталт Bag-of-Words, Doc2Vec, TF-IDF та для тих самих наборів даних та порівняти результати.

Література

1. Mikolov T., Kai C., Corrado G., Dean J. Efficient Estimation of Word Representations in Vector Space. Computing Research Repository. – 2013. – P.12.
2. Quoc V., Mikolov T. Distributed Representations of Sentences and Documents. Computing Research Repository. – 2014. – P.9.
3. Bafna P., Pramod D., Vaidya A. Document clustering: TF-IDF approach. IEEE int. conf. on electrical, electronics, and optimization techniques (ICEEOT). – 2016. – pp 61-66.

4. Nguyen T., Shirai K. Text classification of technical papers based on text segmentation. Int. conf. on application of natural language to information systems. – 2013. – pp 278-284.
5. Tianqi C., Guestrin C. XGBoost: A Scalable Tree Boosting System. Computing Research Repository. – 2016. – P.13.

Доцент Сирота С. В., студент Агафонов Д. С.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

ПОРІВНЯЛЬНИЙ АНАЛІЗ ПІДХОДІВ ТА МЕТОДІВ ОЦІНЮВАННЯ ЕМОЦІЙНОГО ЗАБАРВЛЕННЯ ФІНАНСОВОЇ ІНФОРМАЦІЇ ПРО ПОТОЧНИЙ СТАН КОМПАНІЇ

Abstract

Sergei Syrota, docent; Dmitriy Agafonov, student.

Comparison of approaches and methods for evaluating the sentiment of financial information about the companies

This article discusses neural network architecture and its applications for extracting sentiment from special financial text and label it. News and investor sentiment are important factors in securities prices in the financial markets. Thus, using the capabilities of modern NLP approaches to analyze financial sentiment is a crucial component in identifying patterns and trends that are beneficial to market participants and regulators. Analysis of financial sentiment is a difficult task due to the specialized language and the lack of labeled data in this area. General purpose models are not effective enough due to the specialized language used in the financial context. Pre-trained language models can help with this problem, as they require fewer examples, and can be further taught on specific corpora. In recent years, methods using Transformer-based transfer learning, such as BERT, have performed well in text classification tasks, including sentiment analysis.

Вступ

Існує декілька теорій щодо ефективності ринків, що ціни відображають всю відому інформацію та пристосовуються до нової інформації або складаються випадковим чином. Одна з застосованих серед учасників фінансових ринків теорій – Гіпотеза ефективного ринку (Efficient Market Hypothesis ЕМН – гіпотеза, згідно з якою вся суттєва інформація негайно та повною мірою відбивається на ринковій курсовій вартості цінних паперів) Юджина Фама [1]. У цій теорії було описано три рівні ринкової ефективності: слабкої, середньої та сильної форми. Середня ефективність означає, що поточна ціна відображає всю загальнодоступну інформацію, і прихована інсайдерська інформація може призвести до зміни руху ціни.

Коригування цін у формі середньої ефективності за рахунок нової інформації є широко використовуваним ринковим принципом. Фінансові тексти, такі як новини, звіти аналітиків, офіційні повідомлення компанії, є джерелом нової інформації. Ручна обробка великої кількості даних, які оновлюються в кожен момент часу та отримання корисних висновків є надто складним завданням для будь-якої окремої організації. Вирішити вищезазначену проблему може автоматизований аналіз настроїв або полярності текстів, створених фінансовими суб'єктами за допомогою методів обробки природної мови (Обробка природної мови NLP – це загальний напрямок штучного інтелекту та математичної лінгвістики, вивчає проблеми комп'ютерного аналізу та синтезу текстів природними мовами) [4].

Методи машинного або глибокого навчання для обробки інформації є важливим та ефективним інструментом на фінансових ринках. Алгоритми текстового аналізу та торгівлі на основі новин використовуються, щоб визначити, чи є поточний потік новин про компанії та їхні ціни на акції, валюти чи товари позитивними чи негативними. У цьому контексті застосування аналізу настроїв учасників фінансового ринку стосовно оцінювання стабільності фінансової установи та коректності інформації в публікаціях про цю фінансову установу та інші особливості є надзвичайно актуальною задачею.

Автором цієї статті вже були започатковані дослідження щодо моделі BERT для вирішення фінансових задач в рамках виконання бакалаврської роботи і частково ці результати будуть використовуватись в цій статті.

Постановка задачі

Метою даної статті є дослідження та проведення порівняльного аналізу підходів та методів оцінювання семантичного забарвлення фінансових текстів (новини, твіти, звіти аналітиків, офіційні повідомлення компанії) про поточний стан компанії з точки зору інвестора, тобто, чи може текстова інформація мати позитивний, негативний або нейтральний вплив на ціну акцій.

Предметом дослідження у статті є обґрунтування вибору методів, моделей та засобів оцінювання емоційного забарвлення фінансової інформації про поточний стан компанії.

Кінцевим результатом роботи є систематизований перелік методів, моделей та інструментів оцінювання емоційного забарвлення фінансової інформації про поточний стан компанії, необхідних і достатніх для практичного застосування.

Підходи до аналізу настроїв у фінансах

Аналіз настроїв – це завдання вилучення настроїв чи думок із письмової мови [6]. Можна розділити існуючі підходи на дві групи: 1) Методи машинного навчання з ознаками, отриманими з тексту за допомогою «word counting» [7], 2) Методи глибокого навчання, де текст представлений послідовністю ембедингів. [8]. Перший страждає від нездатності відобразити семантичну інформацію, що впливає з певної послідовності слів, тоді як другий часто вважається занадто «вимогливим до даних», оскільки він вивчає набагато більшу кількість параметрів.

Аналіз фінансових настроїв відрізняється від загального аналізу настроїв не тільки предметною областю, а й метою. Метою аналізу фінансових настроїв зазвичай є прогнози про те, як ринки відреагують на інформацію, представлену в тексті [9].

Використовуючи перший підхід – машинне навчання з ознаками, отриманими з тексту за допомогою «word counting», Loughran і McDonald (2016) представляють огляд останніх робіт з аналізу фінансового тексту з використанням машинного навчання з підходом «bag-of-words» або методами на основі лексиконів [10]. Створено словник фінансових термінів із присвоєними значеннями («позитивний», «невизначений») і виміряно тональність документів, підраховуючи слова з певним словниковим значенням [11]. Іншим прикладом є Pagolu et al. (2016), де n-грами з твітів з фінансовою інформацією надходять у контрольовані алгоритми машинного навчання для виявлення настроїв щодо згаданої фінансової організації.

Однією з перших робіт, які використовували методи глибокого навчання для аналізу фінансової полярності тексту, були Kraus and Feuerriegel (2017) [12]. Вони застосовують нейронну мережу LSTM до спеціальних оголошень компанії, щоб передбачити рух фондового ринку та показати, що цей метод більш точний, ніж традиційні підходи машинного навчання. Є кілька інших робіт, які використовують різні типи нейронних архітектур для аналізу фінансових настроїв. Lutz et al. 2018 [13] використовують підхід до використання doc2vec для створення ембедингів речень що стосуються певної компанії та використовують багатократне навчання (multiple-instance learning) [14] для прогнозування результатів фондового ринку.

Через відсутність великих позначених фінансових наборів даних важко використовувати нейронні мережі в повній мірі для аналізу настроїв. Навіть коли їхні перші шари (вбудовування слів) ініціалізуються попередньо навченими значеннями, решті моделі все одно потрібно вивчати складні відносини з відносно невеликою кількістю позначених даних. Більш перспективним рішенням може бути ініціалізація майже всієї

моделі з попередньо навченими значеннями та тонка настройка цих значень щодо завдання класифікації.

Навчання з перенесенням (Transfer Learning)

В даній статті розглядається аналіз настроїв, який класифікує текст як позитивний, негативний або нейтральний у певній області. Це вимагає вирішення двох проблем: 1) Найскладніші методи класифікації, які використовують нейронні мережі, вимагають величезної кількості розмічених даних, а позначення фрагментів фінансового тексту вимагає втручання експертів. 2) Моделі аналізу настроїв, навчені на загальних корпусах, не підходять для цього завдання, оскільки фінансові тексти мають спеціалізовану мову з унікальним словником і мають тенденцію використовувати нечіткі вирази замість легко ідентифікованих негативних/позитивних слів.

Методи навчання з перенесенням (TL) NLP є перспективним рішенням обох проблем, згаданих вище. Основна ідея трансферного навчання полягає в тому, що шляхом навчання мовних моделей на дуже великих корпусах, а потім ініціалізації шарів моделей з ваговими показниками, отриманими під час донавчання, можна досягти набагато кращої продуктивності. Ініціалізовані шари можуть варіюватися від шару ембедингу [5] одного слова до всієї моделі.

Однією із особливостей методів навчання з перенесенням є можливість подальшого донавчання мовних моделей на предметному немаркованому корпусі. Таким чином, модель може вивчати семантичні відносини в тексті цільової області, яка, ймовірно, матиме іншу специфічну лексику та змістовний напрямок, ніж загальний корпус. Цей підхід є особливо перспективним для фінансової галузі, оскільки мова та лексика, що використовуються, різко відрізняються від загальних.

Класифікація тексту з використанням переднавчених моделей

Моделювання живої моделювання — це завдання передбачити наступне слово в заданому фрагменті тексту. Одним з найважливіших останніх досягнень в обробці природної мови є усвідомлення того, що модель, розроблена і навчена для моделювання мови, може бути успішно налаштована для більшості завдань NLP з невеликими модифікаціями. Такі моделі навчаються на дуже великих корпусах тексту, а потім до них додаються відповідні шари для конкретних завдань, які налаштовуються на цільовому наборі даних [15]. Класифікація тексту є одним із очевидних варіантів використання цього підходу.

ELMo (Embeddings from Language Models) – це тип глибокого контекстуалізованого представлення слів, який моделює як складні

характеристики вживання слів (наприклад, синтаксис і семантику), так і те, як ці вживання змінюються в різних мовних контекстах [16] було одним із перших успішних застосувань цього підходу. З ELMo, глибока двонаправлена мовна модель попередньо тренується на великому корпусі. Для кожного слова приховані стани цієї моделі використовуються для обчислення контекстуалізованого ембедингу. Використовуючи попередньо навчені ваги ELMo, контекстуалізовані ембединги слів можна розрахувати для будь-якого фрагмента тексту. Було показано, що ініціалізація ембедингів для наступних завдань з вагами покращує продуктивність більшості завдань порівняно зі статичними ембедингами слів, такими як word2vec або GloVe.

Хоча ELMo використовує попередньо навчені мовні моделі для контекстуалізації уявлень, інформація, отримана за допомогою мовної моделі, присутня лише на першому шарі будь-якої моделі, яка її використовує. ULMFit (Universal Language Model Fine-tuning) [17] було першим рішенням, завдяки якому було досягнуто справжнього трансферного навчання для NLP, використовуючи нові методи, такі як дискримінаційне тонке налаштування, похилі трикутні темпи навчання. Рішення дало змогу ефективно налаштувати цілу попередньо навчену мовну модель для класифікації тексту. Також запроваджено подальше попереднє навчання мовної моделі на предметному корпусі, припускаючи, що дані окремого завдання з іншого розподілу, ніж загальний корпус, на якому навчалася початкова модель.

Основна ідея ULMFit щодо ефективного тонкого налаштування попередньо навченої мовної моделі для завдань була виведена на інший рівень за допомогою двоспрямованих кодувальних представлень з Transformers (BERT).

BERT для вирішення фінансових задач: FinBERT

BERT (Bidirectional Encoder Representations from Transformers) – це двоспрямовані кодувальні представлення з Transformers, техніка машинного навчання, призначена для переднавчання глибоких двонаправлених представлень на нерозмічених текстових даних шляхом спільної обробки лівого та правого контекстів, розроблена працівниками компанії Google. [18] Завдяки цьому, при додаванні лише одного вихідного шару, модель може отримувати гарні результати для різних задач, починаючи від прогнозування наступного речення та додавання пропущених слів в правильному контексті до відповідей на запитання та аналізу настроїв в тексті.

При реалізації моделі BERT існує два етапи: попереднє навчання (pre-training) та налаштування (fine-tuning). Під час попереднього навчання

модель тренується на нерозмічених даних у складі різних задач. Під час налаштування BERT ініціалізується з параметрами, отриманими в ході попереднього навчання, а потім параметри налаштовуються на розмічених даних. Перевага моделі BERT – вона уніфікована для розв’язання різних задач. Між наперед навченою архітектурою і фінальною є мінімальні відмінності. Наприклад, за виключенням вихідних шарів, одна й та сама архітектура використовується для переднавчання та для налаштування моделі, адже при налаштуванні параметри оптимізуються. Даний процес представлено на рисунку 1. [CLS] – це спеціальне позначення, яке додається на початку кожного вхідного масива, [SEP] – це позначення розділення (двох речень, або питання та відповіді)

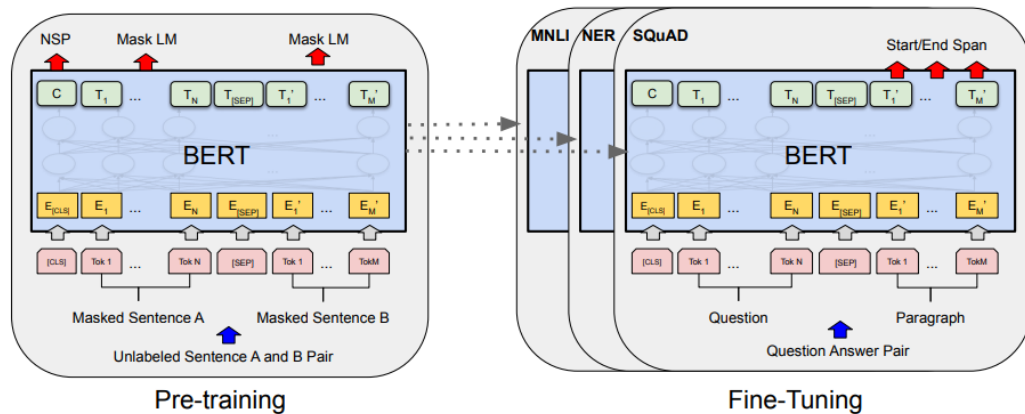


Рис. 1. Попереднє навчання та налаштування моделі BERT [18]

Архітектура моделі BERT є багатoshаровий двоспрямований енкодер-трансформер (multi-layer bidirectional Transformer encoder), що, в свою чергу, є архітектурою для моделювання послідовної інформації (альтернатива RNN). Початкова реалізація архітектури трансформера описана в [19].

В моделі BERT використовується алгоритм токенизації підслів WordPiece [20]. WordPiece спочатку ініціалізує словник, щоб включити кожен символ, присутній в навчальних даних, і поступово вивчає задану кількість правил злиття. WordPiece вибирає не найчастішу пару символів, а ту, яка максимізує ймовірність того, що навчальні дані будуть додані в словник.

Для даного токена вхідне представлення створюється шляхом суми відповідних вкладень токена, сегмента і позиції. Візуалізацію цієї конструкції можна побачити на рисунку 2.

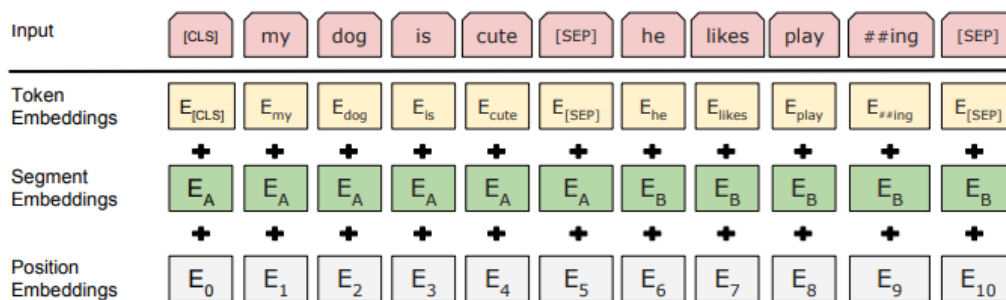


Рис. 2. Вхідне представлення BERT [18]

Під час попереднього навчання двосторонній аналіз контекстів на всіх шарах створює деякі обмеження, тому для переднавчання моделі використовують масковану мовну модель (Masked Language Model MLM) [21]. Суть MLM полягає в тому, що замість прогнозування наступного слова на основі попереднього, як було б в мовній моделі на основі LSTM або інших стандартних односпрямованих моделях, «маскується» 15% всіх токенів WordPiece в випадковому порядку, які потім прогножуються. Це дозволяє отримати двоспрямовану наперед навчену модель.

BERT має дві версії:

- базова BERT-base, яка складається з 12-ти шарів енкодера, 768-ми прихованих шарів, 12-ти голівок multi-head attention та 110-ти мільйонів параметрів;
- розширена BERT-large, яка складається з 24-х шарів енкодера, 1024-х прихованих шарів, 16-ти голівок multi-head attention та 340-ка мільйонів параметрів.

Для семантичного аналізу настрою тексту використана модель financial BERT (finBERT) [22]. FinBERT це преднавчена NLP модель для аналізу настрою в фінансовому тексті на основі BERT. Класифікація настроїв відбувається завдяки додаванню шару Dense (повнозв'язного нейрону з нелінійною функцією активації GELU [23]) після останнього прихованого стану токена [CLS]. Це продемонстровано на рисунку 3. Після цього нейронна мережа навчається на розміченому датасеті.

Для попереднього навчання моделі використано велика кількість фінансового тексту з Financial PhraseBank. Financial PhraseBank складається з великої кількості речень на англійській мові, які було випадково вибрано з фінансових новин бази даних LexisNexis [24] та розмічено експертами з досвідом роботи у фінансовій та бізнес-сфері.

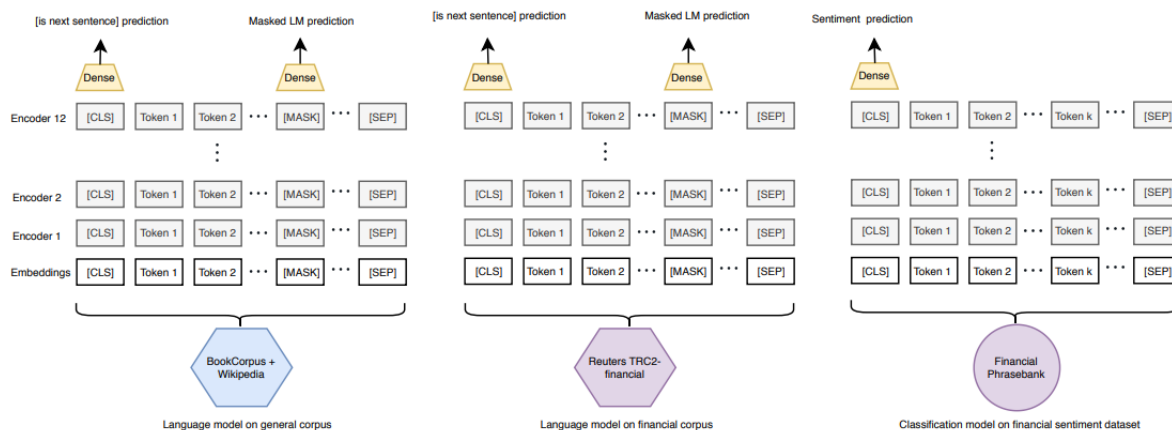


Рис. 3. Перший і другий етап переднавчання та налаштування моделі finBERT [22]

Порівняння результатів FinBERT та інших моделей

В роботі [22] було виконано порівняння чотирьох методів:

- Класифікатор LSTM з ембедінгом GloVe [25]
- Класифікатор LSTM з ембедінгом ELMo [16]
- Класифікатор ULMFit [17]
- FinBERT

Для оцінки моделей було застосовано три метрики: accuracy, cross-entropy loss та macro F1 average на одному наборі даних.

Результати FinBERT та інших методів при класифікації наборів даних Financial PhraseBank представлені в таблиці 1. Представлено результат для всього набору даних і для підвиборки зі 100% узгодженням експертів.

Таблиця 1
Результати роботи моделей на датасеті Financial PhraseBank у вигляді метрик [22]

Model	Всі дані			Дані 100% узг. експертів		
	Loss	Accuracy	F1 Score	Loss	Accuracy	F1 Score
LSTM	0.81	0.71	0.64	0.57	0.81	0.74
LSTM with ELMo	0.72	0.75	0.70	0.50	0.84	0.77
ULMFit	0.41	0.83	0.79	0.20	0.93	0.91
FinBert	0.37	0.86	0.84	0.13	0.97	0.95

Найгірше працює класифікатор LSTM без інформації про мовну модель зі статичним ембедінгом GloVe, показники метрик:

- cross-entropy loss = 0.81
- accuracy = 0.71
- macro F1 average = 0.64

Це пов'язано з тим, що в нейтральному класі алгоритм працює набагато краще.

Класифікатор LSTM з ELMo ембедингами демонструє кращі результати, ніж LSTM зі статичними ембедингами в усіх вимірних показниках, але має низький показник середнього значення F1 через погану роботу з менш представленими класами.

Показники метрик ULMFit значно перевершують дві попередньо розглянуті моделі. ULMFit працює набагато краще з дисбалансом класів, що відображується в покращенні значення метрики macro F1 average. Це свідчить про ефективність попереднього навчання мовної моделі.

FinBERT перевершує ULMFit, а отже, і всі інші методи за всіма показниками. Модель FinBERT продемонструвала найкращі результати серед реалізованих методів (LSTM і ULMFit) за всіма показниками метрик:

- cross-entropy loss = 0.37
- accuracy = 0.86
- macro F1 average = 0.84

Висновки

1. Досліджено різні підходи та методи семантичного аналізу настроїв в текстах з фінансовою тематикою, що стосуються поточного фінансового, іміджевого стану компаній з метою отримання висновків щодо поведінки вартості цінних паперів. Встановлено найбільш ефективний підхід – глибоке навчання. Досліджено результати порівняння чотирьох моделей глибокого навчання та результати їх роботи [22] у вигляді метрик cross-entropy loss, accuracy, macro F1 average на наборі даних з Financial PhraseBank.
2. Досліджено та встановлено особливості Transfer Learning (такі як здатність налаштування для конкретної вузької задачі з використанням значно меншої кількості навчальних даних та швидкість донавчання моделі) і переваги архітектури моделі BERT для вирішення задачі визначення емоційної забарвленості фінансових текстів.
3. Встановлено, що показники метрик моделей, які переднавчені та налаштовані з застосуванням мовних моделей для контекстуалізації, перевершують показники моделі зі звичайним статичним ембедингом. При проведенні дослідження [22] встановлено, що показники метрик cross-entropy loss, accuracy, macro F1 average моделі FinBERT є найкращими серед інших моделей для розв'язання задачі аналізу настроїв спеціалізованих фінансових текстів.

4. Результати систематизації та порівняльного аналізу підходів та методів призначені для застосування до прогнозування вартості фінансових активів компанії із застосуванням генеративних змагальних мереж GAN.

Література

1. E. Fama, "The Behavior of Stock Market Prices," *Journal of Business*, vol.38, pp.34-105, 1965.
2. M. G. Kendall, "The Analysis of Economic Time-Series-Part I: Prices," *Journal of the Royal Statistical Society*, vol.116, no.1, pp.11-34, 1953.
3. Burton G Malkiel. 2003. The Efficient Market Hypothesis and Its Critics. *Journal of Economic Perspectives* 17, 1 (feb 2003), 59–82.
4. Li Guo, Feng Shi, and Jun Tu. 2016. Textual analysis and machine leaning: Crack unstructured data in finance and accounting. *The Journal of Finance and Data Science* 2, 3 (sep 2016), 153–170
5. Mikolov, Tomas; Sutskever, Ilya; Chen, Kai; Corrado, Greg & Dean, Jeffrey (2013), Distributed Representations of Words and Phrases and their Compositionality
6. Bing Liu. 2012. Sentiment Analysis and Opinion Mining. *Synthesis Lectures on Human Language Technologies* 5, 1 (may 2012), 1–167.
7. Basant Agarwal and Namita Mittal. 2016. Machine Learning Approach for Sentiment Analysis. Springer International Publishing, Cham, 21–45.
8. Oscar Araque, Ignacio Corcuera-Platas, J. Fernando Sánchez-Rada, and Carlos A. Iglesias. 2017. Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Systems with Applications* 77 (jul 2017), 236–246.
9. Xiaodong Li, Haoran Xie, Li Chen, Jianping Wang, and Xiaotie Deng. 2014. News impact on stock price return via sentiment analysis. *Knowledge-Based Systems* 69 (oct 2014), 14–23.
10. Tim Loughran and Bill McDonald. 2016. Textual Analysis in Accounting and Finance: A Survey. *Journal of Accounting Research* 54, 4 (2016), 1187–1230
11. Tim Loughran and Bill McDonald. 2011. When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks. *Journal of Finance* 66, 1 (feb 2011), 35–65.
12. Mathias Kraus and Stefan Feuerriegel. 2017. Decision support from financial disclosures with deep neural networks and transfer learning. *Decision Support Systems* 104 (2017), 38–48.
13. Bernhard Lutz, Nicolas Pröllochs, and Dirk Neumann. 2018. Sentence-Level Sentiment Analysis of Financial News Using Distributed Text

- Representations and Multi-Instance Learning. Technical Report. arXiv:1901.00400
14. Babenko, Boris. "Multiple instance learning: algorithms and applications." View Article PubMed/NCBI Google Scholar (2008).
 15. Neel Kant, Raul Puri, Nikolai Yakovenko, and Bryan Catanzaro. 2018. Practical Text Classification With Large Pre-Trained Language Models. (2018).
 16. Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. (2018).
 17. Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Finetuning for Text Classification. (jan 2018).
 18. Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova . BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding / Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova – 2019 – Association for Computational Linguistics.
 19. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in Neural Information Processing Systems. / Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin – 2017 – p. 6000–6010
 20. Mike Shuster, Kaisuke Nakajima. Japanese and Korean Voice Search / Mike Shuster, Kaisuke Nakajima – Schuster et al. – 2012
 21. Wilson L Taylor. Cloze procedure: A new tool for measuring readability. / Wilson L Taylor – 1953 – Journalism Bulletin, 30(4):415–433.
 22. Dogu Tan Araci. FinBERT: Financial Sentiment Analysis with Pre-trained Language Models / Dogu Tan Araci – 2019
 23. Dan Hendrycks, Kevin Gimpel. Gaussian Error Linear Units (GELUs) / Dan Hendrycks, Kevin Gimpel – 2016
 24. Веб-сайт постачальника юридичних, державних, ділових і високотехнологічних джерел інформації. [Електронний ресурс]. - Режим доступу: <https://www.lexisnexis.com/>
 25. GloVe: Global Vectors for Word Representation / Jeffrey Pennington, Richard Socher, Christopher Manning – October 2014

УДК 519.688

К.т.н, доцент, Сирота С. В., старший викладач Копичко С.М.,
студент Бондаренко С.А.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ПІДСИСТЕМИ АНАЛІЗУ НАСТРОЇВ ТЕКСТОВИХ ДАНИХ ЗА ДОПОМОГОЮ ЗАСОБІВ ОБРОБКИ ПРИРОДНОЇ МОВИ

Abstract

Serhiy Syrota, docent; Serhiy Kopychko, senior lecturer;
Serhii Bondarenko, student.

*The system of analysis of moods of text data with the Natural
Language Processing tools*

This paper concerns the task of classification in product reviews. Semantic analysis, (or Mood Analysis) is the part of natural language processing techniques (NLP) responsible for identifying emotional coloring, in textual data. As a rule, this is used in publications in social networks, audits and customer reviews, in order to analyze and investigate the attitude of customers to an object, service or cases.

Вступ

Задача класифікації за емоційним забарвленням у відгуках користувачів є однією з основних в обробці природної мови. Серед методів класифікації, в класі методів обробки природної мови, існують ті, що дозволяють класифікувати неструктуровані дані за емоційним забарвленням. Основними методами в цій галузі є методи машинного навчання.

У цій статті розроблена програмна система для аналізу настроїв за допомогою засобів обробки природної мови, призначена для аналізу текстових даних відгуків на підписки на журнали Amazon.

Постановка задачі

Мета цього дослідження – аналіз, обробка за допомогою засобів обробки природної мови, неструктурованих текстових даних з подальшою їхньою класифікацією на предмет позитивного чи негативного емоційного настрою.

Предметом дослідження є математичне та програмне забезпечення системи класифікації емоційного контексту у відгуках на підписки на журнали Amazon.

Завданням є розробка системи із використанням методів обробки природної мови, представлення неструктурованих текстових даних у вигляді числових векторів з подальшою їх класифікацією за емоційним забарвленням.

Вимоги до системи:

- a) система передбачає роботу з повною версією тексту відгуку, та його короткого резюме;
- b) система виділяє дві категорії класифікації відповідно до відповіді на питання: «негативний відгук чи ні?»;
- c) межею визначення хорошого відгуку для системи є оцінка 0.5 і вище;
- d) система може бути розширена для використання на інших типів задач.

Термінологія

ОПМ- Обробка природної мови (*NLP* - *Natural Language Processing*) - напрям математичної лінгвістики, інформатики та штучного інтелекту, який вивчає проблеми комп'ютерного аналізу та синтезу природної мови.

TF-IDF (*TF* — *term frequency*, *IDF* — *inverse document frequency*) — статистичний показник, що використовується для оцінки важливості слів у контексті документа, що є частиною колекції документів чи корпусу. Вага (значимість) слова пропорційна кількості вживань цього слова у документі, і обернено пропорційна частоті вживання слова у інших документах колекції.

ROC-крива (*Receiver Operating Characteristic*) – графік, отриманий за допомогою значень *TPR* та *FPR*. Дозволяє оцінити якість бінарної класифікації.

AUC (*Area Under Curve*) – площа під графіком *ROC*-кривої.

AP (*Average Precision*) — площа під графіком кривої *Precision Recall*.

Компонентна модель системи

Представлений у роботі класифікатор є частиною більш загальної системи прогнозування відтоку клієнтської бази з урахуванням даних, отриманих із відгуків. На рисунку 1 зображена модель системи прогнозування відтоку клієнтської бази з урахуванням даних, отриманих із відгуків, та деталізована діаграма процесів другого рівня з уточненням кожного рівня в нотації *UML*.

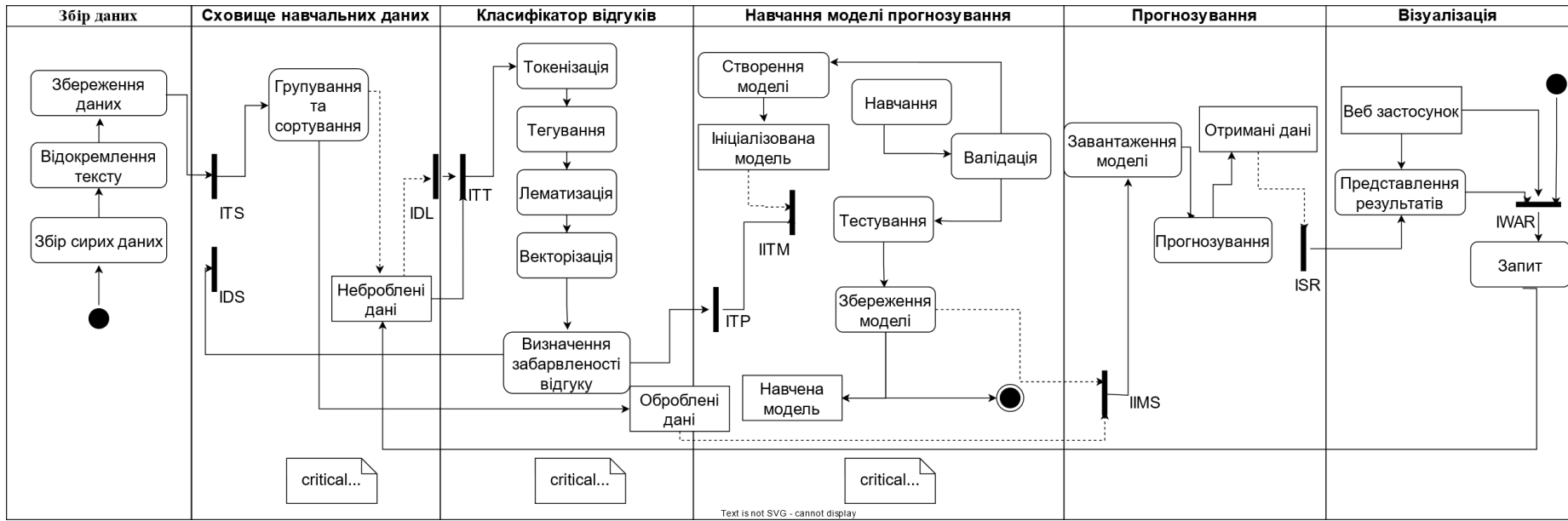


Рис. 1. Модель системи прогнозування відтоку клієнтської бази з урахуванням даних, отриманих із відгуків

Як видно з представленого рисунку, класифікатор відгуків є безпосередньою складовою системи, зі своїми внутрішніми процесами, тоб

Опис алгоритму

Завантаження і попередня обробка даних

Процес починається із завантаження вхідних даних. Навчальні дані представляють собою експертні оцінки (числові значення від 1 до 5) та тексти відгуків на товари. Відгук складається із двох частин – тексту відгуку і короткої рецензії, тому необхідно відокремити їх і опрацювати паралельно в рамках одного набору неструктурованих даних.

Необхідно провести очищення неструктурованих даних для підготовки їх до моделювання за допомогою засобів та методів обробки природної мови. Для очистки даних використовується функція «clean_text», яка виконує операції попередньої обробки тексту, зокрема:

- Заміна регістру слів на малий
- Лексемізація текст і видаляє знаки пунктуації
- Усунення слів із цифрами
- Видалення стоп-слів (the, this, a...).
- Тегування частин мови
- Лематизація слів у тексті

Реалізація функції аналізу настроїв

За допомогою лексикону слів модуля Vader, що є частиною модуля NLTK, відповідального за аналіз настроїв, можна визначити позитивні чи негативні слова, які є у тексті. Він також враховує контекст речень, щоб визначити оцінку настроїв. Для кожного тексту Vader повертає 4 значення: нейтральність, позитив, негатив та загальна оцінка. Для повної і короткої версій відгуків, отримані значення інтегруються в набір даних.

Реалізація функції векторного представлення тексту

Модель Doc2Vec перетворює кожне слово в корпусі слів на числовий вектор, для подальшого використання цих представлень у моделі. Це робиться з використанням контекстів, в яких ці слова з'являються, за допомогою нейронних мереж. Подібні за семантикою слова будуть ближче розташовані у семантичному просторі і, відповідно, будуть мати подібні векторні представлення. Ця концепція працює і на рівні текстів.

Застосувавши попередньо-навчену модель Doc2Vec до вхідних даних, можна отримати відповідні векторні представлення.

Застосування метрики TF-IDF

Звичайне перерахування слів у тексті не є доцільним, оскільки таким чином неможливо врахувати відносну «важливість» кожного слова у тексті. Зокрема можливі випадки, коли слова, які часто зустрічаються

Рис. 3. Оцінка відгуків. а) Найімовірніші позитивні. б) Найімовірніші негативні

Далі треба дослідити розподіли цих імовірностей. Результати показані на графіках рисунку 4, а) – для повних версій текстів, і б) – для скорочень.

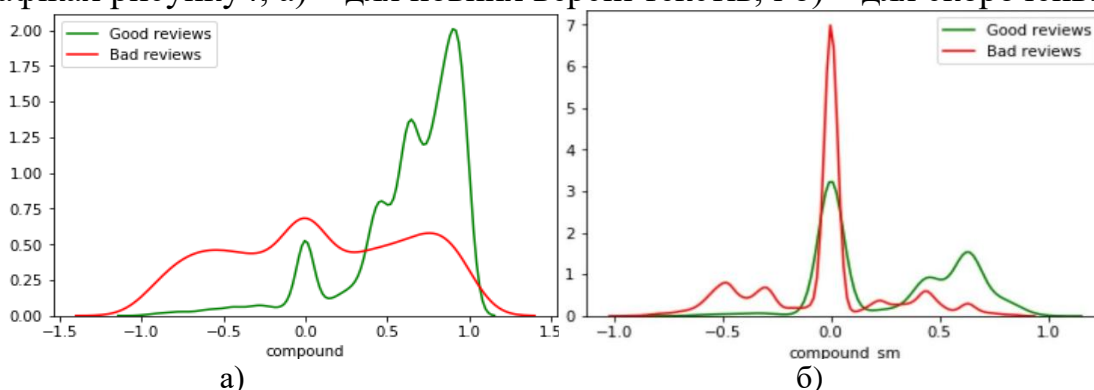


Рис. 4. Графіки щільності розподілу настроїв. а) Для повних текстів відгуків. б) Для коротких рецензій

На цих графіках зображено розподіли настроїв повних та коротких версій відгуків за результатами роботи моделі. Видно, що для повних версій текстів, істино-позитивні відгуки в більшості модель теж вважає їх ймовірно позитивними, а істино-негативні відгуки, навпаки, мають низьку ймовірнісну оцінку. Водночас, для скорочених версій відгуків, модель передбачає в більшості нейтрально-визначені відгуки як ймовірно-негативні, чи ймовірно -позитивні і набагато менше випадків правильного визначення істинних значень.

Реалізація класифікатора

На цьому етапі формуються дві вибірки з набору даних, одна для тренування моделі, інша для тестування.

У цій роботі для здійснення прогнозів використовується класифікатор на основі методу градієнтного підсилювання (XGBClassifier). Необхідно виділити найважливіші параметри для навчання. Результати показані на рисунку 5.

У виділений перелік найвагоміших характеристик входять зокрема результати попереднього аналізатора (Vader+Doc2Vec) і ті слова, які часто зустрічались у тексті.

	feature	importance
18	doc2vec_sm_vector_1	0.053122
21	doc2vec_sm_vector_4	0.049896
17	doc2vec_sm_vector_0	0.045567
19	doc2vec_sm_vector_2	0.045292
16	doc2vec_rw_vector_4	0.044662
12	doc2vec_rw_vector_0	0.043954
20	doc2vec_sm_vector_3	0.041868
15	doc2vec_rw_vector_3	0.040491
13	doc2vec_rw_vector_1	0.036123
9	nb_chars_sm	0.035139
3	compound	0.034825
14	doc2vec_rw_vector_2	0.031519
0	neg	0.029158
2	pos	0.027938
1	neu	0.027112

Рис. 5. Найвагоміші характеристики

Валідація

Після навчання класифікатора були отримані наступні результати, зображені на рисунку 6. Рисунок а) ілюструє оцінки навчання моделі, рисунок б) - ROC-криву.

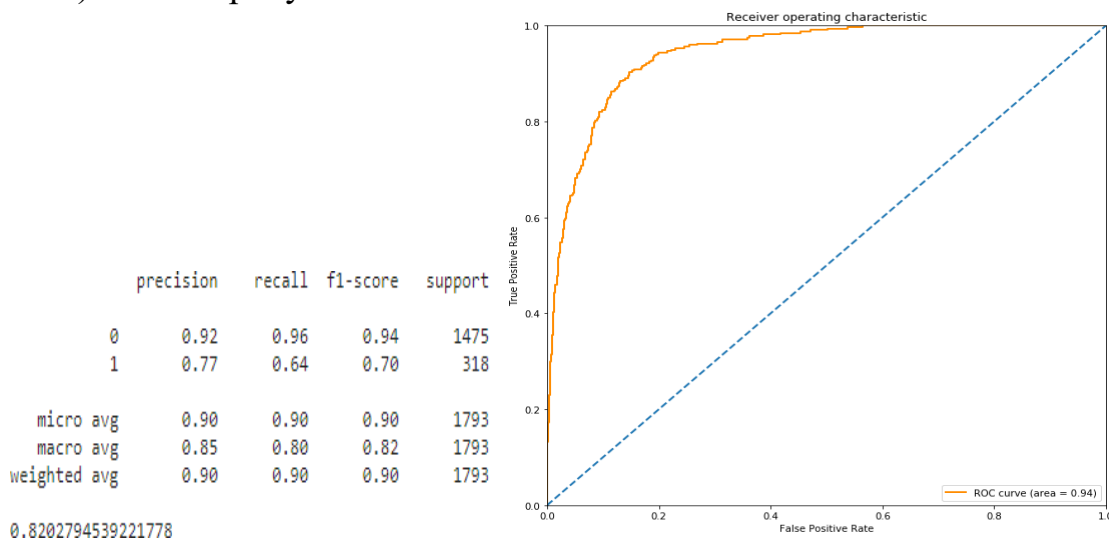


Рис. 6. Результати навчання моделі. а) оцінки моделі. б) ROC-крива для оцінки моделі.

Як можна побачити, загальна оцінка роботи класифікатора метрикою $F = 0.827$, що є дуже непоганим результатом для такого незбалансованого набору даних. Побудована ROC-крива, хоч і має велику площу під графіком - AUC ROC, але не підходить для оцінки якості роботи моделі. Це пов'язано з незбалансованістю набору даних.

Згідно з формулою хибно позитивного коефіцієнта, яка відповідає осі x ROC-кривої:

$$FPR = \frac{FP}{TN + FP},$$

де $TN+FP$ відповідає позитивним рецензіям, кількість яких набагато вища за кількість негативних, оскільки набір даних не є збалансованим. Це означає, що навіть при деяких хибно-позитивних спрацьовуваннях, хибно-позитивний коефіцієнт буде низьким. Тобто модель буде робити багато хибно-позитивних прогнозів і все одно матиме низький рівень хибно-позитивних результатів, одночасно збільшуючи істинно-позитивний показник і, отже, штучно збільшуючи показник AUC ROC. Для оцінки ефективності потрібен інший показник, а саме Average Precision. Для цього необхідна крива Precision Recall, результат побудови якої можна побачити на рисунку 7.

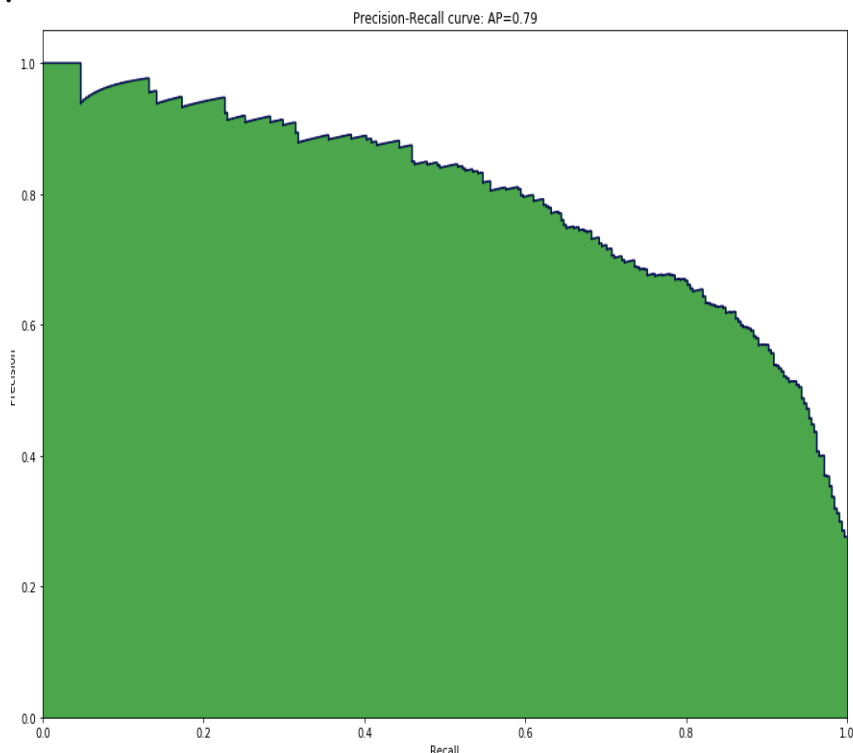


Рис. 7. Крива Precision Recall

Зображена крива Precision Recall, а точніше, площа під цією кривою (Average Precision) більш точніше підходить для оцінки точності моделі, і ось чому. Значення Precision зменшується, коли збільшується Recall. Це означає, необхідно вибрати поріг прогнозування, в залежності від потреб. Якщо метою є високе значення Recall, треба встановити низький поріг передбачення, який дозволить виявити більшість спостережень позитивного класу, але з більш низькою точністю. Якщо ж необхідна висока якість прогнозування і при цьому є можливість знехтувати низкою TP, тоді має сенс поставити високий поріг, який дасть більшу точність і менше значення Recall.

Отримане значення $AP = 0,79$

Висновки

1. Було досліджено наявні застосунки та теоретичні методи ідентифікації оціночних суджень і розроблено класифікатор, який дозволяє їх визначати.
2. Встановлено, що особливістю даного класифікатора є насамперед кращі показники тренування при незбалансованості вхідних даних. У порівнянні з іншими класифікаторами, отримані результати є досить суттєвими, оскільки для, наприклад класифікатора випадкових значень, значення Precision у якого буде дорівнювати частці позитивних спостережень (4,3%) Для кожного Recall точність залишиться сталою, і це означає що AP у такому випадку $= 0,043$. AP розглянутої моделі становить приблизно 0,8, що більш ніж у 18 разів більше, за AP випадкового методу. При застосуванні класифікатора Random Forest результати AP виявились приблизно 0,33, що в порівнянні з розглянутої моделлю майже в 2,5 рази менше. Отже, модель має хорошу прогностичну силу.
3. Запропоновано застосування класифікатора для дослідження поведінки користувачів в компаніях, які постачають послуги, або для аналізу настроїв в середовищах, де певні соціальні групи виражають свою думку, у соціальних мережах, тощо.
4. Заплановано застосування результатів дослідження і розробленого програмного забезпечення в рамках магістерської дисертації.

Література

1. *Z. Madhoushi*: Science and Information Conference (SAI). Sentiment analysis techniques. 2015. 10.1109/SAI.2015.7237157
2. *J. Devlin*: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2019.
3. *Hochreiter S*: Short-Term Memory. Neural Computation. 1997. Vol. 9. P. 1735-1780.
4. *Shaobo Li*: «DeepPatent: patent classification with convolutional neural networks and word embedding». Scientometrics — 2018 — Vol. 117 — P. 721–744.
5. *Quoc V. Le, Tomas Mikolov*: «Distributed Representations of Sentences and Documents». Scientometrics — 2014 — P.3-6.

Доцент Соловйов С.О., ст. Викладач Копичко С.М.,
студентка Страшнова А. С.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІДЕНТИФІКАЦІЇ ВЛАСНИКА ГОЛОСУ НА ОСНОВІ АРХІТЕКТУРИ СІАМСЬКИХ НЕЙРОННИХ МЕРЕЖ

Abstract

Serhii Soloviov, assoc.prof., PhD, DrSc; Sergei Kopychko, senior lector;
Anastasiia Strashnova, student;

Mathematical and software systems for speaker verification using Siamese Neural

This article proposes an efficient way to build a Siamese Neural Network model for solving the task of speaker verification. Siamese Neural Network is a type of neural network architecture containing two identical subnetworks that share parameters and weights. In the described model we will use Convolutional Neural network architecture as an architecture for two sub-models.

Вступ

Оскільки розпізнавання мовлення останнім часом набирає значної популярності, розробники намагаються імплементувати дану систему в свої програмні засоби. Тому виникає необхідність надання лише деяким користувачам доступ до програмного забезпечення, тим самим це створює потребу в системі ідентифікації голосу.

В кожній людини наявні свої особливі вокальні характеристики, що визначаються індивідуальною структурою її голосового апарату. На рівні підсвідомості людина здатна ідентифікувати голоси, але розробка автоматичного класифікатора супроводжується великою кількістю розрахунків, а моделі потребують великих об'ємів вхідних даних для їх навчання, при цьому за необхідності додавання нового класу - повторного навчання на новому датасеті. В даній роботі було розглянуто основи застосування нейронних мереж для розпізнавання власника голосу за аудіозаписом. З ціллю мінімізувати кількість вхідних даних, а також уникнення повторного навчання моделі при додаванні нових класів, архітектурою моделі було обрано сіамську нейронну мережу, яка використовує унікальну структуру природного пошуку подібностей між вхідними даними.

Постановка задачі

Метою дослідження є створення такої моделі, що здатна до класифікації власника голосу за коротким аудіосигналом.

Предметом дослідження є математичне та програмне забезпечення ідентифікації голосу за спектрограмою аудіосегмента.

Об'єктом дослідження є методи, моделі, алгоритми, процеси їх оптимізації та налаштування для ідентифікації власника голосу. Існуючі рішення в галузі ідентифікації людського мовлення.

Методи обробки аудіосигналу, метод перетворення зображення у вектор, методи побудови моделей нейронних мереж, методи машинного навчання, python розробки.

Кінцевим результатом є математичне та програмне забезпечення для ідентифікації голосу за спектрограмою аудіосегмента. Визначення найбільш ефективних підходів для ідентифікації голосу. Отримані результати були порівнянні з існуючими аналогами. Продемонстровано конкурентність моделі.

Призначення. Покращення роботи існуючого програмного забезпечення ідентифікації голосу за спектрограмою аудіосегмента для покращення безпеки користування додатками розпізнавання мовлення.

Технічні вимоги. Модель повинна отримувати на вхід дві спектрограми двох аудіосигналів, розрахувати схожість між сигналами та надати на вихід бінарний результат, що буде вказувати на приналежність цих прикладів до одного класу.

Теоретичні відомості

Для вирішення поставленої задачі потрібно обрати модель, яка буде здатна навчитись розпізнавати голоси на невеликій вибірці даних, а також при необхідності робити прогнози на записах голосів нових спікерів. Тому замість типового класифікатора, постає необхідність в побудові моделі, здатної до порівняння аудіосигналів між собою та прийняття рішення приналежності цих записів до одного класу. А отже модель потребує як мінімум 2 окремі субмоделі, що будуть обробляти різні вхідні дані та в результаті шукати відстані між ними.

Сіамська нейронна мережа є класом архітектур нейронних мереж, що складаються з двох ідентичних субмереж, які на вхід отримують два різних зображення, та об'єднуються функцією втрат (loss function). Обрана функція втрат обчислює певну метрику між представленнями ознак найвищого рівня вхідних зображень. Конфігурації параметрів та вагів дублюються для двох субмереж, це гарантує розподілення схожих зображень поряд у просторі функцій.

Представлення голосу подається до моделі у вигляді спектрограми, тому для роботи з зображенням (спектрограмою) було обрано згорткову нейронну мережу, адже ця мережа є однією з передових в області комп'ютерного зору та підкріплена великою кількістю наукових та емпіричних досліджень [1]. А отже в кожній з субмереж кінцевої розробленої моделі знаходиться шари згортки. Архітектура побудованої субмережі має вигляд зображений на рисунку 1.

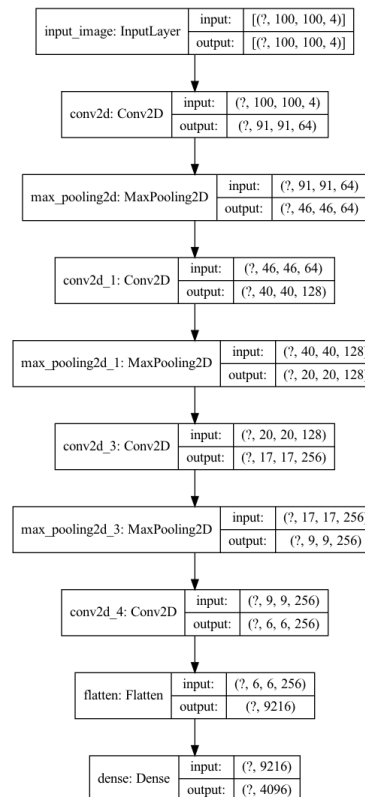


Рис. 1. Архітектура субмережі

На вході в модель отримує 3D тензор, розмірності 100 x 100 x 4, відповідно ширина, висота та кількість каналів зображення, де 4 канали зображення це колірний простір RGB та альфа канал.

Після чого дані потрапляють у перший шар згортки. Шари згортки це складова побудови згорткових нейронних мереж. Такі шари складаються з багатьох фільтрів (вагових матриць), що рухаються вздовж зображення зліва на право та беруть за вхідні дані деяку область зображення. Вони мають свою розмірність та глибину, що відповідає глибині вхідного зображення.

Такі фільтри активуються тоді, коли поелементне множення призводить до високих значень, що повідомляє мережу про наявність краю чи плями кольору на зображенні.

Розмірність вихідної карти ознак розраховується наступним чином [2]:

$$G[m,n] = (f \cdot h)[m,n] = \sum_j \sum_k h[j,k] f[m-j, n-k], \quad (1)$$

де f - вхідне зображення, h - фільтр, а m і n - індекси рядків і стовпців вихідної матриці.

Гіперпараметрами фільтрів являються крок (*stride*) та відступ (*padding*). Крок - це розмір «кроку» фільтра, а відступ виконує функцію розширення по краям вхідного зображення, щоб вихідний шар зберігав розмірність вхідного, заповнюючи комірки нульовими значеннями.

Розмір вихідного зображення можна розрахувати формулою (2).

$$\frac{W - F + 2P}{S + 1}, \quad (2)$$

де W - розмір вхідного зображення, F - розмір фільтрів, P - кількість відступів, S - крок фільтра.

Для розробленої моделі було обрано функцію активації (функція, що вирішує чи варто активувати нейрон, розраховуючи ваги та додаючи зміщення) ReLu (3), зважаючи на швидкість та зручність її обчислення.

$$f(x) = \max\{0, x\} \quad (3)$$

Між кожним шаром згортки знаходиться шар субдискретизації MaxPooling, що поєднує певну кількість вихідних, з попереднього шару, нейронів та об'єднує їх в один нейрон, тим самим зменшуючи розмірність даних. Його параметри, як і параметри шарів згортки, були підібрані під час оптимізації гіперпараметрів.

Всі нейрони останнього згорткового шару об'єднуються в один вектор (повнозв'язний шар), за яким слідує функція втрат. Так як наша класифікація бінарна, на цьому рівні ми обираємо функцію активації Sigmoid.

Більш детально прогноз розраховується за формулою (4) [3].

$$p = \sigma(\sum_j \alpha_j |h_{1,L-1}^{(j)} - h_{2,L-1}^{(j)}|), \quad (4)$$

де σ - Сигмоїдальна (Sigmoid) функція активації, $h_{1,L-1}^{(j)}$ та $h_{2,L-1}^{(j)}$ - фінальні шари обох субмереж, α_j - додатковий параметр, який параметризується під час навчання моделі, розраховуючи важливість відстаней відносно компонентів. З чого слідує, що шар h_L - повнозв'язний шар мережі, що

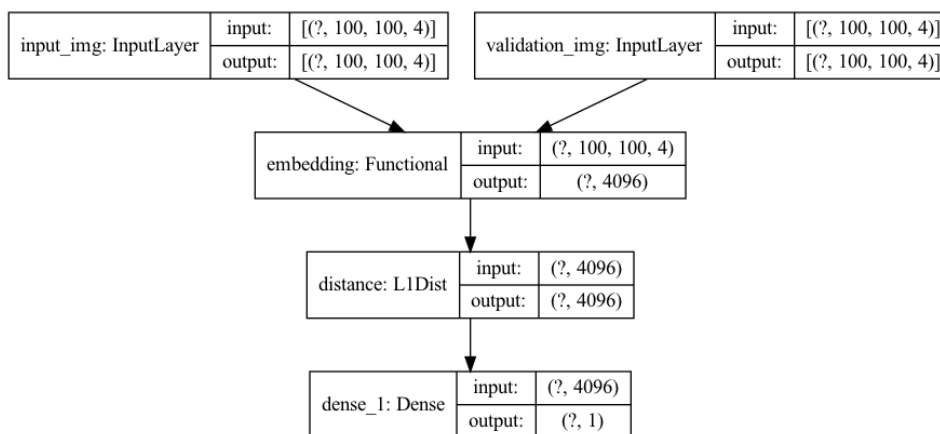


Рис. 2. Архітектура мережі

об'єднує дві Сіамські підмережі. На рисунку 2. зображено архітектуру моделі Сіамської нейронної мережі.

Функція втрат (Loss function). В розробленій моделі було запропоновано використання функції Contrastive Loss, що використовується для навчання на вкладеннях (embeddings) в яких, вхідні дані, що належать одному класу мають малу Евклідову відстань, а різнокласові - максимальну. Дана функція розраховується за формулою (5) [4].

$$(1 - Y) \frac{1}{2} (Dw)^2 + (Y) \frac{1}{2} \{ \max(0, m - Dw) \}^2, \quad (5)$$

де $Dw = \sqrt{\{Gw(X_1) - Gw(X_2)\}^2}$ - Евклідова відстань,

$Gw(X_-)$ - вихідний вектор ознак одного з вхідних випадків, Y - бінарний результат.

Оптимізація гіперпараметрів. Було використано Баєсову оптимізацію для оптимальної селекції гіперпараметрів та підбору кількості згорткових шарів. Метод послідовно, на різних комбінаціях гіперпараметрів з множини заданих значень, навчає модель. Потім, на основі отриманих результатів, відбувається підбір наступної комбінації, що повинна очікувано призвести до покращення. Для кожного згорткового фільтра та шару MaxPooling був обраний інтервал, в якому знаходиться значення його розміру, від 2x2 до 10x10 та кількістю фільтрів в межах від 16 до 256 з кроком 16. Фільтри повнозв'язних шарів знаходились у інтервалі між 256 і 4096 з кроком 16.

В результаті оптимізації було отримано наступні значення, зображені в таблиці 1. Кількість шарів згортки також було отримано під час оптимізації.

Таблиця 1

Параметри субмережі

Шар	Кількість фільтрів	Розмір фільтра
1 Шар згортки	64	10 x 10
2 Шар згортки	128	7 x 7
3 Шар згортки	128	4 x 4
4 Шар згортки	256	4 x 4
Шар MaxPooling	64	2 x 2

Навчання мережі. Для навчання моделі було використано датасет DAPS (Device and Produced Speech) [5] в якій містились 5 аудіозаписів 20 спікерів (10 жінок та 10 чоловіків) тривалістю в 2 хвилини.

Обробка аудіоданих відбулася наступним чином:

1. З кожного аудіозапису було видалено аудіо сегменти, гучність яких не досягала 20дБ. Це було виконано для уникнення навчання моделі на порожніх даних.
2. Випадковим чином з кожного аудіозапису було отримано 2 сегмента тривалістю 4 сек.
3. Всі записи було перетворено у спектрограми формату png.

Тренувальні дані мають наступний вигляд [Спектрограма1, Спектрограма2, бінарний результат]. Якщо два вхідних зображення відповідали одному класу (один спікер) бінарний результат дорівнював 1. Далі дані випадковим чином були розміщені у тренувальний, тестовий та валідаційний датасет, відповідно у пропорції 70%, 20% і 10%. Приклади вхідних даних мережі зображені на рисунку 3.

Кожна модель навчалась з максимальною кількістю в 50 епох. Модель-лідер, конфігурація гіперпараметрів якою була отримана Баєсовою оптимізацією, досягла найкращих результатів на 40 епосі, після чого навчання було зупинене. Значення параметрів відображені в таблиці 1.

Результати. Результати прогнозу мережі на тренувальному та тестовому наборі даних відображені у таблиці 2.

Метрики, використані для аналізу результатів описані нижче:

Точність (Precision) - це відношення всіх правильно класифікованих зображень до суми правильно класифікованих та помилково класифікованих.

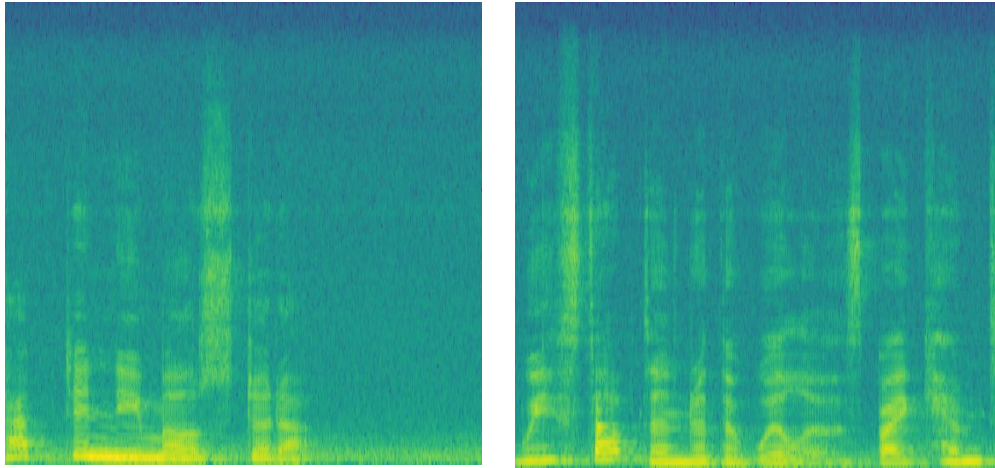


Рис. 3. Вхідні дані мережі

Recall - це відношення всіх правильно класифікованих зображень до суми правильно класифікованих та помилково не класифікованих (тобто тих що відносяться до цього класу, але не були розпізнані).

EER - рівний коефіцієнт помилок (equal error rate)

Таблиця 2

Метрики мережі на тренувальних та тестових даних

Метрики (%)	На тренувальних даних	На тестових даних
Precision	95	89.4
Recall	96.3	99
Accuracy		94.16

Порівняння результатів з існуючими моделями. Для порівняння були обрані сучасні лідери у розпізнаванні голосу. Обидві моделі використовували великі датасети (понад 500 000 тренувальних даних). Перша модель [3] використовує архітектуру Залишкової нейронної мережі (ResNet), а друга [4] - представлення ознак і-го вектору з імовірнісним лінійним дискримінантним аналізом (PLDA).

Результати розпізнавання голосу моделями

Модель	Датасет	Кількість тренувальних даних	Кількість спікерів	EER (%)
Siamese network	DAPS	200	20	3.4
ResNet-50 [6]	VoxCeleb1-E	581,480	1,251	4.42
I-vectors + PLDA (1) [7]	VoxCeleb1	552,536	90	8.8

Як видно з Таблиці 3. Високу точність класифікації було досягнуто за незначну кількість епох, у порівнянні з іншими моделями. Звісно варто взяти до уваги, що така висока точність була досягнута через невелику кількість даних, але варто пам'ятати особливість розробленої моделі - здатність прогнозувати на невідомих класах.

Висновки

1. Даній статті було досліджено модель ідентифікації власника голосу та підбір найбільш ефективних параметрів моделі архітектури сіамської мережі з субмоделями згорткової нейронної мереж.

2. Запропоновано спосіб застосування моделі архітектури сіамської мережі, що дозволяє порівнювати аудіосигнали між собою та прийняти рішення приналежності цих записів до одного класу. Що може бути використано для оптимального і простого способу ідентифікації голосів нових спікерів.

3. В результаті порівняльного аналізу побудованої моделі було показано, що модель досягає точності за показниками Precision = 89.4%, Recall = 99% та Accuracy = 94.16%. А коефіцієнт помилок = 3.4, що є меншим у порівнянні з моделями-лідерами у даній сфері.

4. В подальшому планується повторне навчання моделі на більшому об'ємі даних, для покращення точності класифікації, а також тестування моделі на нових класах.

Література

1. Ebrahimi, Amir et al. "Convolutional neural networks for Alzheimer's disease detection on MRI images." *Journal of Medical Imaging* 8 (2021): 024503 - 024503.

2. Arif, Sheeraz et al. “3D-CNN-Based Fused Feature Maps with LSTM Applied to Action Recognition.” *Future Internet* 11 (2019): 42.
3. Sumit Chopra, Raia Hadsell and Yann LeCun., “Learning a similarity metric discriminatively, with application to face verification.» 2005.
4. Koch, G., Zemel, R. & Salakhutdinov, R., Siamese Neural Networks for One-shot Image Recognition, 2015, c. 2-5.
5. Gautham J. Mysore, “Can We Automatically Transform Speech Recorded on Common Consumer Devices in Real-World Environments into Professional Production Quality Speech? - A Dataset, Insights, and Challenges”, in the IEEE Signal Processing Letters, Vol. 22, No. 8, August 2015
6. Joon Son Chung , Arsha Nagrani, Andrew ZissermanVoxCeleb2, Deep Speaker Recognition.
7. A. Nagrani, J. S. Chung, and A. Zisserman, “VoxCeleb: a largescale speaker identification dataset,” in INTERSPEECH, 2017.

УДК 519.688

к. ф.-м. н., доцент Третиник В.В., студент Бондарчук О. О.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

МЕТОД ВИРІШЕННЯ ПРОБЛЕМИ ХОЛОДНОГО СТАРТУ ДЛЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ФІЛЬМІВ

Abstract

Violeta Tretynyk, assoc. prof., PhD; Oleksandr Bondarchuk, student
Method of Solving the Cold Start Problem for a Movie Recommender System

This article discusses the algorithm and its application for the system to provide recommendations to new users who do not have records in the system. To solve the problem, an ensemble machine learning algorithm based on decision trees, XGBoost, was used. Approaches that allow recommender systems to take into account user similarity, movie similarity, and matrix factorization are also considered. This solution can benefit the business in many ways, such as preventing the loss of new users due to low recommendation accuracy.

Вступ

Рекомендаційні системи допомагають користувачам знаходити та вибрати потрібні елементи: фільми, книги, готелі із великої кількості доступних в інтернеті електронних джерел інформації [1]. Враховуючи великий набір об'єктів та опис потреб користувачів, сервіси пропонують користувачеві невелику кількість елементів, яка добре підходить до опису. Так само система рекомендацій фільмів забезпечує рівень комфорту та персоналізації. Допомагає користувачам краще взаємодіяти з системою та дивитися фільми, які відповідають їхнім потребам. Але, незважаючи на широке застосування рекомендаційних систем, однією складною та поширеною проблемою є проблема холодного запуску, коли для певних користувачів або елементів невідомі попередні події, такі як рейтинги чи кліки. Це може стати проблемою, що в подальшому може призвести до втрати нових користувачів через низьку точність рекомендацій на ранніх стадіях. Зосередившись саме на даній проблемі, можна суттєво збільшити кількість нових користувачів сервісів, які використовують рекомендаційні системи.

Постановка задачі

Як вхідний набір даних береться розмічений масив, тобто вибірка даних, в якій кожен її об'єкт містить рейтингове значення фільму. Задача полягає в тому, щоб отримати рекомендації для наших користувачів, спрогнозувавши їхні рейтинги для фільмів, які вони ще не дивилися. Для цього потрібно буде використовувати минулі записи про фільми та оцінки користувачів, щоб передбачити їхні майбутні оцінки. Задача ускладнюється, якщо буде мала кількість даних і не вистачатиме екземплярів для кожного класу жанру фільму чи автора.

Об'єктом дослідження є методи побудови рекомендаційних систем: контент-орієнтовані (content-based), колаборативної фільтрації (collaborative filtering), гібридні методи (hybrid), методи засновані на матричному розкладанні (matrix factorization). Метрики оцінки точності рекомендаційних систем: MAPE, RMSE. Проблема холодного старту в рекомендаційних системах, коли є недостатня кількість даних для коректної роботи рекомендаційної системи (наприклад, коли додається новий товар чи користувач). Метрики схожості в подібності двох користувачів: косинусна відстань.

Метою роботи є дослідження та розробка ефективного математичного забезпечення для рекомендаційної системи фільмів, що дозволить підбирати рекомендації, які матимуть велику точність з прийнятним рівнем релевантності в умовах великої кількості нових користувачів за неповної або відсутньої кількості інформації про їх уподобання.

Опис набору даних

Як вхідний набір даних для перевірки можливостей методу, що пропонується, було обрано набір даних "MovieLens" з відкритим кодом від "GroupLens" [2]. Цей набір даних містить 100 тисяч точок даних різних фільмів і користувачів. Ми будемо використовувати три стовпці даних:

- "userId" – це ідентифікатор користувача;
- "movieId" – це ідентифікатор фільму;
- "rating" – це значення балу по п'ятибальній шкалі, який поставив користувач даному фільму.

Також, для отримання минулих записів, будемо використовувати метадані:

- 1) для нових користувачів ми можемо використовувати їх місце розташування, вік, стать, браузер і пристрій користувача, щоб передбачити рекомендації.

- 2) для нових фільмів ми можемо використовувати жанр, акторський склад і знімальну групу, щоб рекомендувати їх цільовим користувачам.

Але для прогнозу доведеться ще виконати конструювання ознак. Створимо ознаки, які представляють глобальні середні значення, ознаки, які представлятимуть п'ятірку найбільш схожих користувачів по відношенню до обраного користувача та ознаки, які представлятимуть п'ятірку найбільш схожих фільмів по відношенню до обраного.

Опис алгоритму побудови моделі

Алгоритми машинного навчання в системах рекомендацій зазвичай поділяються на дві категорії: системи на основі вмісту та системи спільної фільтрації. Сучасні рекомендаційні системи поєднують обидва підходи.

1. Системи рекомендацій фільмів на основі вмісту

Методи, засновані на вмісті, засновані на подібності атрибутів фільму. Використовуючи цей тип системи рекомендацій, якщо користувач дивиться один фільм, йому рекомендуються схожі фільми. Наприклад, якщо користувач дивиться комедію з Джонні Деппом у головній ролі, система порекомендує йому фільми в тому ж жанрі або з тим самим актором, або обидва. Зважаючи на це, входними для побудови системи рекомендацій на основі вмісту є атрибути фільму.

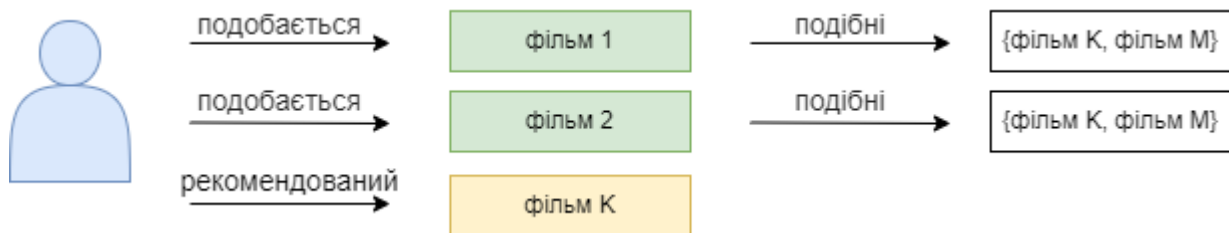


Рис. 1. Огляд системи рекомендацій на основі вмісту

2. Спільна фільтрація систем рекомендацій фільмів

Завдяки спільній фільтрації система базується на попередніх взаємодіях між користувачами та фільмами. Зважаючи на це, входні дані для спільної системи фільтрації складаються з минулих даних взаємодії користувачів із фільмами, які вони переглядають.

Наприклад, якщо користувач А дивиться М1, М2 і М3, а користувач В дивиться М1, М3, М4, ми рекомендуємо М1 і М3 подібному користувачеві С. Ці дані зберігаються в матриці, яка називається матрицею взаємодії між користувачем і фільмом, де рядки – це користувачі, а стовпці – фільми.

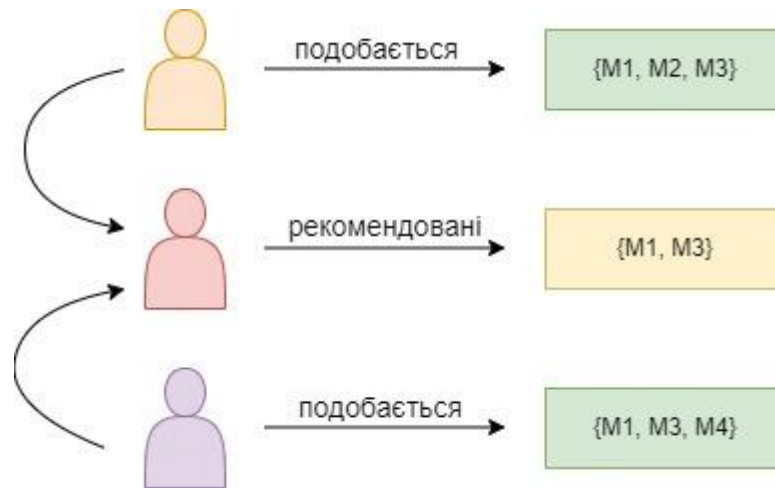


Рис. 2. Приклад системи рекомендованих фільмів спільної фільтрації

Пропонований алгоритм для вирішення проблеми холодного старту в розмічених даних складається з двох послідовних кроків:

- 1) побудувати модель на основі факторизації матриці;
- 2) створити остаточну прогноуючу модель.

Першим кроком методу є побудова моделі на основі факторизації даних. Матрична факторизація – це клас алгоритмів спільної фільтрації, які використовуються в рекомендаційних системах [3]. Алгоритми матриці факторизації працюють шляхом розкладання матриці взаємодії користувач-фільм на добуток двох прямокутних матриць нижчої розмірності, скажімо, U та M . Розкладання виконується таким чином, що добуток призводить до майже подібних значень до матриці взаємодії користувача-фільму. Тут U представляє матрицю користувачів, M представляє матрицю фільмів, n – кількість користувачів, а m – кількість фільмів. Кожен рядок матриці користувача представляє користувача, а кожен стовпець матриці фільму представляє фільм.

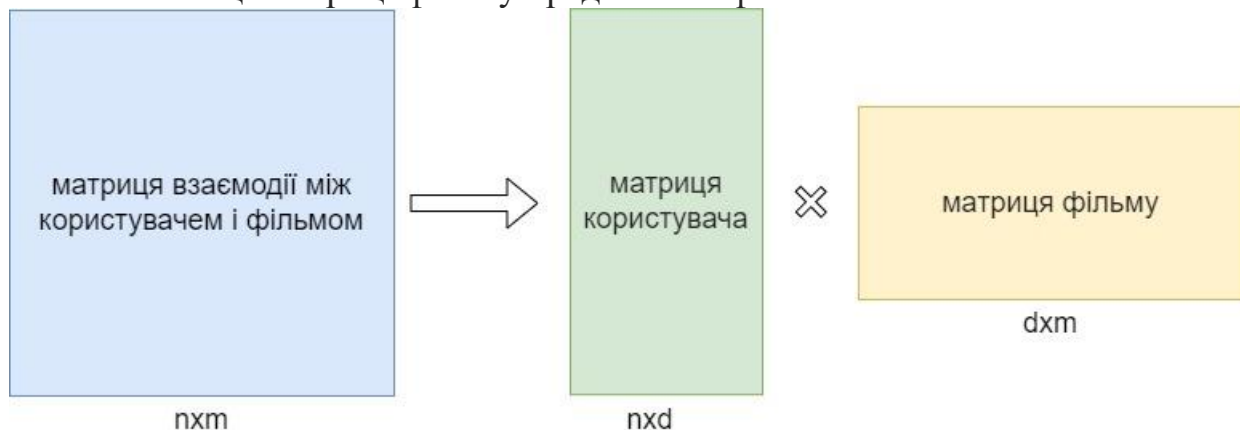


Рис. 3. Матриця факторизації

Отримавши матриці U і M на основі непорожніх клітинок у матриці взаємодії користувач-фільм, виконуємо добуток U і M і прогнозуємо значення непорожніх клітинок у матриці взаємодії користувач-фільм. Збережемо ці прогнози, щоб передати їх у остаточну модель як додаткову функцію. Це допоможе нам включити спільну фільтрацію в нашу систему.

Другим кроком буде побудова прогнозуючої моделі, в даному випадку буде використовуватись алгоритм "XGBoostRegressor" [4]. XGBoost – це ансамблевий алгоритм машинного навчання на основі дерева рішень, який використовує структуру посилення градієнта. Він має гарні результати на структурованих, табличних даних малого та середнього розміру, алгоритми на основі дерева рішень вважаються найкращими в своєму класі. Його алгоритм має такий вигляд:

$$L^{\lambda}(f) = \sum_{i=1}^n l(y_i, \hat{y}_i) + \frac{1}{2} \sum_{j=1}^m \omega_j^2 \sum_{i=1}^n f_j(x_i) \quad (1)$$

, де $g_i = \partial_{\hat{y}_i} l(y_i, \hat{y}_i)$ та $h_i = \partial_{\hat{y}_i}^2 l(y_i, \hat{y}_i)$ це статистика градієнтів першого та другого порядку, ω_j – вагові коефіцієнти листків дерева, – коефіцієнт регуляризації. – це додатковий член регуляризації для дерева рішень, допомагає згладити остаточні вивчені ваги, щоб уникнути надмірної підгонки.

Результати

Розроблена система дозволяє підбирати релевантні фільми новим користувачам, які не мають жодних записів в системі. Роботу запропонованого алгоритму було продемонстровано на тестовому наборі даних, для оцінки якого розглядалися дві метрики: середньоквадратична помилка (RMSE) та абсолютна відсоткова помилка (MAPE). В цих метриках нижчі значення означають нижчий рівень помилок моделі, а отже вона має кращу прогнозуючу значимість.

RMSE – це квадратний корінь із середнього квадрата помилок і визначається за наведеною нижче формулою:

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (r_i - \hat{r}_i)^2}{N}} \quad (2)$$

, де r_i – значення фактичного рейтингу, \hat{r}_i – прогнозовані значення моделі, N – загальна кількість об'єктів.

MAPE вимірює похибку у відсотках. Він визначається формулою нижче:

$$MAPE = \frac{1}{N} \sum_{i=1}^N \frac{|r_i - \hat{r}_i|}{r_i} \times 100 \quad (3)$$

, де r_i – значення фактичного рейтингу, \hat{r}_i – прогнозовані значення моделі, N – загальна кількість об’єктів.

Модель “XGBoost” показала 0,67 RMSE і 19,86 MAPE на невидимих тестових даних. Це означає, що “XGBoost” є досить гарною моделлю. Значення RMSE менше 2 вважається добрим, а MAPE менше 25 — відмінним. Тим не менш, цю модель можна вдосконалити, додавши функції, які будуть рекомендовані на основі найкращих виборів залежно від місця чи жанру.

Висновки

Був запропонований метод по вирішенню проблеми з холодним стартом нових користувачів у рекомендаційних системах, який базується на матричній факторизації для вдосконалення системи. Також було проведено конструювання нових ознак набору даних для отримання гарних результатів. Обрали алгоритм машинного навчання для прогнозування рейтингу фільмів. В результаті була створена система рекомендацій, яка враховує схожість між користувачами, подібність фільмів і глобальні середні значення та матричну факторизацію. Ці концепції можна застосувати до будь-яких інших систем взаємодії між користувачем і елементом. Метрики оцінки показали гарні результати: RMSE=0.67 та MAPE=19.86.

Література

1. Ricci, Francesco; Rokach, Lior; Shapira, Bracha (2022). "Recommender Systems: Techniques, Applications, and Challenges". In Ricci, Francesco; Rokach, Lior; Shapira, Bracha (eds.). *Recommender Systems Handbook* (3 ed.). New York: Springer. pp. 1–35.
2. MovieLensdataset [Електронний ресурс] – Режим доступу: <https://grouplens.org/datasets/movielens/>
3. Koren, Yehuda; Bell, Robert; Volinsky, Chris (August 2009). "Matrix Factorization Techniques for Recommender Systems". *Computer*. 42 (8): 30–37.
4. Tianqi Chen, Carlos Guestrin “XGBoost: A Scalable Tree Boosting System” KDD ’16, August 13-17, 2016, San Francisco, CA, USA.

УДК 519.688

к. ф.-м. н., доцент Третиник В.В, магістр Гуро Д.А., доцент
Андрусенко О.М.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

ЗАСТОСУВАННЯ МЕТОДІВ МАШИННОГО НАВЧАННЯ В ПИТАННЯХ ОЦІНКИ ЯКОСТІ ЖИТТЯ МОЛОДІ

Abstract

Violeta Tretynyk, assoc. prof., PhD; Dmytro Huro, student

Applying machine learning methods to problems of quality of life of adults

This paper suggests an alternative method of dealing with quality-of-life researches and analyses. Against traditional manual and statistical methods of finding unsatisfied groups of people it is proposed to rely on unsupervised clusterization algorithms such as K-means, hierarchical, DBSCAN(Density Based Spatial Clustering of Application with Noise) or BIRCH(Balanced Iterative Reducing and Clustering using Hierarchies).

Вступ

Оскільки соціальний розвиток є прямим наслідком економічного зростання країни є актуальним питання аналізу якості життя населення, що обумовило в останні десятиліття підвищений інтерес дослідників до цього питання [5,6]. Такі чинники як, наприклад, матеріальне положення сім'ї, погане географічне розташування країни, забрудненість навколишнього середовища, не найкраще керування державою та інші впливають на якість життя людини в певному місці проживання.

На сьогоднішній день, не зважаючи на фундаментальність поставлених питань вони залишаються недостатньо вивченими. До цих пір дискусійними залишаються проблеми побудови і оцінки показників якості життя населення. Тому важливо класифікувати чинники, які впливають на якість життя людей, запропонувати сукупність показників для комплексної оцінки якості життя.

Постановка задачі

Мета роботи полягає в спробі провести аналіз рівня якості життя молодіза допомогою методів машинного навчання. Згідно з Законом України «Про сприяння соціальному становленню та розвитку молоді в Україні»[7], до молоді належать громадяни віком від 14 до 35 років, проте

в даному дослідженні громадяни віком до 20 років не будуть брати участь, за домовленістю авторів роботи. В ході дослідження необхідно виділити важливі показники, що можуть бути використані в дослідженні, зібрати необхідні дані: взяти з відкритих джерел та/або провести опитування за необхідності. На основі цих даних необхідно провести аналіз задоволеності громадян, виділити пені групи громадян за спільними проблемами та запропонувати рішення або рекомендації для покращення тих чи інших знайдених ситуацій, опираючись на дані. Предметом дослідження є виділення можливих чинників покращення якості життя молоді з використанням даних з проведених опитувань за допомогою методів машинного навчання. Метою роботи є загально-теоретичний аналіз основних підходів та методів для дослідження якості життя молоді, виявлення певних залежностей, кластерів.

Показники

Багатомірне поняття якості життя включає наступні складові: здоров'я, стан навколишнього середовища, психологічний добробут, досягнення цивілізації, стійкий розвиток суспільства, інфраструктура, свобода та ін. Окрім чинників, які впливають на якість життя населення розглядають також низку індексів, що виражають життя та розвиток суспільства. Наприклад, індекс людського розвитку, індекс освіти, індекс тривалості життя та ін.

Опираючись на наказ Міністерства праці та соціальної політики України про Державний класифікатор соціальних стандартів і нормативів[1], в якому приведено список сфер інтересу для створення вищезазначеного класифікатора, в даній роботі будуть використані показники, охоплені цим списком:

1. доходів населення;
2. соціального обслуговування;
3. житлово-комунального обслуговування;
4. транспортного обслуговування та зв'язку;
5. охорони здоров'я;
6. забезпечення навчальними закладами;
7. обслуговування закладами культури;
8. обслуговування закладами фізичної культури та спорту;
9. побутового обслуговування, торгівлі та громадського харчування;
10. соціальної роботи з дітьми, молоддю та різними категоріями сімей.

На основі обраних показників буде створено опитування, відповідями на питання котрого буде або число від 1 до 10, або якась опція

із запропонованих, або вказати власну відповідь. В залежності від контексту питання, може змінюватись точне трактування чисельної відповіді, проте ідея полягає у відображенні задоволеності опитуваного або опитуваної відносно певного питання.

Враховуючи можливість зміни списку запитань та їхню кількість, буде приведено лише декілька прикладів запитань: «Серед запропонованих оберіть діапазон Вашого загального місячного доходу», «Вкажіть назву та тип населеного пункту, де ви знаходитесь більшу частину року» або «Наскільки Ви задоволені рівнем медичних послуг?».

Обрані методи та технології

Спочатку, для загального огляду та розуміння зібраних даних необхідно провести статистичний аналіз. До нього входять кількісні таблиці загальних показників та їх порівняння, таблиця розсіянь тощо.

Для виділення певних категорій людей необхідно застосувати, так звані, алгоритми кластеризації, а саме кластеризації без вчителя, враховуючи, що у вихідних даних нема жодної вихідної кластеризації, тому алгоритму необхідно буде самостійно визначити за якими критеріями виділяти кластери.

В даній роботі будуть розглянуті деякі з них, а саме: алгоритм k-середніх[3], ієрархічна кластеризація [3], кластеризація на основі щільності DBSCAN[8] та метод BIRCH[4].

Алгоритм k-середніх — це ітераційний алгоритм, який намагається розділити набір даних на k-попередньо визначені окремі підгрупи (кластери), що не перекриваються, де кожна точка даних належить лише одній групі. Він намагається зробити точки даних всередині кластера якомога подібними, водночас зберігаючи кластери якомога різними (далекими). Він призначає точки даних кластеру таким чином, щоб сума квадратів відстані між точками даних і центроїдом кластера (середнє арифметичне всіх точок даних, які належать цьому кластеру) була мінімальною. Чим менше варіацій у нас всередині кластерів, тим однорідніші (подібніші) точки даних в одному кластері.

Ієрархічна кластеризація — це загальне сімейство алгоритмів кластеризації, які створюють вкладені кластери шляхом їх послідовного злиття або розбиття. Ця ієрархія кластерів представлена у вигляді дерева (або дендрограми). Корінь дерева — це унікальний кластер, який збирає всі зразки, а листя — це скупчення лише з одним зразком.

Просторова кластеризація додатків із шумом на основі щільності (DBSCAN) — це непараметричний алгоритм кластеризації на основі щільності: заданий набір точок у деякому просторі, він групує точки, які щільно упаковані разом (точки з багатьма сусідніми сусідами), позначаючи

як викиди точки, які лежать окремо в регіонах з низькою щільністю (чий найближчі сусіди занадто далекі). DBSCAN є одним із найпоширеніших алгоритмів кластеризації, а також одним з найбільш цитованих у науковій літературі.

Birch будує дерево під назвою Clustering Feature Tree (CFT) для переданих йому даних. Дані, по суті, зіштовхуються з втратами до набору вузлів Clustering Feature (CF Nodes). Вузли CF мають ряд підкластерів, які називаються субкластерами функцій кластеризації (CF Subclusters), і ці субкластери CF, розташовані в нетермінальних вузлах CF, можуть мати CF-вузли як дочірні.

Підкластери CF містять необхідну інформацію для кластеризації, що запобігає потребі зберігати всі вхідні дані в пам'яті. Ця інформація включає:

- Кількість вибірок у підкластері.
- Лінійна сума – n-вимірний вектор, що містить суму всіх вибірок
- Squared Sum - сума квадратів L2 норми всіх вибірок.
- Центроїди - щоб уникнути перерахунку лінійної суми / $n_samples$.
- Норма в квадраті центроїдів.

Висновки

У даній роботі виділено основні показники, які впливають на якість життя населення. Розглянуто методи та технології, які можуть бути використані для розв'язання поставленої задачі.

Проведення запропонованих оцінок дозволить керівництву підприємств, закладів, країн тощо знаходити причини незадоволеності або поганого самопочуття серед людей та покаже напрями підвищення задоволення людей від життя.

Список літератури

1. <https://zakon.rada.gov.ua/rada/show/v0293203-02#Text>
2. Ярчук А. В. Сутність поняття якості життя населення [Електронний ресурс] / А.В. Ярчук. - Режим доступу :<http://jeou.donnu.edu.ua/article/view/2917/2956>
3. <https://medium.com/analytics-vidhya/comparative-study-of-the-clustering-algorithms-54d1ed9ea732>
4. <https://www.geeksforgeeks.org/ml-birch-clustering/>
5. Богданова Т. І. Якість життя населення – як мета політики соціально-орієнтованої держави. //Вісник соціально-

економічних досліджень. Вип.37./ Одес.держ.екон.ун-т.- Одеса:
ОДЕУ,2009.-343 с. -С.8-15

6. Дзюба І. В. Категорія рівня життя населення: сучасний підхід /
І. В. Дзюба // Статистика України. – 2004. - № 1. – С. 68 – 71.
7. <https://zakon.rada.gov.ua/laws/show/2998-12#Text>
8. <https://uk.wikipedia.org/wiki/DBSCAN>

УДК 519.688

к.ф.-м.н., доцент Третиник В.В, магістр Давиденко М.А.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

ПОПЕРЕДНЕ ПОКРАЩЕННЯ ЯКОСТІ ЗОБРАЖЕННЯ ДЛЯ ПІДВИЩЕННЯ ТОЧНОСТІ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ

Abstract

Violeta Tretynyk, assoc. prof., PhD; Mykola Davidenko, student

Preliminary improvement of image quality to improve the accuracy recognition of objects
This paper concerns the problem of preliminary improvement of the quality of photographs before recognizing objects on them. Described modern algorithms with open source code. These are algorithms that can be used for most recognition models. In conclusion, chosen the optimal algorithm for subsequent application.

Вступ

Розпізнавання об'єктів з фотографій у наш час є дуже актуальним. Це може бути корисно для комерційного, державного так і військового проєкту. Щоб досягнути результату така система повинна мати дуже гарну математичну базу (алгоритми нейронних мереж), але щоб результати були корисними таку систему потрібно оснастити додатковою обробкою якості вхідних фотографій. Розумне підвищення якості зображень дає гарні результати і може бути в подальшому використане в різних системах комп'ютерного зору. Методи глибокого навчання як передовий засіб моделювання у машинному навчанні можна ефективно застосовувати для обробки зображень. Тому далі в роботі буде розглянуто деякі актуальні підходи до вирішення такої задачі та порівняно їх особливі характеристики роботи.

Постановка задачі

Мета роботи проаналізувати алгоритми глибокого навчання, що застосовуються для попередньої обробки зображення. Потрібно вибрати кращий метод обробки та підвищення якості зображення перед подальшим застосуванням його у комбінації з нейронними мережами. Також необхідно розібратися з особливостями роботи з моделями та їх реалізації. Врешті потрібно провести порівняльний аналіз між обраними методами та обрати слушний. Теоретичні інструменти, що використовуються у роботі

наступні: алгоритми, метрики оцінювання та розпізнавання об'єктів, критерії оптимізації, вибір нейромережевої моделі.

Опис існуючих методів підвищення якості фотографії

На сьогодні super-resolution відносять до процесу масштабування або підвищення деталей зображення. Зі збільшенням розміру зображення надлишки пікселів повинні якось інтерполювати. Основні методи обробки зображень не дають гарних результатів, оскільки вони не враховують контекст масштабування. Глибоке навчання зокрема генеративні змагальні мережі, приходять на допомогу і дають набагато кращі результати.

Найпопулярніша бібліотека комп'ютерного зору з відкритим кодом, яка містить велику колекцію гарних алгоритмів це OPENCV. Вона містить простий інтерфейс для використання та реалізації Super Resolution (SR) на основі методів глибокого навчання. Інтерфейс містить попередньо натреновані моделі, які є легкими у використанні. Наразі є багато реалізацій на різних мовах програмування зокрема і на Python.

1) Перший алгоритм глибокого навчання для масштабування зображень, який було обрано – це EDSR (Enhanced Deep Residual Networks for Single Image Super-Resolution). Він є схожим до MDSR, але для різних масштабів у EDSR потрібні різні моделі. Для порівняння, одна модель може реконструювати різні масштаби моделі MDSR, тому було розглянуто тільки EDSR.

EDSR - це модель машинного навчання, яку можна використовувати для збільшення роздільної здатності зображення [1].

Архітектура у виді ResNet використовується без шарів пакетної нормалізації. Розробники виявили, що ці рівні позбавляють гнучкості діапазону функцій у мережах, підвищуючи продуктивність. Це дозволяє їм побудувати більшу модель із кращою продуктивністю, щоб протистояти нестабільності, що виявляється у великих моделях. Розробники використовували залишкове масштабування з коефіцієнтом 0,1 у кожному залишковому блоці, поміщаючи шари постійного масштабування після останніх шарів згортання. Крім того, шари активації ReLu не використовуються після залишкових блоків.

Спочатку архітектура використовується для коефіцієнта масштабування 2. Потім ці попередньо навчені ваги використовуються при навчанні коефіцієнта масштабування 3 і 4. Це не тільки прискорює навчання, але і підвищує продуктивність моделей.

2). ESPCN (Efficient sub-pixel convolutional neural network) - це ефективна субпіксельна згорткова нейронна мережа. Замість того, щоб створювати надвисоку роздільну здатність після масштабування низької за допомогою

бікубічного фільтра, розробники вирішили отримувати “feature maps” об'єктів у малій кількості та використовувати складні фільтри роздільної здатності масштабування, щоб отримати результат[2].

Масштабування шарів розвертаються тільки в кінці мережі, що гарантує, що складні операції, що знаходяться в моделях, виконуються в менших вимірах, що робить їх швидшими, особливо в порівнянні з іншими методами[2].

Оригінальна структура ESPCN була натхненна SRCNN. Замість звичайних шарів згорткової мережі використовуються субпіксельні, що діють як шари деконволюції. Субпіксельні шари все ж використовуються в останньому шарі для створення “feature maps” високої роздільної здатності. Разом з цим до них була обрана функція активації Tanh, яка працює набагато краще, ніж стандартна функція ReLu.

3).LapSRN (Laplacian Pyramid Super-Resolution Network)

Таке рішення пропонує золоту середину між контрастними стратегіями масштабування на початку та в кінці. Його назва походить від пірамід Лапласа, але й архітектура в основному схожа на піраміду, яка масштабує зображення з нижчою роздільною здатністю до кінця. Для швидкості є дуже важливим спільне використання параметрів, так само як і в моделі EDSR, можна також запропонувати єдину модель, яка може реконструювати різні масштаби.

Модель складається з двох гілок: вилучення ознак і гілки реконструкції зображення. Спільне використання параметрів відбувається між різними масштабами, тобто 4x використовує параметри з моделі 2x і так далі. Це означає, що одна піраміда використовується для масштабування 2x, дві для 4x і три для 8x.

Порівняння результатів

Щоб отримати більш чітке уявлення про результати було порівняно роботу всіх трьох моделей на одній фотографії. Щоб якісно показати результати, із зображення була обрізана лише область метелика. Цю область збільшили в чотири рази за допомогою моделей із надвисокою роздільною здатністю.



Рис.1. Результати

Важко відрізнити результати неозброєним оком, просто збільшивши зображення. Таким чином, щоб перевірити всю продуктивність моделі, ці методи були застосовані до трьох зображень розміром 500×333 і зменшені до необхідного розміру перед підвищенням дискретизації до 500×333 . Збільшені зображення потім порівнювали з вихідним зображенням за допомогою PSNR(Пікове ставлення сигналу до шуму) і SSIM(Індекс структурної подібності). Середні результати всіх зображень були розраховані та наведені нижче.

Таблиця 1

Середні значення PSNR

Назва моделі	2x	3x	4x	8x
EDSR	28.5503	26.484	25.3513	-
ESPCN	28.3803	25.9613	25.0947	-
LapSRN	28.098	-	25.053	21.587

Таблиця 2

Середні значення SSIM

Назва моделі	2x	3x	4x	8x
EDSR	0,885	0,825	0,762	-
ESPCN	0,877	0,799	0,736	-
LapSRN	0,874	-	0,735	0,554

Треба зазначити, що розмір зображення, переданий у 3x, менший, ніж 2x, і є таким самим із ще більшими коефіцієнтами масштабування.

Таблиця 3 - Середній час роботи алгоритму у секундах

Назва моделі	2x	3x	4x	8x
EDSR	32,501	16,718	10,224	-
ESPCN	0,049	0,032	0,018	-
LapSRN	0,501	-	0,742	0,765

Висновок

В даній роботі було розглянуто 3 метода підвищення якості фотографій за допомогою сучасних алгоритмів, проаналізовано способи вирішення цієї задачі. Також коротко описано їх архітектура та результати. Підсумовуючи можна стверджувати, що метод EDSR дає найкращі результати, але час його роботи є більшим порівняно з іншими. Модель LapSRN краще працює в масштабуванні 8x, проте метод ESPCN є оптимальнішим, маючи задовільну якість і швидкий час роботи.

Засоби поліпшення якості зображень на основі штучного інтелекту, що застосовують методи глибокого навчання можуть допомогти у розробці інтелектуальних додатків, виявленні та розпізнаванні об'єктів на зображеннях і відеозаписах, впровадженні елементів комп'ютерного зору.

Література

1. arxiv.org[електронний ресурс]: інтернет сервіс з статтями. Режим доступу:<https://arxiv.org/abs/1707.02921>
2. arxiv.org[електронний ресурс]: інтернет сервіс з статтями. Режим доступу:<https://arxiv.org/abs/1609.05158>
3. G. Freedman and R. Fattal. Image and video upscaling from local self-examples. ACM Transactions on Graphics (TOG),30(2):12, 2011. 2

УДК 519.688

к. ф.-м. н., доцент Третиник В.В, магістр Коваленко О.П.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

АВТОМАТИЗАЦІЯ ПІДБОРУ ГІПЕРПАРМЕТРІВ LSTM ДЛЯ ЗАДАЧІ РОЗПІЗНАВАННЯ ЕМОЦІЙ ЗА АУДІОЗАПИСАМИ

Abstract

Violeta Tretynyk, assoc. prof., PhD; Olexandra Kovalenko, student
*Automation of hyper parameters selection of a neural network for emotion recognition
from audio recordings*

This paper concerns the task of automation of hyper parameters selection of a neural network. Neural network models, in our case LSTM, has many hyperparameters. They have a strong influence on the performance and accuracy of the final results, although the correct selection of parameters for a neural network is not an easy task. The paper examines ways of solving this issue and elucidates their features. The work also includes a table where the time and accuracy values of the test are presented in an accessible and understandable way, and it is possible to compare the obtained results.

Вступ

Задача класифікації аудіозаписів є затребуваним питанням в сучасних реаліях і може бути використана широким напрямком галузей, як для цивільних, так і військових потреб.

Існує ряд підходів та можливостей роботи з аудіозаписами, проте серед сучасних методів актуальна нейромережева реалізація вирішення такої задачі. Нейроні мережі стали популярним інструментом для створення аналізу та класифікації аудіозаписів. Численні види штучних нейронних мереж успішно застосовуються для знаходження відповідей на різні запити. Одна з таких LongShort-TermMemorylayer (LSTM). Ця модель має декілька гіперпараметрів, які істотно впливають на її точність та результати, проте пошук і налаштування оптимальних гіперпараметрів у багатовимірному просторі не є тривіальним завданням, оскільки потребує високого рівня кваліфікації і розуміння архітектури нейронних мереж, а також затрати часу та ресурсів для визначення оптимальних значень характеристики роботи нейронної мережі. Тому у даній роботі будемо розглянуто деякі методи пошуку для автоматизованого налаштування гіперпараметрів у мереж LSTM.

Постановка задачі

Об'єктом дослідження є способи організації засобів нейромережевого розпізнавання емоцій за аудіозаписами.

Теоретичні інструменти, які будуть використовуватись це:

- 1) методи і алгоритми реалізації архітектур нейронних мереж, аналіз та класифікація даних за заданими параметрами;
- 2) методи оптимізації гіперпараметрів: пошук архітектури нейронної мережі, автоналаштування; обрання моделі.

Мета даного дослідження полягає у виборі найкращого методу серед порівняних та проаналізованих підходів оптимізації та автоматизації роботи обраної нейронної мережі.

Спочатку необхідно обрати чотири гіперпараметри: швидкість навчання, кількість прихованих одиниць, довжину введення та кількість епох у нашому експерименті; далі провести порівняння між обраними методами оптимізації цих гіперпараметрів: випадковий пошук, баєсівська оптимізація, генетичний алгоритм. Нарешті потрібно провести остаточне налаштування гіперпараметрів у LSTM для підвищення якості класифікації емоцій людини.

Термінологія

Long Short-Term Memory layer— це різновид архітектури штучної рекурентної нейронної мережі, що використовується в галузі машинного навчання — глибинному навчанні. Мережа довгої короткочасної пам'яті є особливою формою рекурентної нейронної мережі, що в свою чергу є загальним терміном серії нейронних мереж, здатних обробляти послідовні дані. Для тренування моделі також використовується алгоритм зворотного розповсюдження помилки [1].

Регуляризація - це будь-яка модифікація алгоритму навчання з метою зменшити його помилку.

Гіперпараметрами у машинному навчанні називають параметри алгоритмів, значення яких встановлюються перед запуском процесу навчання. Гіперпараметри використовуються для управління процесом навчання. Вони надають можливість використовувати експериментальний підхід, щоб знайти найкращу конфігурацію для «структури» моделі.

Опис набору даних

CREMA-D — це набір даних із 7442 оригінальних нарізок від 91 актора. Ці нарізки були від 48 чоловіків і 43 жінок-акторів віком від 20 до 74 років, які належать до різних рас і етнічних груп (афроамериканців, азіатів, кавказьких, іспаномовних та невизначених). Актори виступали з

фрагментами тексту з 12 речень. Речення були подані з використанням однієї з шести різних емоцій (гнів, огида, страх, щастя, нейтральність, сум) і чотирьох різних рівнів емоцій (низький, середній, високий і невизначений).

Метод класифікації аудіозаписів

Нейронні мережі LSTM були розроблені спеціально для усунення проблеми довгострокової залежності. Дана проблема полягає в наступному: звичайна рекурентна нейронна мережа добре справляється з типом задач, де між актуальною інформацією та місцем, де вона потребується існує невелика відстань. В міру збільшення цієї відстані, рекурентна нейронна мережа втрачає здібність пов'язувати інформацію. Цю проблему і вирішує мережа LSTM, яка запам'ятовує інформацію на довгостроковий період. Як і в звичайній RNN, LSTM також має форму ланцюга модулів нейронної мережі, що повторюються. Проте на відміну від RNN, LSTM має більш складну структуру самого такого модуля (рис.1).

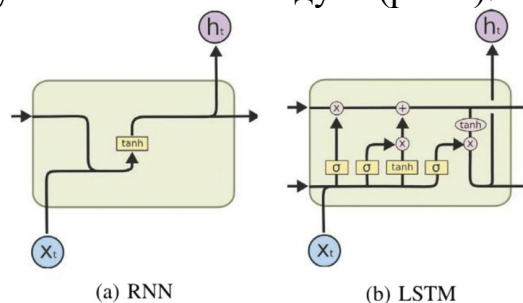


Рис.1 Порівняння архітектур RNN та LSTM

У порівнянні з рекурентним навчанням у реальному часі, LSTM призводить до більш успішних запусків і навчається набагато швидше. LSTM також вирішує складні, штучні завдання з великою затримкою, які ніколи не вирішувалися попередніми рекурентними мережевими алгоритмами [1].

LSTM має аширокий діапазон гіперпараметрів, таких як швидкість навчання, кількість прихованих одиниць, довжини вхідних даних, розмірів партій тощо. Неправильний вибір і відсутність оптимізації гіперпараметрів в комбінації з найсучаснішою нейронною мережею показують посередні результати.

Методи оптимізації підбору параметрів

Регуляризацію, яку потрібно застосовувати під час навчання, можна контролювати за допомогою гіперпараметрів [3]. Їх оптимізація повністю автоматизована та позбавляє користувача від ручного налаштування. Це

вдосконалення демонструє, що це є важливим завданням для аналізу даних у галузі машинного навчання.

Нижче буде представлено аналіз існуючих методів для вибору оптимального до обраної нейронної мережі:

1) Випадковий пошук

Це метод, при якому випадкові комбінації гіперпараметрів вибираються та використовуються для навчання моделі. Набір гіперпараметрів із зазначеного діапазону генерується непередбачуваним чином для кожної ітерації.

В цьому випадку було самостійно прописаний код для випадкової підстановки параметрів. Кількість ітерацій становила 5.

2) Баєсівська оптимізація

Такий метод починається шляхом вибірки випадкових значень. Алгоритм спостерігає за результатами, які генеруються. На основі цих спостережень оптимізатор використовує середнє цього Гаусового процесу в якості наближення функції. Щоб зрозуміти, які гіперпараметри потрібно відібрати далі, він використовує функцію збору. Баєсівська оптимізація характеризується формулою (1):

$$x^* = \operatorname{argmin}_x f(x) \quad (1)$$

де f – чорний ящик.

Баєсівську оптимізацію можна використовувати кожен раз, коли потрібно оптимізувати функцію чорного ящика. Це означає, що можна використовувати цей метод для пошуку глобальних оптимумів щодо будь-якої функції, для якої можна спостерігати лише вхідні та вихідні дані.

Для баєсівської оптимізації було використано бібліотеку з відкритим кодом Nuregas. Nuregas використовує nuregopt для виконання баєсівської оптимізації. Кількість ітерацій становила 5, щоб узгодити його з випадковим пошуком і подивитись, чи має він якусь перевагу порівняно з ним.

3) Генетичний алгоритм

Компоненти популяції, тобто елементи, називаються генами в цьому алгоритмі. Ці гени формують індивіда в популяції. Створюється пошуковий простір, в якому накопичуються всі особини. Усі особини закодовані в межах кінцевої довжини в просторі пошуку. Кожній особині в просторі пошуку (популяції) присвоюється оцінка придатності, яка говорить про її здатність конкурувати з іншими особинами. Усі особини з відповідними показниками фізичної підготовки шукаються та підтримуються за допомогою генетичного алгоритму, а особини з високими показниками фізичної підготовки отримують можливість відтворюватися. Нове потомство має кращі «часткові рішення» порівняно з

їхніми батьками. Цей процес повторюється до тих пір, поки нащадки не матимуть жодних нових атрибутів/особливостей порівняно з попередніми. В нашому випадку кожен індивід складається з чотирьох гіперпараметрів. Кожна особина представлена своїм геном, що кодується. Чисельність популяції в кожному поколінні становить 2 і є 5 поколінь.

Порівняння результатів

Згідно Таблиці 1 можна стверджувати, щорезультати отримані за допомогою генетичного алгоритму є якіснішими порівняно з іншими рішеннями.

Таблиця 1

Результати застосування методів оптимізації

	Генетичний алгоритм	Випадковий пошук	Баєсівська оптимізація
Час	бгод	2 год	2год 30 хв
Точність	0.556	0.532	0.502

Генетичний алгоритм має стохастичний розподіл, який відповідає ймовірнісним правилам переходу. Він також може працювати в шумному середовищі. Завдяки своїй ефективності генетичний алгоритм в комплексі з LSTM демонструє найкращі результати.

Висновки

В даній роботі було описано вибір методів для розробки системи роботи з аудіозаписами, обрані сучасні методи вирішення проблеми та проаналізовано алгоритми оптимізації для нейромережевої класифікації емоцій за звуковими доріжками та розглянуті їх переваги.

У статті показано, як вибір архітектури моделі та налаштування гіперпараметрів можуть значно підвищити продуктивність визначення належності емоції до певної групи. Виявлено, що найкращих результатів можна досягти, лише при умові правильного підбору параметрів. Оскільки їх вибірка доволі велика, то ручне налаштування виявиться занадто трудомістким та енергозатратним. Для запобігання таких проблем запропоновано автоматизувати процес. Серед використаних методів генетичний алгоритм дав найкращі результати з точністю 55%.

Надалі було б доцільним використання цього алгоритму в комбінації разом з нейронною мережею LSTM.

Література

1. [researchgate.net/\[електронний ресурс\]](https://researchgate.net/): інтернет сервіс для управління закупками. Режим доступу:

https://www.researchgate.net/publication/13853244_Long_Short-term_Memory

2. Гудфеллоу Я., Бенджио И., Курвилль А. Г93 Глубокое обучение / пер. с англ. А. А. Слинкина. – 2-е изд., испр. – М.: ДМК Пресс, 2018. – 652 с. – ISBN 978-5-97060-618-6

Violeta V. Tretynyk, assoc. prof., PhD; Naser J. Hamad, student

**National Technical University of
Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”**

SYSTEM OF CLUSTERIZATION OF ARABIC PAPERS

Abstract

Violeta V. Tretynyk, assoc. prof., PhD; Naser J. Hamad, student

System of clusterization of Arabic papers

The problems of clusterization of Arabic paper are solved. The techniques of Text mining are used for solving the problem. Step-by-step the systems of clusterization are builded. For clusterization we used the program WEKA.

Introduction

A big challenge today is the problem of the exponentially growing volume of information generated by humanity. Information on the Internet is growing especially rapidly. The data we receive from the Internet is usually raw, unstructured. Therefore, there is an urgent task of creating intelligent systems that allow extracting knowledge from unstructured information, to somehow consolidate the flow of data using methods of classification and clustering of the initial text, thereby facilitating the interpretation of this text. Text mining, which is part of a more general concept of intelligent data analysis or Data mining, deals with the automated search for extracting useful information from text documents.

Before analyzing text data, it is necessary to carry out preprocessing. It includes the following stages: tokenization, stemming or lemmatization, removal of stop words, frequency analysis of word forms based on vector model technologies.

One of the common forms of intelligent data analysis is the categorization of texts, that is, the assignment of a text to one or another class or category. Here, clustering methods can be used when there is no a priori information about the classes of documents under consideration. As a rule, in the process of clustering, a vector of topics is compared to each document, which determines the weighting measure of correspondence to each cluster.

Clustering methods [1] are divided into hierarchical and non-hierarchical. In the hierarchical method [4], sequential detailing of texts by context is implemented. In non-hierarchical methods, the set of all texts is divided into a given number of non-intersecting clusters. The formal formulation of the

clustering problem is as follows: we have a sample $S^l = \{s_1, \dots, s_l\}$ and a distance function between objects $\rho(s, s)$. It is necessary to divide the sample into subsets that will not overlap and that each cluster consists of objects that are close in terms of metrics ρ , while objects of different clusters are significantly different. Therefore, each object $s_i \in S^l$ is assigned a cluster number t_i . Thus, the clustering algorithm is a function $f: S \rightarrow T$ that matches each object with a cluster label $t \in T$.

Machine learning models that use text clustering need to translate text into a vector representation. The simplest representation is a Bag-of-words [3]. This representation is simple to implement, but has disadvantages: it does not take into account either the syntax or the context of the word. This model only calculates the frequency characteristics of the word in the document. More progressive is the Tf-idf (term frequency-inverted document frequency) method [3], which normalizes the frequency of the word in this document taking into account the frequency of its appearance in other documents. But the best vector implementation of words is realized by Word2Vec [6] and GloVe models in which the word is a vector in space in such a way that the vectors of similar words are next to each other. The cosine measure can be chosen as a measure of the distance between vectors.

Formulation of the problem

In this paper we carry out the clasterization of text that has written in Arabic. We have analyzed 31000 papers from Arabic newspapers (Saudi Newspapers Corpus). The papers divided by clusters according to the topic. The system of intelligent analysis of these papers is represented on Fig.1.

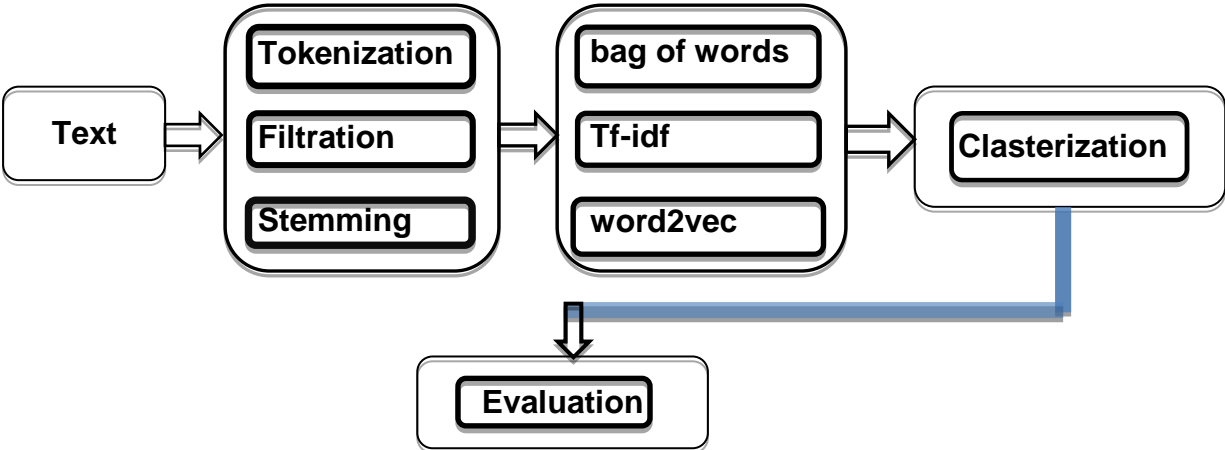


Fig. 1. System of clasterization

The program WEKA (Waikato Environment for Knowledge Analysis) was used for realization of the problem. It has graphical interface for model projection. There are components for preprocessing, clasterization, deep learning, and visualization. This tool is suitable for quickly model prototyping and then it is possible to integrate these models in Java, R, Python.

Description of the program WEKA

Here we introduce WEKA, which includes a set of visualization tools, data analysis algorithms, and predictive modeling.

We can say that it is open-source software that provides tools for data preprocessing, due to the implementation of various machine learning algorithms, we can develop machine learning techniques by providing tools, and use them for data mining problems. WEKA contains clusters for finding groups of a similar instance in a dataset.

This is generally WEKA graphic how WEKA works (Fig.2):

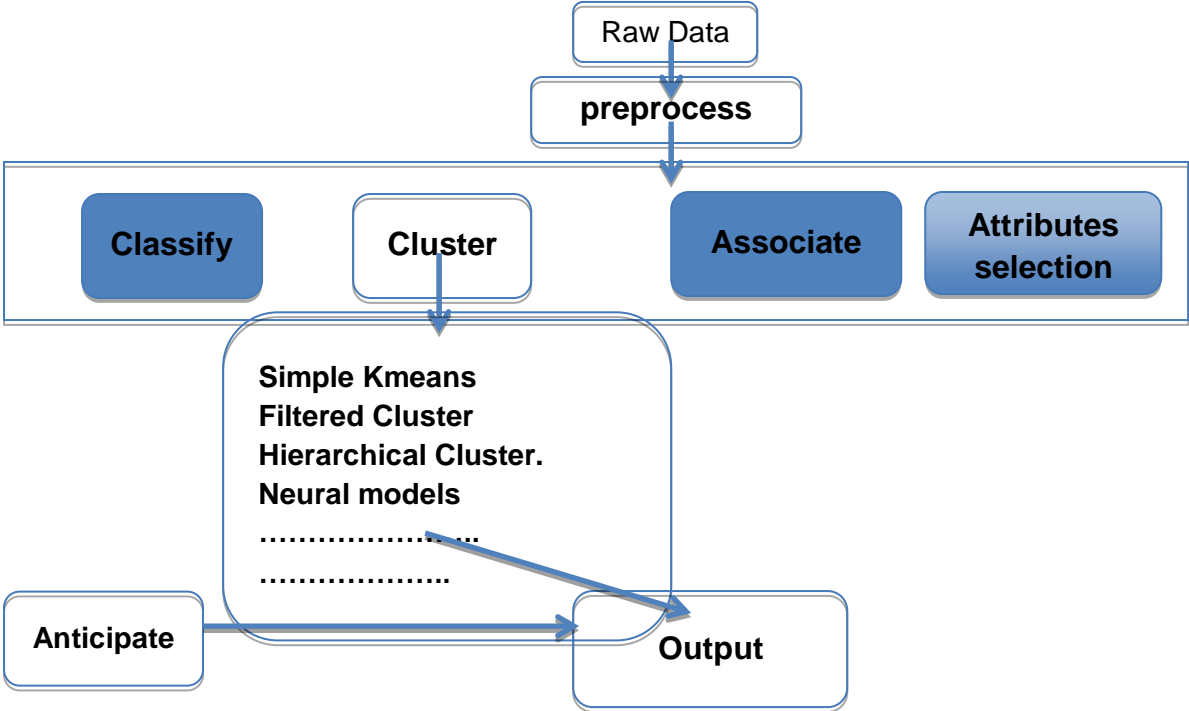


Fig. 2. Weka graphic system

WEKA is very helpful in preparing the data we have for classification, that is, everything that needs to be done before the data processing process, WEKA does it all for us, and all the unnecessary information is cleaned up in WEKA. This was a general definition of the WEKA program. Now we will discuss how WEKA can help us with clustering text.

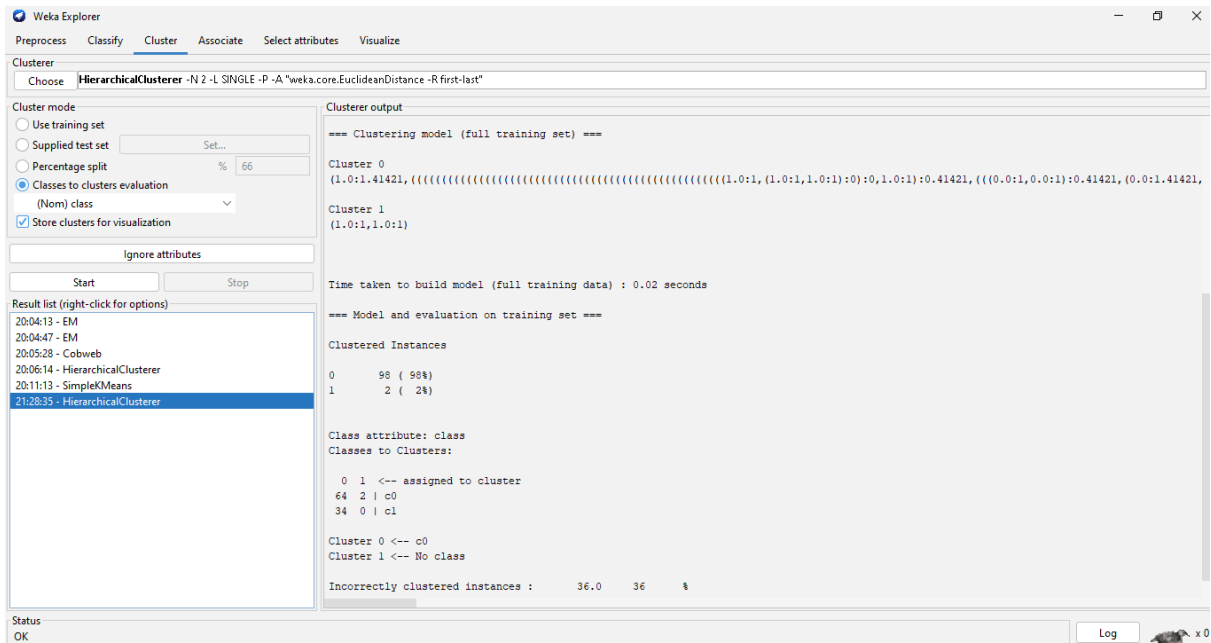


Fig. 4. The inputs result of Hierarchical Cluster.

Down here we have showed some visualize of WEKA program for the algorithm that we have used before, which shows each results in any case true and false (Fig.5):

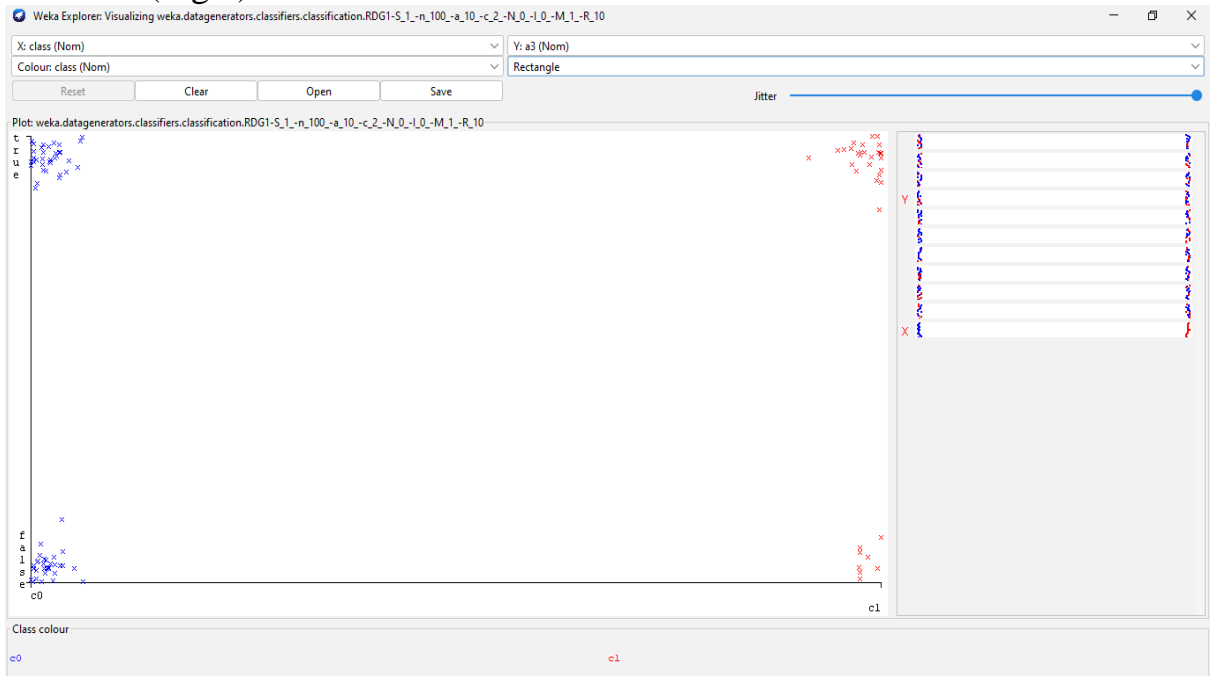


Fig. 5. Visualize the results,

Conclusions

Today's world is a digital world, and we have a lot of data, but we cannot get it properly because of the amount of data, the lack of a suitable program to classify them. We have classified a number of Arabic newspapers, classifying their information to make it easier for users to obtain the information they want, which has worked on getting the most results in the shortest time that are close to the input we have given. And the data is constantly growing; we need to try to classify the data so that we can use it. We have used some algorithms to do this like Neural models, Centroid models, Hierarchical, Statistical models and like as, and we try let's choosing the algorithm that gives the best results. And we have taken advantage of the Weka program, as explained in the section above.

References

1. *Agrawal, R., Gehrke, J., Gunopulos, D., & Raghavan, P.* (1998). Automatic subspace clustering of high dimensional data for data mining applications, Vol. 27, No. 2, pp. 94-105.
2. *P. Achimugu, A. Selamat, R. Ibrahim,* A clustering based technique for large scale prioritization during requirements elicitation, in *Recent Advances on Soft Computing and Data Mining*, vol. 287, ed. by T. Herawan, R. Ghazali, M. Deris (Springer, Cham, Switzerland, 2014), p. 3039.
3. *C.D.Manning, P. Raghavan, H. Schutze,* *Introduction to Information Retrieval*, Cambridge University Press. 2008.
4. *David Eppstein,* Fast Hierarchical Clustering via Dynamic Closest Pairs, Dept. Information and Computer Science, Univ. of California, Irvine, <http://www.ics.uci.edu/~eppstein/>
5. *Dempster, A.P.; Laird, N.M.; Rubin, D.B.* (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm". *Journal of the Royal Statistical Society. Series B (Methodological)*.
6. *T Mikolov, K Chen, G Corrado, J Dean,* Efficient estimation of word representations in vector space, (2013), arXiv preprint arXiv:1301.3781

Д.т.н., професор Чертов О. Р., студент Бірук С. В.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ КУПІВЛІ / ПРОДАЖУ ЕЛЕКТРОЕНЕРГІЇ ПОТУЖНОСТЕЙ ВІДНОВЛЮВАНОЇ ЕНЕРГЕТИКИ

Abstract

Oleg Chertov, Prof., Sc.D.; Stanislav V. Biruk, student

Mathematical and software system of buy/sale of electricity of renewable energy capacities

This article describes an approach to solving the problem of electricity arbitrage of renewable energy capacities. The component diagram of a hybrid energy system was considered. Historical data on meteorological indicators, cost and electricity consumption are presented. Planning and decision making on electricity purchase/sale is considered as a reinforcement learning problem. Ways for further research and improvements were proposed.

Вступ

Глобальне потепління та екологічні проблеми в цілому змусили світ шукати альтернативні джерела енергії для викопного палива. Протягом останніх років відновлювані джерела енергії впевнено впроваджуються в побут завдяки своїм техніко-економічним та екологічним особливостям.

Однак, величина початкових інвестицій у відповідне технічне забезпечення та середній термін окупності відлякують потенційних інвесторів, що робить майбутній перехід до гібридних систем електроспоживання довготривалим. Ця проблема ускладнюється ще й тим, що енергетичні потужності відновлюваної енергетики та засоби управління ними потребують високотехнологічних рішень і систем обробки великих обсягів даних. Тому впровадження інтелектуальних систем, що здатні обробляти великі об'єми даних та приймати оптимальні рішення в межах гібридних систем дозволяє зменшити період окупності початкових капіталовкладень.

У статті запропоновано рішення на основі алгоритмів навчання з підкріпленням, що здійснюють арбітраж електроенергії. Рішення щодо закупівлі та продажу електроенергії приймаються на основі історії попереднього споживання електроенергії споживачем, моделей

ціноутворення електроенергії та метеорологічних факторів за попередні періоди часу.

Постановка задачі

Об'єкт дослідження. Гібридна енергетична система та її складові, поняття арбітражу на ринку електроенергії, системи керування гібридними системами.

Предмет дослідження. Математичне та програмне забезпечення системи купівлі / продажу електроенергії, в основі якої використовується прогнозування часових залежностей та навчання з підкріпленням.

Метою дослідження є оптимізація процесу купівлі / продажу електроенергії з метою зменшення періоду повернення інвестицій, які були залучені при встановленні технічних засобів для генерації та збереження відновлюваної енергії.

Кінцевий результат. Моделі прогнозування та прийняття рішень для системи купівлі / продажу електроенергії. Математичне, програмне та методичне забезпечення системи.

Для розв'язання поставленої задачі використовувалися методи: теорії аналізу часових рядів, методів оптимізації, моделювання та проектування систем науки про дані (Data Science), машинного та глибокого навчання, візуалізації даних.

Об'єкт автоматизації

В межах даної роботи під об'єктом автоматизації розуміється гібридна енергетична система, що складається з таких компонентів (див. рис. 1):

- Solar PV Modules – підсистема генерації електроенергії;
- Electricity Storage – підсистема збереження відновлювальної та отриманої від електромережі електроенергії;
- Power Grid – мережа передачі електроенергії; ця підсистема додатково регулює вартість електроенергії;
- Consumer of Electricity – споживач електроенергії;
- Decision-Making Module – підсистема планування, яка вирішує (1) коли і в якому об'ємі купити електроенергію або її продати; та (2) від якого із джерел (Electricity Storage, Power Grid) електроенергія потрапляє до споживача.

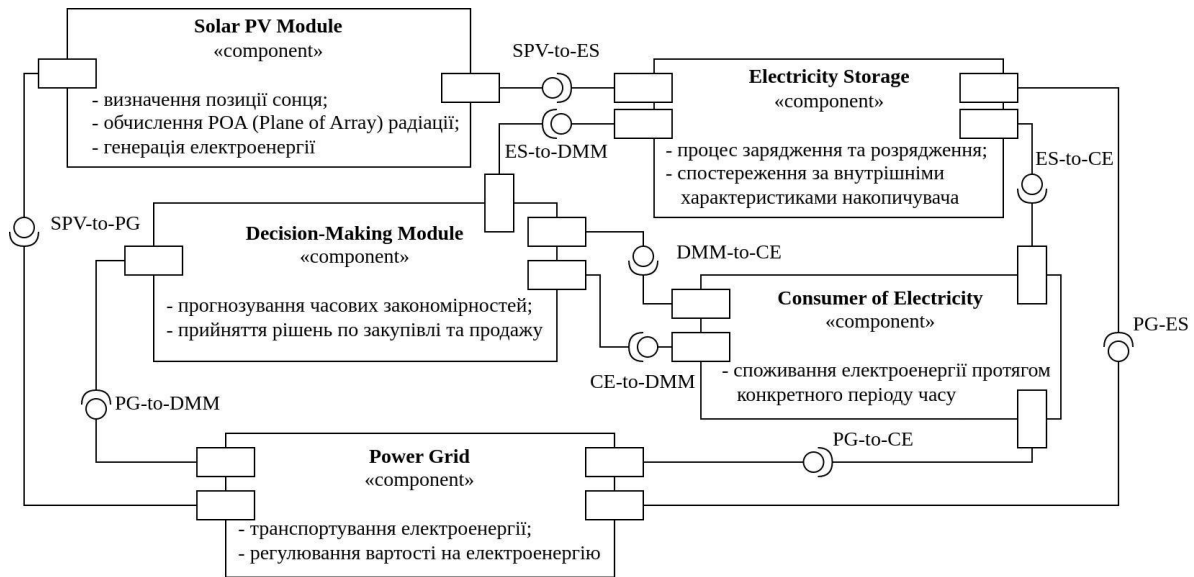


Рис. 1. Компонентна діаграма гібридної енергетичної системи.
Діаграма компонентів у нотації UML

Зв'язок між компонентами на рис. 1 забезпечується за допомогою інтерфейсів:

- SPV-to-PG – інтерфейс передачі інформації про кількість надлишкової згенерованої електроенергії;
- PG-to-DMM – інтерфейс передачі інформації про вартість електроенергії у попередні періоди часу;
- SPV-to-ES – інтерфейс передачі інформації про кількість згенерованої електроенергії, яку необхідно зберегти;
- ES-to-DMM – інтерфейс передачі інформації про стан накопичувача та кількість енергії, яку можна зберегти;
- DMM-to-CE – інтерфейс передачі інформації про шлях передачі електроенергії від джерела до споживача;
- CE-to-DMM – інтерфейс передачі інформації про споживання електроенергії;
- PG-to-CE та ES-to-CE – інтерфейс передачі інформації про кількість електроенергії, яка необхідна споживачу;
- PG-ES – інтерфейс передачі інформації про кількість купленої та проданої електроенергії.

Далі в роботі розглядається детально математичне забезпечення компоненту Decision-Making Model, як центрального алгоритмічного блоку системи прогнозування та прийняття рішень в межах гібридної системи.

Математичне забезпечення

Планування та прийняття оптимальних рішень по закупівлі та продажу електроенергії здійснюється за два етапи: спершу вирішується задача прогнозування часових рядів для вартості електроенергії, метеорологічних значень та кількості споживання електроенергії протягом певного періоду часу. Після чого, використовуючи спрогнозовані значення, здійснюється послідовний процес прийняття рішень: (1) коли і в якому об'ємі купити електроенергію або її продати; та (2) від якого із джерел (Electricity Storage, Power Grid) електроенергія потрапляє до споживача. Цей процес прийняття рішень формалізується як задача Markov Decision Process (MDP) [1].

Для прогнозування часових залежностей було обрано модель Temporal Fusion Transformer (TFT) [2], яка за рахунок механізму уваги дозволяє досягти меншої похибки прогнозування та інтерпретувати тимчасову динаміку двох процесів: споживання електроенергії та ціноутворення на неї. Для пошуку тимчасових зв'язків у різних масштабах TFT використовує рекурентні шари для локальної обробки та шари уваги для пошуку довгострокових залежностей. Зазначена модель прогнозування також використовує компоненти для відбору релевантних характеристик, що забезпечує високу продуктивність у широкому діапазоні роботи.

Навчання з підкріпленням (Reinforcement Learning, RL) [3] розглядають як послідовний цикл взаємодії агента з оточенням в якому він існує. На відповідний стан (навколишні умови) агент здійснює дію, яка винагороджується або штрафується. Формально, MDP формулюється як кортеж виду $\langle S, A, R, P, \rho_0 \rangle$, де

- S – множина всіх можливих станів. Спостереження s_t має вигляд кортежу виду $[X_m, X_p, X_c, X_b]$, де X_m – передбачені метеорологічні значення за допомогою TFT; X_p – передбачена вартість електроенергії; X_c – передбачені значення споживання електроенергії; X_b – внутрішні характеристики накопичувача електроенергії;
- A – множина всіх можливих дій;
- $R: S \times A \times S \rightarrow R$ – функція винагороди в грошовому еквіваленті;
- $P: S \times A \rightarrow P(S)$ – функція ймовірності переходу, де $P(s' | s, a)$ – ймовірність переходу в стан s' , якщо в стані s зроблено вибір a ;
- ρ_0 – початковий розподіл станів.

Для формалізації оптимізаційної задачі RL вводять нижче перелічені поняття.

Стратегія (policy) – набір правил, які використовує агент для вибору дії. Позначається наступним чином: $a_t \sim \pi(\cdot | s_t)$. Параметризована стратегія

$a_t \sim \pi_\theta(\cdot|s_t)$ представлена нейронною мережею з параметрами θ , які треба дізнатися [3].

Траєкторія – послідовність спостережень та дій $\tau = (s_0, a_0, s_1, \dots)$.

Функція винагород R залежить від стану s_t в момент часу t , відповідної дії a_t та спостереження s_{t+1} :

$$r_t = R(s_t, a_t, s_{t+1}).$$

Винагорода в межах однієї траєкторії визначається як:

$$R(\tau) = \sum_{t=0}^{\infty} \gamma^t r_t,$$

де γ^t – коефіцієнт дисконту, що приймає значення на інтервалі $(0, 1)$.

Таким чином, ймовірність переходу протягом T -кроків визначається таким чином:

$$P(\tau|\pi) = \rho_o(s_0) \prod_{t=0}^{T-1} P(s_{t+1}|s_t, a_t) \pi(a_t|s_t).$$

Очікувана винагорода в цьому випадку обчислюється як математичне очікування протягом всієї траєкторії:

$$J(\pi) = \int_{\tau} P(\tau|\pi) R(\tau) = E_{\tau|\pi}[R(\tau)].$$

Оптимізаційна проблема RL [3] зводиться до пошуку стратегії, що максимізує очікувану винагороду:

$$\pi^* = \operatorname{argmax}_{\pi} J(\pi).$$

Опис набору даних

Внутрішні процеси деяких з компонентів гібридної енергетичної системи відповідають історичним спостереженням:

1. End-Use Load Profiles for the U.S. Building Stock [4]. Цей набір даних надає інформацію про попит на електроенергію в межах дня з часовим проміжком в 15 хв. для всіх основних типів житлових та комерційних будівель. Загальний його розмір складає 39.68 ТБ.

2. National Solar Radiation Database (NSRDB) [5]. Вказаний набір даних є колекцією годинних значень метеорологічних даних та трьох основних складових сонячної радіації: глобальна горизонтальна, пряма нормальна та дифузна горизонтальна радіація. Загальний його розмір складає 341.49 ТБ.

3. New York Independent System Operator (NYISO) надає інформацію про вартість електроенергії в межах агломерації Нью-Йорка.

В цій роботі розглядаються спостереження протягом 2018 - 2020 років в географічних межах роботи NYISO.

Висновки

1. В ході роботи було досліджено і встановлено інструменти, які на сьогодні дозволяють виконувати управління енергетичними потужностями відновлювальної енергетики.

2. Запропоновано систему, що автоматизує процес купівлі / продажу електроенергії потужностей відновлюваної енергетики на основі методів прогнозування часових закономірностей та навчання з підкріпленням.

3. Перспективи подальших досліджень спрямовані на розширення функціональності за рахунок розподілених та хмарних технологій.

Література

1. Puterman M. Markov decision processes / Martin. L Puterman // Handbooks in Operations Research and Management Science, 1990. – (Elsevier Science Publishers B.V.). – p. 331–366.
2. Temporal Fusion Transformers for interpretable multi-horizon time series forecasting / B. Lim, S. Arık, N. Loeff, T. Pfister // International Journal of Forecasting. – 2021. – 37 (4), p. 1748–1764.
3. Li Y. Deep Reinforcement Learning [Електронний ресурс] / Yuxi Li. – 2018. – Режим доступу до ресурсу: <https://arxiv.org/pdf/1810.06339.pdf>.
4. End-Use Load Profiles for the U.S. Building Stock [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://data.openei.org/submissions/4520>.
5. National Solar Radiation Database (NSRDB) [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <https://data.openei.org/submissions/1>.

УДК 519.688

Аспірант Щьоголев М.О.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

МЕТОДИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ТА ОБЧИСЛЮВАЛЬНОЇ ЛІНГВІСТИКИ ДЛЯ ВИЗНАЧЕННЯ ЗАГАЛЬНОЇ ОЦІНКИ РІВНЯ НАПРУЖЕНОСТІ В СУСПІЛЬСТВІ

Abstract

Maksym Shchoholiev, PhD student

Decision-making and Computational Linguistic Methods for Overall Estimation of the Level of Social Tension

The rapid spread of coronavirus disease and related restrictive measures have led to increased levels of tension in societies around the world. The purpose of this research is to develop the system for determining the level of tension in society on the basis of data from social networks using decision-making and computational linguistic tools. The impact of various factors on the growth of social tensions is determined with the help of knowledge bases built by means of decision support systems. To analyze the emotionality of comment on news publications the TF-IDF and Word2vec methods are used. Within the framework of the method of goal dynamic estimating of alternatives, the level of social tension is determined as the degree of achievement of the main goal, the ratings of news publications and the ratings of news events that contributed to the increase in social tension are calculated. The study presents the efficient method of determining the level of social tension and defining the main factors that have the greatest impact on the growth of social tension.

Вступ

Кризові явища є однією з головних причин швидкого зростання рівня напруженості в суспільстві. Пандемія COVID-19 стала важким випробуванням для систем охорони здоров'я більшості країн світу. Уряди країн були змушені ввести обмежувальні протиепідемічні заходи, щоб запобігти поширенню хвороби. Поява нових штамів коронавірусу та висока смертність зумовили необхідність масштабних кампаній з вакцинації та обмеження деяких прав невакцинованих людей. Численні обмеження, пов'язані з пандемією, призвели до значного скорочення доходів працівників різних сфер, зокрема сфери послуг та деяких галузей виробництва. Усе це призвело до зростання тривожності та невдоволення людей, яке в різних частинах світу вилилося у вуличні протести.

Основними завданнями розроблюваної системи є визначення рівня напруженості в суспільстві на основі даних із соціальних мереж та визначення динаміки зміни цієї напруженості. Дані збирались зі сторінок українських новинних ресурсів у соціальній мережі Facebook. Метою розробки цієї системи є отримання державними органами України оперативної інформації щодо рівня напруженості в суспільстві, а також інформації щодо причин її виникнення.

Розробка системи визначення рівня напруженості в суспільстві продовжується вже протягом кількох років. Розширений опис системи подано у статті [1].

Постановка задачі

Метою даної роботи є опис інструментів підтримки прийняття рішень та обчислювальної лінгвістики, а також системи визначення рівня напруженості в суспільстві, що розробляється з використанням цих інструментів. Окрім того, метою є демонстрація результатів роботи прототипу цієї системи.

Термінологія

Суспільна напруженість (СН) – емоційно-психологічний стан, викликаний тривалим впливом зовнішніх подразнюючих факторів, до яких належать як реальні проблеми (низька соціальна захищеність, безробіття, підвищення цін, пандемія, війна тощо), так і інформаційні впливи (інформаційні атаки, політична агітація, новини, що генерують почуття тривожності). В контексті системи, що розробляється, напруженість розглядається як емоційність, масштабована на певну суспільну групу.

СППР – система підтримки прийняття рішень.

БЗ – база знань.

ЧКВ – частковий коефіцієнт впливу.

Застосування СППР для визначення рівня напруженості в суспільстві

Запропонований підхід має певні спільні риси з підходами, що використовуються для розпізнавання інформаційних операцій [2]. У межах запропонованого підходу засобами СППР створюються бази знань, в яких шляхом послідовної декомпозиції розкриваються фактори СН, вказуються відповідні новинні події та новинні публікації. При цьому використовуються результати контент-моніторингу новинних сторінок у соціальних мережах та знання експертів. У результаті аналізу емоційності коментарів до новинних публікацій з використанням методів TF-IDF та Word2vec [3] визначається рівень емоційної напруженості, пов'язаної з

тими чи іншими новинними подіями і, як наслідок, їхній вклад у підвищення СН. Після цього за допомогою методу ієрархічного цільового динамічного оцінювання альтернатив [4] рівень СН визначається як ступінь досягнення головної цілі, розраховуються рейтинги новинних публікацій та рейтинги новинних подій, що роблять свій вклад у підвищення СН.

Зміст методики застосування інструментарію СППР для вимірювання СН полягає в такому:

1. Проводиться попереднє дослідження об'єкта, для якого буде вимірюватись СН, підбирається група експертів.
2. Проводиться експертна декомпозиція факторів (критеріїв) СН. У БЗ СППР вводяться критерії СН у вигляді відповідних цілей.
3. У БЗ СППР вводяться цілі, що відповідають новинним подіям. Ці цілі підв'язуються до відповідних факторів СН.
4. У БЗ СППР вводяться проекти, що відповідають новинним публікаціям, встановлюється їхній вплив на відповідні новинні події.
5. Проекти можуть об'єднуватись у комплексні проекти, що відповідають кожній новинній події.
6. ЧКВ проекту (новинної публікації) визначається як нормоване значення добутку кількості переглядів новинної публікації на рівень емоційної напруженості, який вона провокує.
7. ЧКВ цілі (новинної події) визначається як нормоване значення суми добутків кількостей переглядів пов'язаних з нею новинних публікацій на рівень емоційної напруженості, який вони провокують.
8. ЧКВ цілей (критеріїв СН) визначаються або за допомогою експертів [5], або шляхом підрахунку нормованих значень суми добутків кількостей переглядів новинних публікацій, що підв'язані до відповідних новинних подій, на рівень емоційної напруженості, який вони провокують.
9. Обчислюються степені досягнення головної цілі як показник рівня СН. Також за необхідності обчислюється рівень СН за конкретними факторами.
10. Обчислюються рейтинги ефективності проектів та комплексних проектів як показники внеску кожної з новинних публікацій у створення СН.

Застосування методів сентимент-аналізу для визначення рівня напруженості в суспільстві

Для підрахунку рейтингу новинних публікацій та новинних подій використовується оцінка СН, що визначається на основі аналізу коментарів

з соціальних мереж та на основі популярності сторінок новинних ресурсів, з яких отримані ці коментарі.

Схема підрахунку оцінки СН для певної події або новинного ресурсу подана на рисунку 1.

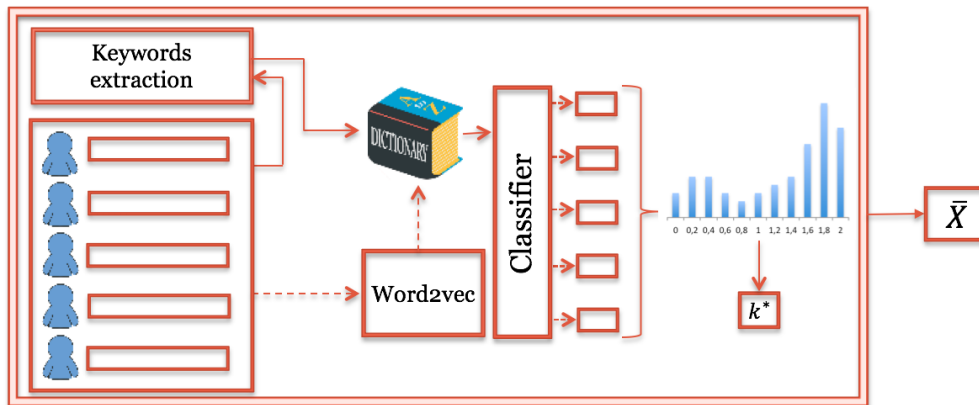


Рис. 1. Схема визначення рівня напруженості в суспільстві

Оцінки емоційності коментарів визначаються на основі аналізу їхніх ключових слів. Для виділення ключових слів, серед яких, відповідно до проведених експериментів, найчастіше зустрічається емоційна лексика, використовується метод TF-IDF. З виділених слів формується словник, в якому для кожного слова подано його контекстний вектор. Останній формується за допомогою методу Word2vec з навчальним алгоритмом CBOW (ContinuousBag-of-words). Група векторів ключових слів, що характеризує досліджуваний коментар, потрапляє на вхід класифікатора, який визначає оцінку емоційності цього коментаря. Як класифікатор використовується штучна нейронна мережа.

Оцінки коментарів визначаються на інтервалі $[0; 2]$, де 0 показує найвищу позитивну емоційність, 2 – найвищу негативну емоційність, 1 – нейтральність коментаря.

Також враховується рівень вкладеності коментаря (коментаря на коментар). Що більший рівень вкладеності, то менший вплив цього коментаря на загальну оцінку напруженості.

На основі отриманих оцінок для кожної публікації можна побудувати графік розподілу напруженості. Схематично цей графік можна подати у вигляді однієї з восьми гістограм, якщо об'єднати всі оцінки у три рівновеликі групи (рисунок2). Показником того, що публікація провокує підвищення рівня напруженості, є велика кількість коментарів з оцінками емоційності 1,4-2. Тому надалі увага зосереджуватиметься насамперед на тих публікаціях, емоційний рисунок яких має вигляд b, d, або h.

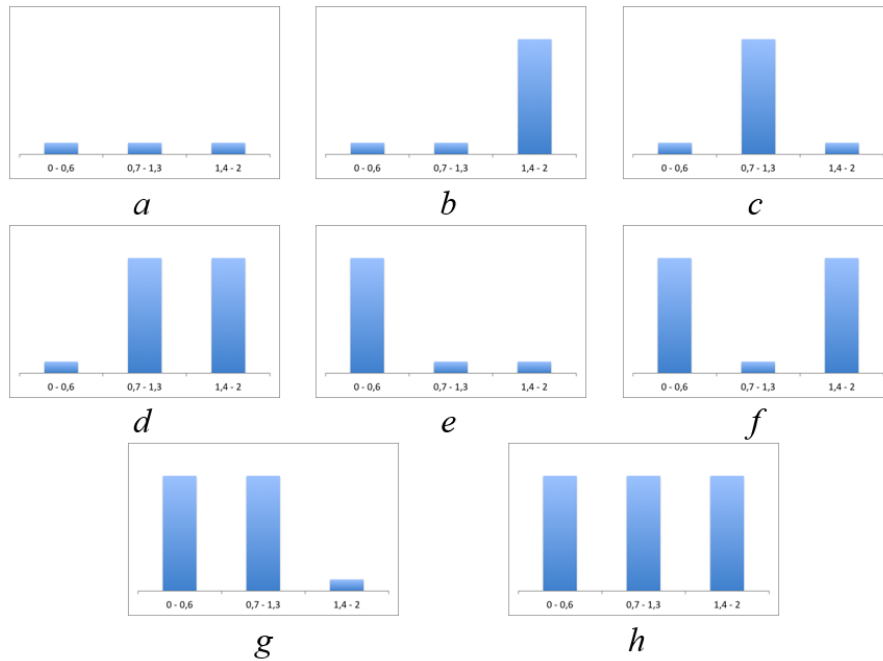


Рис. 2. Спрощені схеми розподілу коментарів за емоційністю

Середня кількість коментарів з оцінками 1.4-2 буде підраховуватись як зважене середнє арифметичне:

де $N_i \in \mathbf{N}$ - кількість коментарів з певною оцінкою емоційності, $n_1 = 7$ - кількість можливих оцінок емоційності на інтервалі 1.4-2, $w_i = \frac{i}{10} + 0,6; i \in [1; 7]$ - ваги оцінок (подані у таблиці 1).

Таблиця 1

Розподіл коефіцієнтів для коментарів оцінками емоційності 1.4-2

Оцінки емоційності коментарів	1.4	1.5	1.6	1.7	1.8	1.9	2.0
Вагові коефіцієнти	0.7	0.8	0.9	1.0	1.1	1.2	1.3

Загальна середня кількість коментарів буде підраховуватись як звичайне середнє арифметичне:

$$x_2 = \frac{\sum_{i=1}^{n_2} N_i}{n_2},$$

де $N_i \in \mathbf{N}$ - кількість коментарів з певною оцінкою емоційності, $n_2 = 21$ - загальна кількість можливих оцінок емоційності.

Коефіцієнт, що показує відношення між загальною середньою кількістю коментарів та середньою кількістю коментарів з негативним емоційним забарвленням, може бути підрахований за формулою:

$$k = \frac{x_2}{x_1}.$$

Основну увагу буде зосереджено на тих публікаціях, для яких $k \leq 1$, тобто де середня кількість коментарів з високою негативною емоційністю перевищує загальну середню кількість коментарів. При цьому підраховуються значення n_{neg} , що показує кількість досліджуваних публікацій, для яких $k \leq 1$, та n_{pos} , що показує кількість досліджуваних публікацій, для яких $k > 1$.

Щоб взяти до уваги популярність новинних ресурсів, підраховується значення k^* для кожного k за формулою:

$$k^* = 1 - a(1 - k),$$

де a - кількість підписників, нормована за всіма новинними ресурсами, $a \in (0; 1]$. Графік значень k^* для різних a та k на інтервалі $[0; 1]$ подано на рисунку 3.

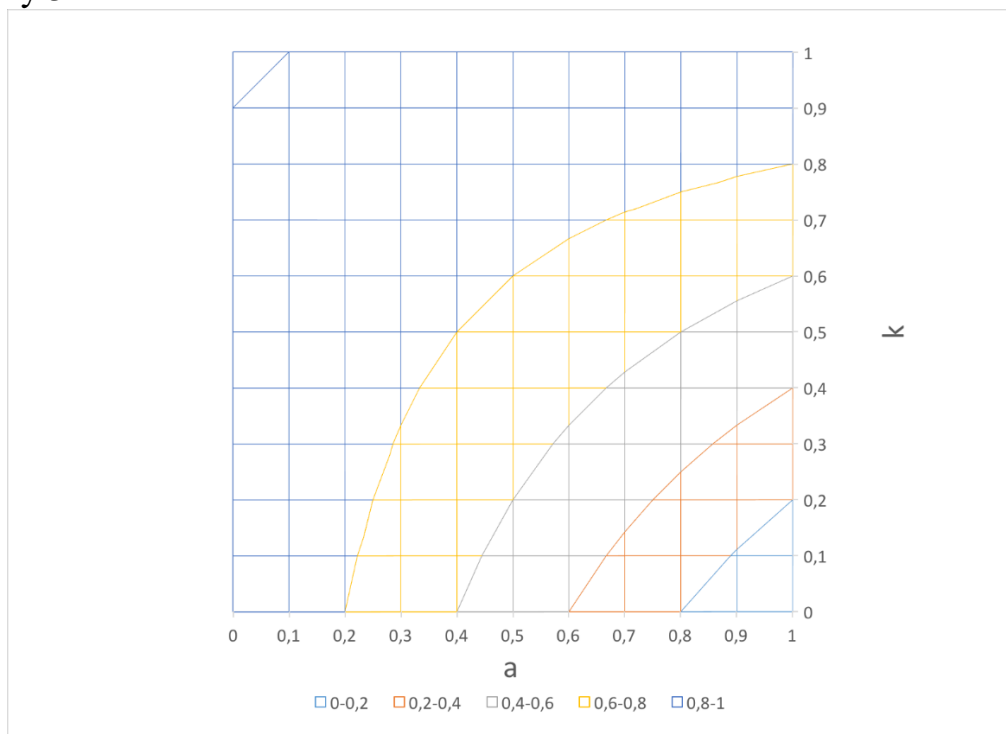


Рис. 3. Графік значень k^* для різних a та k на інтервалі $[0; 1]$

Для підрахунку середнього рівня напруженості може бути застосована формула середнього геометричного:

$$\bar{X} = \sqrt[n]{\prod_{j=1}^n k_j^*}.$$

$\bar{X} \in [0, 1]$. В залежності від поставленої задачі за цією формулою може бути підрахована середня негативна напруженість або для певного новинного ресурсу, або для певної новини.

Практичний приклад розрахунку соціальної напруженості засобами СППР

Розглянемо застосування описаної вище методики для вимірювання СН на прикладі українського суспільства у період 01.01.2020 – 15.05.2020.

На рисунку 4 подано приклад побудови засобами СППР БЗ СН у цей період. Таблиця 2 містить список усіх цілей (проектів) БЗ, що розглядається.

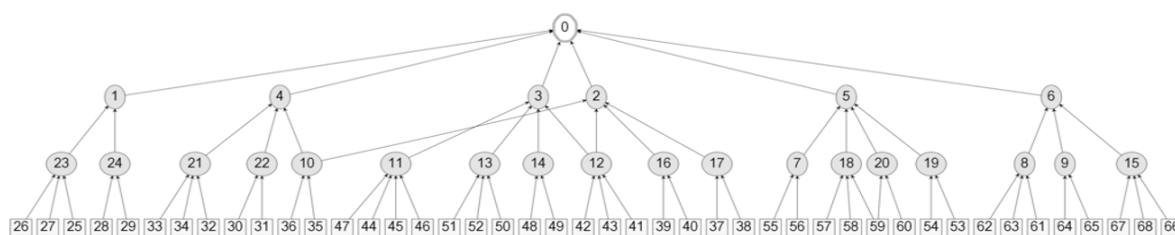


Рис. 4. Фрагмент побудованої БЗ СППР соціальної напруженості

В результаті підрахунку рівня соціальної напруженості засобами СППР як ступеня досягнення головної цілі за період 01.01.2020 – 15.05.2020 було отримане значення 0.42278.

Фрагмент (топ 5) результатів підрахунку за допомогою СППР рейтингу новинних публікацій як ефективності проектів подано у таблиці 3.

Таблиця 3

Фрагмент (топ 5) результатів підрахунку рейтингу новинних публікацій

Ефективність	Новинні публікації, пов'язані з подією
0.051279277	Введення карантину
0.034086009	Зупинка громадського транспорту через карантин
0.031551813	Інтерв'ю Гордона з Гіркіним
0.020846635	Стрільянина біля озера у Житомирській області
0.019569476	Нестача захисних масок в аптеках

Фрагмент (топ 5) результатів підрахунку за допомогою СППР рейтингу новинних подій як ефективності комплексних проектів подано у таблиці 4. Кожен комплексний проект складається із простих проектів, яким відповідають новинні публікації.

Таблиця 4

Фрагмент (топ 5) результатів підрахунку рейтингу новинних подій

Ефективність	Новинні події
0.143001	Введення карантину
0.10456	Зупинка громадського транспорту через карантин
0.096787	Інтерв'ю Гордона з Гіркіним
0.069838	Лісові пожежі у Чорнобильській зоні відчуження
0.057445	Нестача захисних масок в аптеках

Таблиця 2

Список формулювань цілей

#	Формулювання цілей
0	Соціальна напруженість в Україні
1	Криміногенна ситуація
2	Безробіття
3	Проблеми охорони здоров'я
4	Проблеми в області ЖКГ
5	Загроза воєнного конфлікту
6	Проблеми екології та навколишнього середовища
7	СБУ викрила генерал-майора Шайтанова, який працював на ФСБ Росії
8	Лісові пожежі у Чорнобильській зоні відчуження
9	Київ повернувся до рейтингу міст з найбільшим забрудненням повітря
10	Підвищення абонплати для мобільних мереж
11	Введення карантину
12	Зупинка громадського транспорту через карантин
13	Нестача захисних масок в аптеках
14	Реформа системи охорони здоров'я
15	Лісові пожежі у Житомирській області
16	Зростання цін на продукти
17	Підвищення курсу долара
18	Інтерв'ю Гордона з Гіркіним
19	Обмін полоненими. 20 заручників були звільнені з окупованих територій
20	Справа МН17
21	Обвал будинку в Одесі
22	Падіння мосту у Дніпропетровській області
23	Стрілянина біля озера у Житомирській області
24	Пограбування у Харкові. Невідомі підірвали банкомат і вкрали гроші
25-68	Новинні публікації

Висновки

Починаючи з 2020 року, пандемія COVID-19 стала одним з найважливіших факторів, що впливає на економічну та політичну ситуацію у більшості країн світу. І Україна в цьому сенсі не є виключенням. Це підтверджується результатами поданого тут дослідження.

У цій статті була описана система визначення рівня СН з використанням інструментів обчислювальної лінгвістики та БЗ СППР. Запропоновано методологію використання інструментарію СППР для побудови БЗ, у якій шляхом послідовної декомпозиції визначено основні фактори СН. Зокрема у поданій роботі було використано результати контент-моніторингу та експертні оцінки. Серед численних методів оцінки альтернатив було обрано метод ієрархічного цільового динамічного оцінювання альтернатив. Під час дослідження емоційності коментарів застосовувались методи сентимент-аналізу, зокрема методи TF-IDF та Word2vec. Розрахунок СН в Україні проводився в період 01.01.2020 – 23.05.2020, за результатами якого були виявлені новинні публікації та новинні події, які показали найбільшу ефективність у підвищенні рівня напруженості. Запровадження карантинувало основним фактором підвищення СН в Україні протягом досліджуваного періоду.

Література

1. M. Shchokoliev, O. Andriichuk, V. Tsyganok, V. Tretynyk (2021). Decision-making and computational linguistic tools application for overall estimation of the level of social tension. *Journal of Physics: Conference Series*, 1780, 9 p.
2. A. Dodonov, D. Lande, V. Tsyganok, O. Andriichuk, S. Kadenko, A. Graivoronskaya (2019). *Information Operations Recognition. From Nonlinear Analysis to Decision-Making*, Lambert Academic Publishing, 275 p.
3. T. Mikolov, Q. V. Le, I. Sutskever (2013). Exploiting Similarities among Languages for Machine Translation, *arXiv: Computation and Language*, 10 p. [<https://arxiv.org/pdf/1309.4168.pdf>]
4. V. G. Totsenko (2001). One Approach to the Decision Making Support in R&D Planning. Part 2. The Method of Goal Dynamic Estimating of Alternatives, *Journal of Automation and Information Sciences*, vol. 33, issue 4, pp. 82-90.
5. V. G. Totsenko, V. V. Tsyganok (1999). Method of paired comparisons using feedback with expert. *Journal of Automation and Information Sciences*, 13 (7-9), pp. 86-96.

К.т.н., доцент Петрашенко А.В., студент Бербега В.О.

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

ОПТИМІЗАЦІЯ АЛГОРИТМУ ТРАСУВАННЯ ПРОМЕНІВ ШЛЯХОМ ВИКОРИСТАННЯ СПЕЦІАЛІЗОВАНОГО АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ

Abstract

Andriy V. Petrashenko, assoc. prof., PhD; Volodymyr Berbeha, student
Optimization of the Ray Tracing algorithm with using specialized hardware

This paper concerns the task of ray tracing research and the method of the optimization of the ray tracing algorithm. The Basic Ray tracing algorithm is studied. The Local and Global illumination is researched as well. GPU implementation of the ray tracing algorithm is presented.

Вступ

Трасування променів є важливою темою, коли йдеться про комп'ютерну графіку. Цей алгоритм поєднує питання ефективності та реалістичності, таким чином знаходячи сприятливий баланс часу та зусиль, залучених для створення реалістичних тривимірних зображень.

У реальному житті часто існує кілька джерел світла та безліч відбиваючих об'єктів, які взаємодіють один з одним різними способами. «Глобальне освітлення», більш вдосконалена форма трасування променів, додає локальну модель, відбиваючи світло від навколишніх поверхонь до об'єкта. Модель глобального освітлення є більш повною, фізично коректнішою та створює більш реалістичні зображення.

Проте обчислення трасування променів в реальному часі є достатньо обчислювально містким процесом. Так в даній роботі досліджується оптимізація алгоритму трасування променів шляхом використання спеціалізованого апаратного забезпечення.

Постановка задачі

Обчислення алгоритму трасування променів є часозатратною операцією в силу специфіки архітектури центрального процесора. Тож питання оптимізації залишається актуальним. Проте оптимізація алгоритму шляхом використання спеціалізованого апаратного забезпечення досліджена не достатньо.

Задача полягає в дослідженні алгоритму трасування променів з використанням методу глобального освітлення. Оптимізації алгоритму трасування променів шляхом використання спеціалізованого апаратного забезпечення.

Термінологія

Трасування променів – у комп'ютерній графіці є способом створення зображення тривимірних об'єктів чи сцени за допомогою відстеження ходу променя світла крізь точку екрану і симуляції взаємодії цього променя з уявними об'єктами, що підлягають відображенню [5].

Глобальне освітлення — це загальна назва для групи алгоритмів, що використовуються в тривимірній графіці, які призначені для більш реалістичної імітації світла 3D сцени. Такі алгоритми враховують не лише пряме світло (англ. Direct illumination), що надходить безпосередньо від джерела світла, але також і відбиті промені світла від інших поверхонь об'єктів сцени [2].

Графічний процесор (GPU) – пристрій, апаратна складова персонального комп'ютера, що призначена для обробки графіки, зменшуючи навантаження на центральний процесор.

CUDA – програмно-апаратна архітектура паралельних обчислень, яка дозволяє істотно збільшити обчислювальну продуктивність завдяки використанню графічних процесорів компанії Nvidia [6].

Спеціалізоване апаратне забезпечення

В якості спеціалізованого апаратного забезпечення виступає графічний процесор. Графічний процесор (GPU) – це окремий пристрій, що є частиною відеоадаптера персонального комп'ютера і призначений для виконання графічного рендеру, що в свою чергу зменшує навантаження на центральний процесор. Так, центральний процесор може бути зайнятий виконанням інших задач, поки графічний процесор буде виконувати обробку графіки.

На основі можливостей, наданих CUDA [6] для програмування GPU, графічний процесор можна розглядати як набір багатоядерних процесорів із

спільною пам'яттю. Графічні процесори дотримуються парадигми паралельного програмування з кількома потоками даних з однією програмою (SPMD), у якій ядра виконують ту саму програму на кількох частинах даних, але не обов'язково виконують ту ж інструкцію одночасно [4]. CUDA складається зі стеку програмних рівнів, що включає в себе апаратний драйвер, програмний інтерфейс програми на мові C і драйвер CUDA, призначений для передачі даних між графічним процесором і центральним процесором.

Архітектура CUDA побудована навколо масштабованого багатопроцесорного масиву. Кожен мультипроцесор на графічних процесорах з підтримкою CUDA складається з восьми скалярних процесорів, а також додаткових блоків, таких як блок багатопоточних інструкцій і спільної мікросхеми пам'яті. Коли частина програми виконується багато разів з різними даними, її можна виділити у функцію, що називається функцією ядра, яка буде виконуватися на пристрої через багато різних потоків. Під час виклику функції ядра на GPU генерується велика кількість потоків. Група всіх згенерованих потоків називається сіткою, яка розділена на багато блоків. Кожен блок групує потоки, які виконуються одночасно на одному мультипроцесорі GPU. Немає фіксованого порядку виконання між блоками. Якщо на GPU достатньо мультипроцесорів, блоки виконуються паралельно

Основи трасування променів

Базова модель трасування променів випускатиме один промінь на піксель. Якщо розмір зображення 800x600 пікселів, то після завершення трасування буде знято 480000 променів. Кожен з них починається біля глядача й закінчується на найближчому перетині з об'єктом сцени. Розташування глядача визначається іншими об'єктами сцени у вхідному файлі. Для того, аби визначити скільки світла падає на цю точку та який колір буде отримано застосовується модель освітлення. Модель освітлення — це рівняння, яке використовується для обчислення інтенсивності світла, яке ми повинні бачити в даній точці на поверхні об'єкта [3].

Об'єкти у межах сцени мають властивості, що визначають колір, світловідбивання (дзеркальний), заломлення світла (як скло), а також розташування в сцені. Об'єктами можуть бути сфери, трикутники (багатокутники), кільця, циліндри тощо [1]. Будь-який з цих об'єктів також може бути джерелом світла (зональне джерело світла), випромінює світло на всій поверхні об'єкта. Наразі для нашої моделі освітлення ми будемо

використовувати точкові джерела світла, тобто світло, що надходить з однієї точки сцени.

Графічні процесори з підтримкою CUDA мають шість різних областей пам'яті: реєстри, локальна пам'ять, спільна пам'ять блоків, глобальна пам'ять, постійна пам'ять і пам'ять текстур.

Розрахунок найближчого перетину

Параметричні рівняння для лінії в тривимірному просторі використовуються для розрахунку близького відтинку об'єкта від умовного ока (глядача) [2].

$P_0 (x_0, y_0, z_0)$ – це розташування точки глядача в початковій точці променя. $P_1 (x_1, y_1, z_1)$ – це точка на площині зображення. $P (x, y, z)$ – це будь-яка точка на прямій, визначеній P_0 та P_1 зверніть увагу, що $(x_1 - x_0)$ це компонент вектора, те ж саме і для y та z . Отже, промінь можна представити вектором: $(x_1 - x_0), (y_1 - y_0), (z_1 - z_0)$.

$$x = x_0 + t * (x_1 - x_0)$$

$$y = y_0 + t * (y_1 - y_0)$$

$$z = z_0 + t * (z_1 - z_0)$$

Застосування освітлення

Тепер, коли знайдено розрахунок того, що глядач може побачити в конкретному пікселі, застосовується режим освітлення. Використаний режим локального освітлення визначає, якого кольору буде піксель.

Дифузний матеріал - це тьмянний матеріал, як крейда. У точці перетину створюється вектор від перетину до світла. Це утворює світловий вектор L . N є нормаллю поверхні в цьому перетині. Нормаль перпендикулярна до поверхні. Формула для розрахунку розсіяної складової моделі локального освітлення має такий вигляд [3]:

$$\text{Diffuse} = K_{\text{diffuse}} * \text{Color}_{\text{diffuse}} * \text{Cos}\theta$$

Колір дзеркала залежить від глядача. Чим ближче вектор відбиття R спрямований до ока, тим яскравішим стане піксель. Простіше кажучи, дзеркальний колір освітлює точку більше, якщо світло відбивається назад в око. Формула для розрахунку дзеркальної складової моделі локального освітлення має такий вигляд [3]:

$$\text{Specular} = K_{\text{Specular}} * \text{Color}_{\text{Specular}} * \text{Cos}\varphi$$

K_{diffuse} та $\text{Color}_{\text{diffuse}}$ є попередньо визначеними вхідними даними програми, що описують дифузні властивості конкретного об'єкта. Кут між L і

N дорівнює θ , який розраховується та змінюватиметься відповідно до розташування світла. Це надасть об'єкту затінений вигляд залежно від світла.

K_{Specular} , $\text{Color}_{\text{Specular}}$ і блиск є входними для опису об'єкта. Показник блиску впливає на дзеркальну пляму на об'єкті. Чим більший відблиск, тим концентрованіша пляма. ϕ — це кут між вектором нормалі N і вектором ока.

Пікселі будуть у кольоровому просторі «червоний-зелений-синій», відомому як (R, G, B). Кожен компонент RGB матиме діапазон [0,0 - 1,0]. Білий буде (1, 1, 1) і чорний (0,0,0). Формула моделі ілюмінатів виглядає:

$$\text{Pixel}_{\text{ColorR}} = \text{Diffused}_R + \text{Specular}_R$$

$$\text{Pixel}_{\text{ColorG}} = \text{Diffused}_G + \text{Specular}_G$$

$$\text{Pixel}_{\text{ColorB}} = \text{Diffused}_B + \text{Specular}_B$$

Глобальне освітлення дасть більш реалістичне зображення. Воно враховуватиме все світло, пряме та непряме, щоб сформувати кращу модель освітлення на поверхні. При локальному освітленні один промінь спрямовується на кожне світло, щоб обчислити, скільки світла падатиме на цю поверхню. При глобальному освітленні після обчислення перетину багато променів викидаються в різних напрямках, щоб створити світло, що падає на цю точку. Щоб створити максимально реалістичне зображення, слід протестувати всі напрямки світла. Це неможливо, оскільки на будь-яку окрему точку падає нескінченна кількість напрямків світла. Замість цього пробні промені знімаються для створення моделі освітлення [2].

Промінь може випадковим чином відбиватися від об'єкта, поки не досягне світла. Якщо він ніколи не досягає максимально дозволеного відскоку світла, він викидається з остаточного обчислення кольору пікселя. Якщо зразок потрапляє на світло безпосередньо, у розрахунок береться повна інтенсивність світла. Якщо воно не потрапляє на світло прямо, після кожного відбиття інтенсивність світла зменшується на коефіцієнт розсіяності компонента об'єкта, від якого воно відбивається. У глобальному освітленні вибірковий промінь займає місце вектора світла L у розрахунках кольору пікселя. Формула кольору пікселя тепер змінюється на наступну.

$$\text{Pixel}_{\text{color}} = \frac{1}{N} * \sum_{i=1}^{\#samples} (\text{Diffused}_i + \text{Specular}_i) * \text{gl}(\text{samples})$$

Трасування променів з використанням графічного процесора

Алгоритм трасування променів за своєю суттю чудово підходить для розпаралелювання з методами SPMD, оскільки розрахунок освітлення в кожному пікселі є незалежним процесом. Це робить можливим його

впровадження на графічному процесорі, використовуючи окремий потік для обчислення кожного променя.

На першому кроці алгоритму дані сцени завантажуються з текстового файлу. Він визначає елементи сцени (наприклад, вершини, точки, лінії, багатокутники, криві тощо) і матеріали елементів. Також на цьому етапі завантажується файл конфігурації, який містить параметри, необхідні для виконання алгоритму, такі як роздільна здатність зображення, максимальна кількість відскоків променів тощо.

На другому кроці будується структура просторового прискорення що відповідає єдиному розподілу. Алгоритм побудови сітки оптимізовано так, щоб кожен об'єкт у сцені пов'язувався з (у координатах сітки) рамкою, вирівняною по осі, яка його оточує. Потім отримуються просторові області-кандидати, які перекриваються зі згенерованим блоком. Для кожного регіону-кандидата перевіряється накладання регіону на об'єкт, і якщо це трапляється, об'єкт додається до регіону.

Третій етап передбачає передачу даних з пам'яті центрального процесора в пам'ять графічного процесора. Передаються такі дані: площина огляду, камера, роздільна здатність зображення, список трикутників і його нормалі, джерела світла, області сітки для просторового поділу та властивості матеріалу кожного об'єкта. Після копіювання даних у графічний процесор викликається ядро, яке обчислює основні промені. Даними, необхідними для розрахунку основних променів, є площина огляду, камера та роздільна здатність зображення. Основні промені є вхідними даними для розрахунку кожного кольору пікселя, що є ядром алгоритму трасування променів. Даними, необхідними для розрахунку кольору кожного пікселя, є список трикутників і його нормалі, джерела світла, області сітки для просторового поділу та властивості матеріалу кожного об'єкта.

Ядро, яке обчислює колір кожного пікселя, викликається після поділу на фрагменти (групи пікселів) зображення для візуалізації. Зображення рівномірно розділене, тобто кожен фрагмент має однакову кількість пікселів. Кожен фрагмент відповідає безпосередньо блоку потоків у CUDA, щоб обробити кожне розділення зображення іншим блоком. Крім того, оскільки кожен піксель зображення обробляється іншим потоком, кількість потоків на блок дорівнює кількості пікселів, що містяться в кожному розділі. З цієї причини поділ повністю встановлено при фіксації кількості потоків на блок і роздільної здатності зображення. Наприклад, якщо роздільна здатність зображення становить 640×480 пікселів, а розмір блоку дорівнює 16×8 потоків, зображення обов'язково потрібно розділити на 40×60 фрагментів.

Кожен кинутий промінь перетинає структуру просторового прискорення, слідуючи за відбиттям і заломленням в об'єктах. Реалізований алгоритм трасування променів має лише один тип елемента, трикутник. Таким чином, алгоритм перетину вимагає лише кількох арифметичних операцій.

Нарешті, після розрахунку кольору для кожного пікселя, дані, згенеровані графічним процесором, копіюються до центрального процесора для відображення на екрані.

Так, за рахунок того, що графічні процесори дотримуються парадигми паралельного програмування за методами SPMD, використання GPU в обробці алгоритму трасування променів дає значний приріст продуктивності відносно використання виключно центрального процесора. Це відбувається тому, що при використанні графічного процесора кількість потоків, що займається обрахуванням одного променя значно більша ніж кількість потоків центрального процесора. До прикладу сучасний центральний процесор топ сегменту має 64 ядра, в той час як відеоадаптер з сучасним графічним процесором топ сегменту має близько 4500 ядер. Це означає, що такий графічний процесор одночасно буде обраховувати значно більше променів ніж центральний процесор.

Висновок

Вивчення трасування променів може привести до цікавих результатів у сфері комп'ютерної графіки. Трасування променів є життєздатною технікою створення двовимірних зображень тривимірного світу. Це може бути інструмент, який стає більш цінним у міру того, як наша культура все глибше заглиблюється в створенні комп'ютером світи через ігри, фільми, навчальні тренажери чи навіть архітектурне моделювання. Трасування променів може створювати зображення з різним ступенем реалістичності. Також можна зробити висновок, що використання графічних процесорів в обрахуванні трасування променів досягає значно більшої продуктивності відносно використання лише центрального процесора.

Література

1. *Shirley, Peter. Realistic Ray Tracing. A K Peters, Natick, Massachusetts, 2000.*

2. *Larson, Roland. Hostetler, Robert. Edwards, Bruce.* Calculus Sixth Edition. Houghton Mifflin Company, Boston, New York, 1998
3. *Baker, M. Pauline. Hearn Donald.* Computer Graphics C Version. Prentice Hall, Upper Saddle River, New Jersey, 1986.
4. *Darema, F.,* The SPMD Model: Past, Present and Future, Proceedings of the 8th European PVM/MPI Users' Group Meeting, Lecture Notes in Computer Science 2131, 2001.
5. *Eric Haines, Tomas Akenine-Möller.* Ray Tracing Gems, 2019.
6. *Kandrot, E., Sanders, J.,* CUDA by Example: An Introduction to General-Purpose GPU Programming, 2010

К.т.н., доцент Боярінова Ю.Є., магістрант Воронін Микита

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**ЕФЕКТИВНИЙ СПОСІБ ОРГАНІЗАЦІЇ
ТЕЛЕКОМУНІКАЦІЙ В СПЕЦІАЛІЗОВАНІЙ
КОМП'ЮТЕРНІЙ МЕРЕЖІ РУХОМИХ ОБ'ЄКТІВ**

Abstract

Julia E. Boyarinova, assoc. prof., PhD; Mykita Voronin, student
**Effective method of organizing telecommunications in a specialized computer network
of movable objects**

The essay is dedicated to effective algorithm for telecommunication's organization at the special computer network of moving objects, based on the shortest spanning tree STA. Protocol of designing the structure of telecommunication network, based on Prim-algorithm, increases quality, under criteria the minimal probability of the opening of traffic between the object of computer network of the moving objects. Described algorithm may be used for the unpiloted aviation units UAV control.

Вступ

Події останнього часу показують важливість вивчення існуючих, навіть на перший зір, простих і недосконалих, інформаційних технологій управління рухомими об'єктами, особливо безпілотними літальними апаратами (БПЛА), на предмет виявлення недоліків, можливостей вдосконалення та розробки модернізованого програмно - апаратного забезпечення.

Сучасною тенденцією є заміна концепції інформаційної технології управління літальним апаратом як автономним рухомим об'єктом (РО), концепцією управління РО, із встановленими на них бортовими спеціалізованими комп'ютерними системами, об'єднаними в єдину комп'ютерну мережу РО (особливо така технологія є корисною при спільному проходженні БПЛА через систему протидії).

Необхідною умовою для реалізації такої мережевої інформаційної технології є існування стійкої телекомунікаційної мережі, яка поєднує спеціалізовані комп'ютери (далі – об'єкти), встановлені на РО, у єдину систему з використанням радіоефіру в якості середовища передачі даних.

Постановка задачі

Необхідно визначити способи покращення стійкості телекомунікаційної мережі рухомих об'єктів в умовах навмисних шкідливих впливів, підвищити захищеність комп'ютерної мережі рухомих об'єктів за рахунок впровадження ефективного алгоритму організації телекомунікаційної системи, яка базується на використанні радіоефіру в якості середовища передачі даних.

Задача є у визначенні алгоритму структурування телекомунікаційної системи РО, який дозволить мінімізувати ймовірність виявлення, зовнішнім інструментарієм, трафіку між об'єктами в радіоефірі, і при умові збереження цілісності (функціональності) системи [1].

Термінологія

АНР (Analytic Hierarchical Process) – метод аналітичної ієрархії – один з методів прийняття рішень при виборі альтернатив.

MST (Minimum Spanning Tree) – мінімальне зв'язуюче дерево, - остов графа, що має мінімальну можливу вагу, де під вагою дерева розуміється сума ваг його ребер.

RSTP - Rapid STP (Rapid Spanning Tree Protocol) - швидкісний протокол розгортаючого дерева) - версія протоколу STP з прискореною реконфігурацією дерева, що використовується для виключення петель (виключення дублюючих маршрутів) у з'єднаннях комутаторів Ethernet з дублюючими лініями.

STA (Spanning Tree Algorithm) - алгоритм зв'язуючого дерева, створює мережу у вигляді графа, вершинами якого є комутатори та сегменти мережі.

STP (Spanning Tree Protocol) - протокол остовного дерева - мережевий протокол, що працює на другому рівні моделі OSI/ISO.

Розв'язання задачі

Концептуальним критерієм ефективності структури, алгоритм синтезу якої відшукується в роботі, телекомунікаційної мережі рухомих об'єктів є мінімальна ймовірність виявлення, зовнішнім інструментарієм, трафіку між БПЛА в радіоефірі.

Для формалізації цього критерія вводимо такі припущення.

Припущення 1: ймовірність виявлення трафіку між n рухомими об'єктами, шляхом несанкціонованого сканування електромагнітного випромінювання в зоні їх перебування, є незменшуючою функцією узагальненої потужності P електромагнітного випромінювання передавачів.

Припущення 2: потужність заміряного випромінювання P у зоні перебування рухомих об'єктів суть сума потужностей p_i бортових антенних передавачів– генераторів електромагнітного випромінювання.

Тоді формуємо критерій для визначення ефективної структури, яку повинен синтезувати відшукуваний алгоритм:

$$(p_1 + p_2 + \dots + p_i + \dots + p_{n-1} + p_n) \rightarrow \min.$$

Концептуальним обмеженням задачі є умова збереження цілісності (функціональності) системи. Для формалізації цієї умови вводимо такі припущення.

Припущення 3: телекомунікаційна система вважається цілісною (такою, що може виконувати свої функції), якщо структура її така, що між кожною парою компонентів існує прямий або опосередкований зв'язок, тобто виконуються відповідні функції трьох нижніх рівнів моделі OSI/ISO.

Припущення 4: потужності p_i i -го бортового передавача достатньо для організації *point-to-point* зв'язку (функція канального рівня) від об'єкта i до довільного об'єкта j ($j \in \{1, 2, \dots, n\} \setminus \{i\}$), тобто достатньо для організації прямого (безпосереднього) зв'язку між об'єктами навіть на кінцях діаметра мережі [2].

Тоді маємо такі формалізовані умови, які накладаються на структуру телекомунікаційної системи, яку (структуру) синтезує відшукуваний алгоритм.

Структура системи повинна бути такою, щоб відповідна їй матриця

$$M_0 = M + M + \dots + M^{n-1}$$

не містила нульових елементів, де M – матриця суміжностей, що представляє структуру телекомунікаційної системи ($m_{ij} = 1$, якщо між об'єктом i та об'єктом j існує прямий канал; $m_{ij} = 0$, у іншому випадку).

Структура телекомунікаційної мережі представляється складовими матриці $X_{n \times n}$ ($x_{ij} = 1$, якщо відшукуваний алгоритм встановлює *point-to-point* канал між об'єктом i та об'єктом j ; $x_{ij} = 0$, у іншому випадку).

Очевидно, що найбільший трафік при виконанні вищенаведених формалізованих умов забезпечує повнозв'язна структура телекомунікаційної мережі.

Припущення 5: потужність p_i електромагнітного випромінювання передавача об'єкта i є функцією сумарної довжини інцидентних, тобто створюваних цим передавачем, безпроводових каналів (зокрема – радіоканалів).

В роботі наводиться і обґрунтовується процедура переходу від наведеного вище критерія до наступного:

$$((l_{12} \cdot x_{12})^2 + (l_{13} \cdot x_{13})^2 + (l_{14} \cdot x_{14})^2 + \dots + (l_{ij} \cdot x_{ij})^2 + \dots + (l_{nm} \cdot x_{nm})^2) \rightarrow \min,$$

де l_{ij} - довжина каналу зв'язку, зокрема – фізична, між i -м та j -м об'єктами.

Розглянуто алгоритми відшукування структур телекомунікаційних мереж, якість яких задовольняє сформованому критерію, і які об'єднані у клас STA-протоколів, тобто протоколів побудови найкоротших зв'язуючих дерев.

Представлено і розглянуто, поруч з класичними алгоритмами Ісау-Вільямса, Прима, Краскала, Вогеля [3], також STA для протоколів STP (RSTP) стандарту IEEE 802.1d [4].

Також розглянуто рекомендації по способу організації структури комп'ютерної мережі Блютуз, де кожна пара (рухомих) об'єктів групи *slave* зв'язана між собою через центральний об'єкт *maser* (стандарт IEEE 802.16).

Для остаточного прийняття рішень стосовно найпридатнішого алгоритму застосовано один з відомих методів прийняття рішень – АНР [5], оснований на формуванні експертами матриці

$$A_{s,s} = // w_i / w_j //_{s,s}$$

де w_i – об'єктивна вага алгоритму i ; w_i / w_j ($i, j \in \{1, 2 \dots s\}$), $w_k > 0$, $k \in \{1, 2 \dots s\}$; $\sum_k w_k = 1$.

Для знаходження об'єктивних значень ваг альтернатив було розв'язано по w_k , $k \in \{1, 2 \dots s\}$, таке рівняння

$$(A - n \cdot I) \cdot w = 0,$$

де I – одинична матриця, $w = (w_1, w_2 \dots w_k \dots w_{s-1}, w_s)^T$.

При цьому доведено, що дане матричне рівняння має нетривіальне рішення.

В результаті приходимо до обгрунтованої рекомендації використовувати в якості ефективного способу реалізації STA – протоколу в комп'ютерних мережах РО алгоритм побудови MST Прима.

Іншими словами, пропонується забезпечити ефективність нової, високозахищеної від демаскування, спеціалізованої комп'ютерної системи, шляхом застосування відомого класичного методу і відповідного алгоритму.

Висновки

Запропонований до реалізації протокол створення структури телекомунікаційного середовища, на базі одного алгоритму класу STA (рекомендації IEEE 802.1d), підвищує якість спеціалізованої комп'ютерної

мережі, за критерієм мінімальної ймовірності виявлення трафіку між об'єктами (зокрема – БПЛА), що досягається зменшенням потужності електромагнітних випромінювань системи безпроводової передачі даних при допустимому рівні функціональності мережі.

Слідування рекомендаціям IEEE 802.16 стосовно способу організації структури (теле)комунікацій спеціалізованої системи рухомих об'єктів, згідно з якою кожна пара рухомих об'єктів (БПЛА) комунікує між собою через центральний об'єкт (master), може привести до перевищення рівня потужності електромагнітного випромінювання, з чого, в свою чергу, слідує передчасне демаскування системи рухомих об'єктів.

Запропоновано використовувати в якості способу реалізації STA – протоколу в комп'ютерних мережах рухомих об'єктів алгоритм побудови MST Прима.

Література

1. Чернега В., Платтнер Б. Безпроводні локальні комп'ютерні мережі: навчальний посібник/ МОН МС України. – Київ: Кондор, 2013. – 238 с.
2. 802.11 Wireless LAN Fundamentals. Peiman Roshan, Jonatan Leary. – Cisco Press, 2004. – 304p.
3. Зайченко О.Ю., Зайченко Ю.П. Комп'ютерні мережі. Навчальний посібник для студентів вищих навчальних закладів. – К.: Видавничий дом «Слово», 2010. – 520с.
4. Internetworking Technologies Handbook. Forth Edition. Cisco Systems, et. al – Cisco Press, 2005.- 1040p.
5. Катренко А.В., Пасічник В.В., Пасько В.П. Основи теорії прийняття рішень: Підручник. – К.: Видавнича група ВНУ, 2009. – 352с.

К.т.н, доцент Боярінова Ю.Є., студент Пушняк Д.С.

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

**КОМП'ЮТЕРНА СИСТЕМА ОБ'ЄКТНО-
РЕЛЯЦІЙНОГО ВІДОБРАЖЕННЯ ІЗ
АВТОМАТИЧНОЮ ГЕНЕРАЦІЄЮ МІГРАЦІЙ НА
ОСНОВІ ЗМІН В МОДЕЛЯХ ДАНИХ**

Abstract

Yuliia Y. Boiarinova, assoc. prof., PhD; Dmytro Pushniak, student
*Computer system of object-relational mapping with automatic generation of
migrations based on changes in data models*

This paper concerns the task of automatic generation of migrations based on changes in entity models. Situations where projects are using raw SQL queries alongside with ORMs, which provide some useful automations and an additional level of abstraction over the database driver, are not rare. In this case, after creating or updating entity data model in code, the common solution would be manually creating a migration which does the same modification to table on database level. The alternative solution with automated migrations generation is proposed. The ways for further improvements are suggested as well.

Вступ

Актуальність теми обумовлена тим, що велика кількість нових середніх та малих проєктів, в особливості стартапів, яким необхідні інтенсивні темпи розробки, під час роботи із реляційними базами даних зупиняють свій вибір на використанні систем об'єктно-реляційного відображення. Як правило, такі системи вирішують велику кількість проблем перетворення сутностей бази даних в об'єкти моделей відповідно до об'єктно-орієнтованої парадигми, на якій побудований та якою маніпулює сервіс. У порівнянні з традиційними методами обміну між об'єктно-орієнтованою мовою і реляційною базою даних, система об'єктно-реляційного відображення зменшує кількість коду, який необхідно створити [1]. Крім того, такі системи допомагають полегшити створення запитів до бази даних, даючи можливість створювати динамічні запити, транзакції та міграції в об'єктно-орієнтованому стилі із використанням синтаксису тієї самої мови програмування, замість використання “сирого” SQL. Основна ідея полягає в тому, що такі системи

потрібні для полегшення роботи розробникам шляхом економії їх часу. Отже, подальше розширення можливостей, що допомагатимуть бізнесу економити ресурси, позбавляючи розробників потреби в низькорівневій архітектурі та даючи можливість замість цього зосередитись саме на бізнес-логіці, є актуальним. У даній статті пропонується механізм автоматичної генерації міграцій для реляційної бази даних після змін в кодї моделі даних засобами комп'ютерної системи об'єктно-реляційного відображення (проєкції), який дозволить економити час розробників.

Постановка задачі

Задача полягає в розширенні можливостей деяких сучасних систем об'єктно-реляційного відображення (проєкції) додаванням нового опціонального функціонала для автоматичної генерації міграцій для реляційної бази даних після змін в кодї моделі даних, забезпечивши надійність вихідних міграцій та консистентність бази даних.

Термінологія

Система об'єктно-реляційного відображення (проєкції), ORM — це техніка програмування для перетворення даних між несумісними системами типів в об'єктно-орієнтованих мовах програмування. Це створює, по суті, "віртуальну об'єктну базу даних", яка може бути використана зсередини мови програмування [2].

Міграцією називається інструмент керування інкрементними, зворотними змінами схеми реляційної бази даних. Міграція схеми виконується над базою даних коли необхідно оновити стан схеми бази даних до якоїсь новішої, або повернути до старішої версії.

Опис алгоритму

Суть способу полягає в тому, щоб при кожному запуску сервісу, в процесі ініціалізації системи об'єктно-реляційного відображення виконувати наступні кроки у вказаному порядку:

1. Встановити з'єднання із базою даних.
2. Перевірити наявність таблиці, в якій зберігатимуться стани всіх моделей даних.

Наступні дії залежать від факту наявності потрібної таблиці:

- якщо таблиці не існує, то створити її. Перейти до кроку 4;
- якщо таблиця вже існує, то завантажити всі останні стани моделей в оперативну пам'ять в "сирому" вигляді.

3. Провести структурний аналіз завантажених станів моделей, встановити відповідність між таблицею бази даних та моделлю бази даних, типом моделі даних, обмеженнями тощо.
4. Завантажити всі моделі даних з коду в їх поточному стані. Відрізнити саме моделі від звичайних класів, які не відносяться до стану бази даних можна шляхом зберігання таких класів в окремих модулях зі спеціальним ім'ям, яке містить в собі “.entity.”, наприклад, у випадку мови програмування TypeScript, це може бути “name.entity.ts”.
5. Провести структурний аналіз завантажених поточних станів моделей, встановити відповідність між таблицею бази даних та моделлю бази даних, типом моделі даних, обмеженнями тощо. Порівняти чи всі дані в моделі (тип, обмеження) вказані коректно та підтримуються базою даних. Якщо ні, то повідомити про це користувача.
6. Якщо на кроці номер 2 було завантажено минулий стан бази даних, то провести порівняння поточного стану із минулим. Знайти всі відмінності у моделях.

Далі:

- якщо відмінності були знайдені, то на їх основі створити файли міграції, які міститимуть набір команд для бази даних, щоб синхронізувати стан всіх таблиць із поточним станом моделей даних. Також має бути набір інструкцій, щоб за потреби відмінити цю міграцію та повернутись до минулого стану бази даних;
 - якщо відмінностей немає, завершити виконання алгоритму.
7. Якщо на кроці номер 2 не було завантажено минулий стан бази даних, то створити міграції які ініціалізують базу даних поточним станом моделей даних.
 8. Зберегти поточний стан моделей даних у “сирому” вигляді в таблицю. Ці моделі стають останньою версією моделей до наступного запуску сервісу.

Висновки

При використанні описаного механізму, отримуємо полегшення та підвищення швидкості внесення розробником змін в ділянки коду, які тісно пов'язані зі станом бази даних.

У подальшому необхідно вдосконалити наявний аналізатор для унікального ідентифікування кожного окремого поля класу моделі даних, адже на даному етапі ідентифікація відбувається за порядком їх розташування в моделі, що не є допустимим, адже у випадку якщо одночасно перейменувати поля та змінювати їх порядок, це може призвести до неочікуваних результатів у вихідних міграціях.

Література

1. Solving the Java Object Storage Problem. [Електронний ресурс] – Режим доступу: <https://dl.acm.org/doi/10.1109/2.730734> (дата звернення: 28.10.2022).
2. *Roebuck K. Object-Relational Mapping (Orm) : High-Impact Strategies - What You Need to Know: Definitions, Adoptions, Impact, Benefits, Maturity, Vendors. / K. Roebuck // Samford : Tebbo, 2011. 632 с.*

УДК 004.42

К.т.н., доцент Боярінова Ю.Є., студент Шимчук Т.В.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

СЕРЕДОВИЩЕ ДЛЯ БЕЗПЕЧНОГО ВИКОНАННЯ РОЗШИРЕНЬ ЗАСТОСУНКІВ

Abstract

Yuliia Y. Boyariniva, assoc. prof., PhD; Tymofii Shymchuk, student
Environment for running potentially dangerous plug-ins of the applications

This paper concerns the task of describing the safe environment for executing potentially dangerous extensions for the applications. Untrusted code of the programs is usually able to use all the available system resources which is potential security threat. Described the structure of the system which prevents threats of untrusted extensions.

Вступ

Програмне забезпечення може виконувати не лише обмін та обробку даних. Деякі застосунки реалізують функціонал виконання стороннього програмного коду, який інтегрується у вигляді плагінів, що під'єднуються у вигляді динамічних бібліотек; сценаріїв, створених на скриптових мовах програмування. Такі частини програмного коду, призначені у якості розширення до основного застосунку, можуть створюватися сторонніми авторами, якими є їх користувачі, та, в тому числі, зловмисники, що зацікавленні в написанні потенційно шкідливого коду.

Для забезпечення надійності та безпечності динамічно під'єднаних модулів, що виконують роль розширень, можуть використовуватися віртуальні машини, що надають програмісту обмежений доступ до ресурсів системи та застосунків.

Постановка задачі

Задача полягає в дослідженні та описі програмної системи, що ізолює потенційно небезпечні розширення застосунків від зовнішнього середовища, контролює доступ до системних ресурсів та запобігає безпосередній взаємодії із ними.

Термінологія

Віртуальна машина – програмна система, задачею якої є виконання програмного коду у платформонезалежному оточенні [1].

Розширення або *плагін* – частина застосунку, що розширює його основний функціонал. Розширення можуть бути підключені до основної частини програми різними способами, наприклад, через код динамічної бібліотеки, що реалізує певний інтерфейс [2].

Опис програмної системи

Застосунки, що мають можливість розширення власного функціоналу за допомогою плагінів, у більшості випадків під'єднують їх як динамічні бібліотеки або інтегрують скрипти так, що додатковий програмний код виконується у тому ж оточенні, що й і основна програма. Тобто, розширення працює у тому ж процесі та має доступ до тих же самих ресурсів, що й і основна частина застосунку, такі як процесор, пам'ять, файлова система, мережа та системні пристрої, у тому числі доступ до мережі. Таким чином, сторонні модулі можуть виконувати потенційно зловмисні інструкції. Прикладом таких небезпечних та шкідливих дій є:

1. Використання ресурсів системи для сторонніх обчислень. Завдяки доступу до глобальної мережі, розширення приймає завдання на сторонні обчислення, виконує їх та надсилає результати.

2. У випадку серверної системи, атака на сервіси, частиною яких є розширення, чи запуск на тій же системі, шляхом вичерпування системних ресурсів: використання процесорного часу, використання великої кількості пам'яті, активний запис на постійні носії, використання мережі, тощо.

3. Крадіжка даних, якими можуть бути особисті файли користувача, паролі, облікові записи, інші конфіденційні дані.

4. Для деяких програмних систем, що виконуються інтерпретаторами скриптових мов програмування або віртуальними машинами, існує можливість динамічного підвантаження коду, у тому числі завантаженого через мережу. В свою чергу, такий код може бути введений зловмисником через інтерфейси, що початково не передбачають такого використання. Прикладом є вразливості виконання віддаленого коду у бібліотеці логування log4j для мови програмування Java [3].

Типовим вирішенням проблеми можуть бути використання перевірених розширень, створених лише перевіреними авторами. Файли таких модулів додатково підписують цифровим підписом для ідентифікації їх походження. Проте, це не виключає помилок у коді та наявності шкідливих фрагментів (наприклад, сторонніх бібліотек).

У контексті поставленої задачі, розширення для застосунків можна вважати такими, яким не можна довіряти. Загалом, вимоги до системи для виконання потенційно небезпечного коду повинні бути такими:

1. Кожна потенційно небезпечна інструкція проходить перевірку. Такими інструкціями є операції роботи із пам'яттю. Вичерпання динамічної пам'яті та пам'яті на стеку контролюється і не призводить до аварійного завершення основної програми. Несанкціонований доступ до пам'яті повинен бути забороненим: так як у більшості операційних систем спроба доступу до забороненої для даного процесу адреси призводить до її завершення, тому прямий доступ через вказівники не підтримується. Для масивів контролюється вихід за його межі.

2. Безпосередній доступ до системних ресурсів, таких як файлова система, мережа, периферійні пристрої заборонений. Замість цього основна програма або середовище виконання декларує інтерфейс взаємодії із такими ресурсами, при цьому здійснюючи перевірку даних, якими оперує розширення.

3. Для запису даних використовується інтерфейс, що дозволяє зберігати дані на постійний носій. При цьому розширення не знає про структуру файлової системи.

4. Для взаємодії із мережею створені програмні інтерфейси, що виконують функцію формування трафіку, його шифрування і т. д.. Для розширення доступна можливість лише передавання та отримання даних через інтерфейс так, щоб вони могли проходити перевірку. Таким чином, доступ до сторонніх ресурсів мережі не є можливим.

5. Використання таких ресурсів, як оперативна та постійна пам'ять є контрольованим та не призводить до вичерпування ресурсів системи.

Із огляду на це, такі розширення не можуть бути написані мовами програмування, що транслюються в машинний код, так як такий підхід дозволяє доступ до системних викликів та адресного простору.

Оскільки код модуля повинен виконуватися у тому ж процесі, що й і основний код програми, потрібна інтеграція інструменту, що виконує роль оболонки для коду розширення. Це завдання може виконувати віртуальна машина, створена спеціально для цілей ізоляції системних ресурсів та контролю за їх використанням. Для доступу до ресурсів системи віртуальна машина або ж сам застосунок реалізує інтерфейс, за допомогою якого код розширення використовує надані йому можливості.

При створенні програмного інтерфейсу для подальшого використання кодом розширення, варто зауважити, що використання такого інтерфейсу може спричинити неконтрольоване використання ресурсів, оскільки частина реалізації може не виконуватися віртуальною машиною і не може бути перевірена нею. У такому випадку частину інтерфейсу варто написати мовою програмування для даної віртуальної машини так, щоб

контролювалося виділення ресурсів, а найбільш дрібні та малозатратні операції відводити до головної частини додатку. Таким чином, схема взаємодії повинна складатися із трьох рівнів:

1. Частина інтерфейсу доступу, реалізована на рівні основної програми. Повинна виконувати найбільш прості та малозатратні операції. Код розширення немає безпосереднього доступу до цього рівня.

2. Частина інтерфейсу доступу, реалізована на рівні віртуальної машини. Використовує рівень 1 та використовується рівнем 3.

3. Код розширення. Взаємодіє із основною програмою через посередництвом рівня 2.

Віртуальна машина може складатись як із інтерпретатора скриптової мови програмування, так і використовувати байт-код або інший вид серіалізації.

Висновки

При аналізі проблеми забезпечення безпечного виконання сторонніх розширень, яким не можна довіряти, вирішено, що код розширення повинен контролюватися для коректної обробки помилок при недоцільному використанні динамічної пам'яті та стеку. Доступ до системних ресурсів повинен контролюватися і відбуватися через використання спеціальних інтерфейсів, що запобігають несанкціонованому доступу до них.

Із урахуванням особливостей архітектури сучасних обчислювальних систем, доцільним рішенням є використання віртуальних машин для запобігання виконанню потенційно небезпечного коду.

В подальшому доцільним є більш детальний опис та розробка методів та алгоритмів виконання коду розширень та коректної обробки помилок.

Література

1. Craig, Iain D. Virtual Machines. Springer, 2006, – С. 20-24.
2. Secure loading of libraries to prevent DLL preloading attacks. URL: <https://support.microsoft.com/en-us/topic/secure-loading-of-libraries-to-prevent-dll-preloading-attacks-d41303ec-0748-9211-f317-2edc819682e1> (дата звернення: 31.10.2022).
3. Apache Log4j remote code execution vulnerability — Log4Shell. URL: [ibm.com/support/pages/apache-log4j-remote-code-execution-vulnerability-log4shell](https://www.ibm.com/support/pages/apache-log4j-remote-code-execution-vulnerability-log4shell) (дата звернення: 31.10.2022).

УДК 004.7

К.т.н., доцент Боярінова Ю.Є., студент Юрчишен Ф.В.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

ЗАПОБІГАННЯ БРАНДМАУЕРАМ UDP

Abstract

Yulia Boyarinova, assoc. prof., PhD; Fedir Yurchyshen, student
UDP firewalls avoiding

This paper addresses the problem of frequently blocked UDP traffic in public places such as airports, cafes, shops, etc. Discusses how to avoid Internet connectivity issues when using UDP-based networking tools or applications. A lightweight client-server solution for masking UDP traffic is proposed.

Вступ

Наразі системні адміністратори підприємств, що підтримують публічні мережі, часто блокують UDP трафік у цілях забезпечення безпеки та надійності їхньої мережі. Інакше, наприклад, мережа може бути використана для здійснення DDoS атаки [1].

Однак, UDP має низку переваг над іншим поширеним протоколом транспортного рівня: TCP. Його зручно використовувати для задач, коли швидкість передачі даних переважає потенційну ймовірність втрати частини або всіх даних, наприклад, аудіо чи відео стрімінг, а також для будування VPN тунелю.

Існують інструменти, що дозволяють замаскувати UDP трафік під інший нешкідливий трафік, але їх вони мають досить складний шлях для конфігурування для використання, наприклад, з поширеними VPN сервісами.

Запропоноване рішення створено для більш зручної роботи з VPN сервісами.

Постановка задачі

Задача полягає у розробці інструменту для маскуванню трафіку найпоширенішого протоколу, що використовується VPN сервісами, а саме

UDP, під інший нешкідливий трафік, що зазвичай не блокується, а також забезпечення зручної конфігурації.

Термінологія

VPN (Virtual Private Network) - узагальнена назва технологій, що дозволяють створювати віртуальні захищені мережі базуючись на інших мережах. Забезпечує безпечне передавання інформації за допомогою шифрування, що створює закритий для сторонніх канал обміну інформацією [2].

UDP (User Datagram Protocol) - один із основних комунікаційний протоколів стеку протоколів TCP/IP, що здійснює обмін повідомленнями без підтвердження та гарантій доставки [3].

Опис інструменту

Розроблюваний інструмент має клієнт-серверну архітектуру.

Клієнтська частина виступає у ролі шифратора, що пакує корисні дані у TCP заголовок та відправляє їх на заздалегідь вказану адресу сервера, що розшифровує повідомлення та витягує корисні дані, після чого передає дані кінцевому отримувачу.

Оскільки розроблюваний інструмент має втручатися у транспортний рівень мережових повідомлень (дейтаграми), то є необхідність використання так званих сирих сокетів.

C# має високопродуктивну бібліотеку та безпечне оточення для роботи із низькорівневим програмуванням сокетів.

Будь-який VPN сервіс складається із клієнту та серверу.

Клієнт зазвичай працює на задньому фоні комп'ютерної системи користувача.

Сервером виступає незалежна комп'ютерна система, яка фізично може знаходитися будь-де, проте має доступ до мережі Інтернет.

Загалом, роботу VPN сервісу можна описати наступними кроками [2]:

1. Клієнт VPN сервісу зашифровує трафік (увесь чи лише заздалегідь визначений) та перенаправляє його на сервер замість початкового призначення. Дані проходять через ISP (Internet Service Provider), але завдяки шифруванню дані майже неможливо розібрати.
2. VPN сервер отримує дані та дешифрує їх.
3. VPN сервер пересилає дешифровані дані у місце початкового запиту та отримує відповідь, що мав би отримати користувач.

4. VPN сервер зашифровує дані знову та надсилає зашифровану відповідь клієнту.
5. VPN клієнт отримує дані та дешифрує їх, після чого користувач має змогу зрозуміти відповідь з Інтернету.

Насправді, VPN не будує фізичного тунелю, а лише забезпечує безпечність даних, навіть якщо вони можуть бути потенційно вкрадені, за допомогою шифрування.

Зважаючи на це, можна безпечно додати ще клієнта та сервер, метою яких буде маскуванню трафіку, без дійсної зміни самих корисних даних.

Загалом, маскуванню відбувається наступним чином:

1. Клієнт інструменту перехоплює трафік, що має бути надісланий у Інтернет, замасковує UDP заголовки під TCP заголовки та надсилає замаскований трафік серверній частині інструменту.
2. Север отримує дані, дешифрує дані та відправляє далі за адресою призначення у Інтернеті.

Для комбінованої роботи VPN сервісу та маскувального інструменту необхідно, щоб обидва клієнти працювали на стороні цільового користувача, а обидва сервера працювали на віддаленій комп'ютерній системі.

Таким чином, комбінована система VPN та маскувального інструменту матиме такий вигляд (рис. 1):

1. VPN клієнт перехоплює вихідний трафік та закодує його.
2. Клієнт маскувального інструменту замінює UDP заголовки на TCP та відправляє дані на віддалену комп'ютерну систему.
3. Сервер маскувального інструменту розшифровує змінені заголовки, а VPN сервер у свою чергу декодує корисні дані, що закодував VPN клієнт. Далі VPN сервер відправляє декодовані дані далі у Інтернет опосередковано на запиті користувача.
4. Після отримання відповіді від Інтернет ресурсу, VPN сервер кодує дані, сервер маскувального інструменту замінює заголовки протоколу транспортного рівня та надсилає дані клієнтам.
5. Клієнт маскувального інструменту розшифровує заголовки, а клієнт декодує власне корисні дані.

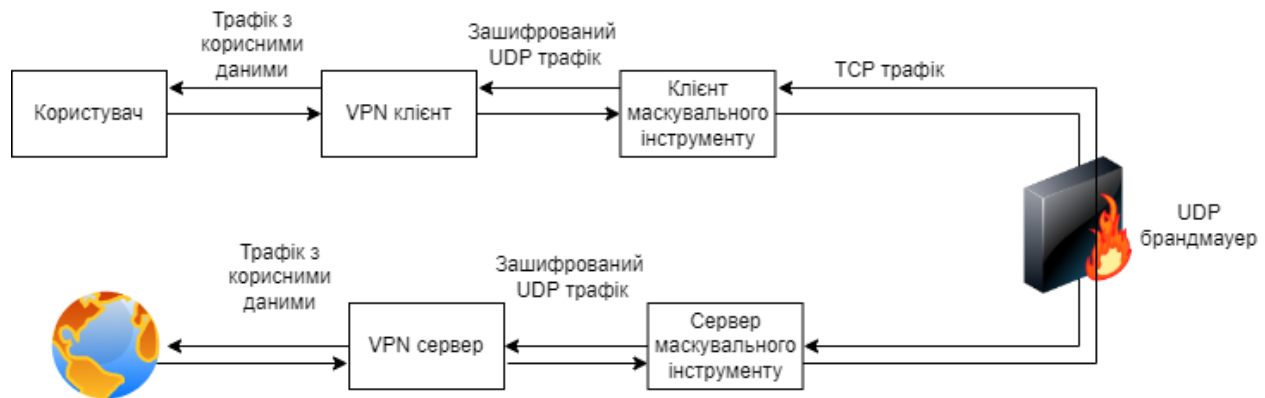


Рис. 1. Комбінована система VPN та маскувального інструменту

Висновки

Запропонований алгоритм спрощує процес конфігурування сервісу при сумісній роботі з VPN сервісом, а також має змогу працювати окремо, але недоліком є те, що необхідно мати доступ до віддаленого серверу поза межами публічної мережі, що блокує певний трафік.

Наявна можливість покращення інструменту шляхом підтримки інших протоколів, наприклад ICMP.

Література

1. NTP amplification DDoS attack. [Електронний ресурс] – Режим доступу: <https://www.cloudflare.com/learning/ddos/ntp-amplification-ddos-attack/> (дата звернення: 28.10.2022).
2. What is a VPN? How Does it Work and Why Do You Need One? [Електронний ресурс] – Режим доступу: <https://vpnoverview.com/vpn-information/what-is-a-vpn/> (дата звернення: 28.10.2022).
3. What is UDP? [Електронний ресурс] – Режим доступу: <https://www.cloudflare.com/learning/ddos/glossary/user-datagram-protocol-udp/> (дата звернення: 28.10.2022).

УДК 004.051

Ст. викладач Дробязко І.П., студент Брославський Р. М.

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

СИСТЕМА МОНІТОРИНГУ ПРОЦЕСУ ЗРОШЕННЯ СІЛЬСЬКОГОСПОДАРСЬКИХ КУЛЬТУР

Abstract

Iryna P. Drobiazko, senior lecturer; Rostyslav M. Broslavsky, student
Irrigation monitoring system

This paper concerns the task of modifying the irrigation monitoring system to make it more energy efficient and extend the battery life. Analyze of energy efficiency of both the default irrigation monitoring system and the modified one is discussed and fulfilled. The further ways possible modification researched and proposed.

Вступ

Сільське господарство відіграє надзвичайно важливу роль в економіці країни та життєдіяльності людини. Тому підвищення його ефективності шляхом використання засобів автоматизації і сучасних інформаційних технологій на різних етапах виробництва сільськогосподарської продукції є наразі актуальною задачею. ІоТ технологія (Інтернет речей) також знаходить своє застосування в цій галузі [1]. Пропонується ІоТ система для моніторингу процесу зрошення сільськогосподарських культур. В даній статті досліджується енергоефективність системи моніторингу та пропонуються способи зниження енерговитрат на функціонування системи, що є на часі.

Постановка задачі

Задача полягає у підвищенні ефективності системи моніторингу процесу вирощування сільськогосподарських культур, зокрема зниженні її енерговитрат, шляхом оптимізації алгоритмів роботи системи.

Термінологія

Zigbee – бездротовий протокол передачі даних [2].

БМС – бездротова мережа сенсорів.

Опис системи моніторингу

До складу системи входять: датчики (сенсорні вузли), вузли-маршрутизатори, головний вузол-маршрутизатор та вузол-координатор. Зокрема, дослідне поле площею 20×20 м² містить 16 сенсорних вузлів, кожен з яких відповідає за збір кліматичних показників, таких як температура повітря, вологість повітря та вологість ґрунту. Територія може бути потенційно розширена до реальних середніх за розмірами комерційних посівних площ сільськогосподарських культур.

Вузли датчиків підключаються за допомогою бездротовому зв'язку до одного з вузлів-маршрутизаторів. Зібрані дані з сенсорних вузлів (кліматичні показники) передаються через вузли-маршрутизатори до центрального маршрутизатора. Головний вузол-маршрутизатор передає їх на вузол-координатор, який знаходиться на базовій станції (наприклад, у фермерському будинку).

Запропонована топологія розглядає тільки стаціонарні вузли-маршрутизатори для зв'язку між сенсорними вузлами та вузлом-координатором. Крім того, не розглядається можлива зміна ролей вузлів-маршрутизаторів, оскільки вона стосується більш високих рівнів системи, таких як управління мережею.

Оскільки відстань між фермерським будинком та полем сільськогосподарських культур є невеликою, для передавання даних можна використовувати такий протокол бездротового зв'язку, як Zigbee (XBee S2C). Він гарантує роботу на необхідній відстані у зовнішніх умовах використання [2], низьке енергоспоживання, низьку вартість та роботу від батареї у бездротовій мережі сенсорів (БМС). З точки зору енергоспоживання, Zigbee (XBee S2C) (споживає 35 мВт), є кращим за протоколи LoRa, SigFox, WiFi, та GPRS, які споживають відповідно 100, 122, 835 та 560 мВт [3]. Тим не менш, розробка може бути реалізована з урахуванням майбутніх бездротових технологій, включаючи LoRa [3].

Система передбачає мінімальний інтерфейс та з'єднання між датчиками та мікроконтролером і бездротовими каналами зв'язку. Цифровий датчик вологості-температури та датчик вологості ґрунту потребують лише три дроти для фізичного інтерфейсу з мікроконтролерами: напруга живлення, заземлення та вихідні дані. Перші два дроти використовуються для живлення датчиків, тоді як третій – це аналогові та цифрові послідовні вихідні сигнали для датчика вологості ґрунту та температури відповідно. Мікроконтролер вузлів датчика та маршрутизатора зв'язується з бездротовим протоколом Zigbee через одно провідну шину даних. Вузли датчика та маршрутизатора закріплюються на поверхні на рекомендованій висоті 1,5 м над землею для уникнення впливу зони Френеля або відбиття сигналу. Живлення вузлів датчика та

маршрутизатора на досліджуваній ділянці здійснюється від акумуляторної батареї. Вузол координатора підключений до ноутбука/ПК базової станції, тому його енергопостачання не розглядаємо. Персональний комп'ютер оснащений програмним забезпеченням з графічним інтерфейсом для моніторингу дотримання кліматичних умов на полі фермерського господарства. Вузли датчиків та маршрутизатора розраховані на роботу в денному та нічному режимах. Вдень датчики, мікроконтролер та XBee S2C живляться від сонячної енергії. Крім того, сонячна батарея, яка розміщена перед коробкою сенсорного вузла, забезпечує зарядження акумуляторних батарей сенсорного та маршрутизаторного вузлів. Це також дозволяє знизити енерговитрати. Для подальшого зниження енергоспоживання сенсорного та маршрутизаторного вузлів пропонується використання у нічний час двох алгоритмів роботи системи, а саме: “сон”/”пробудження” і “мінімізація надлишковості даних”.

Опис алгоритму

Алгоритм функціонування системи з використанням “мінімізації надлишкових даних” та режиму роботи “сон”/”пробудження”, який розроблено та реалізовано в мікроконтролері Atmega328p сенсорного вузла для зменшення енергоспоживання, передбачає виконання наступних дій:

1) Спочатку відбувається вихід мікроконтролера Atmega 328p зі сплячого режиму.

2) Всі компоненти сенсорного вузла (тобто датчики, мікроконтролер та XBee S2C) забезпечуються енергією від сонячної батареї (12 В / 5 Вт) в денний час для зарядки своїх акумуляторів. Батареї ж використовуються для живлення сенсорного вузла в нічний час.

3) Мікроконтролер вимірює кліматичні показники (тобто температуру повітря, вологість повітря та вологість ґрунту).

4) Мікроконтролер визначає різницю між попереднім і наступним значенням вологості ґрунту з метою виявлення надлишкових даних.

5) Якщо різниця між двома значеннями дорівнює нулю або менша чи дорівнює 7% (пороговий рівень), датчики та модуль XBee S2C переходять в сплячий режим. Крім того, мікроконтролер переходить в режим енергозбереження для економії енергії сенсорного вузла. У такому випадку, ґрунт є достатньо вологим, і дані з вузла датчика на вузол маршрутизатора не передаються. Крім того, не потрібно зрошувати ґрунт, а це призведе до економії води. Компоненти сенсорного вузла залишаються в сплячому режимі до моменту перевищення порогового рівня. Невелика різниця значень 7% вибрана для отримання більш точного рішення.

У сільському господарстві робота системи зрошення планується і залежить від результатів вимірювань вологості ґрунту. Коли ґрунт сухий, зрошувальна система працює. В іншому випадку (тобто вологий ґрунт), зрошувальна система вимкнена. Тому алгоритм “мінімізації надлишкових даних” в даному дослідженні базується саме на результатах вимірювання вологості ґрунту. Використовуючи цей показник, планування зрошення здійснюється таким чином, щоб підтримати показники вологості ґрунту еквівалентними або близькими до потенціалу поля, тобто задовольнити потреби сільськогосподарських культур у воді. В подальшому, розглядатимуться кілька датчиків для фіксації таких параметрів, як температура ґрунту, провідність ґрунту, засоленість, вологість листя та датчики опадів.

б) Якщо ж різниця між двома значеннями більше 7%, всі компоненти сенсорного вузла “не сплять”. Виміряні на кроці 3 дані передаються з сенсорного вузла до вузла-координатора через вузли-маршрутизатори. Після завершення процесу передачі, датчики та модуль XBee S2C переходять у сплячий режим. Крім того, мікроконтролер переходить в режим енергозбереження для економії енергії сенсорного вузла.

Дослідження роботи сенсорного вузла дало наступні результати. Вузол датчиків передає виміряні дані про кліматичні умови кожні 15 хв (900 с) протягом 2 с, тобто з надзвичайно низьким енергоспоживанням. Датчики потребують 1 с для вимірювання даних про кліматичні умови. Крім того, ще одна секунда необхідна для передачі даних на відповідний вузол маршрутизатора. Все вказує на те, що 2 с витрачаються кожним сенсорним вузлом на вимірювання та передачу кліматичних умов на відповідний вузол маршрутизатора. Отже, кожен сенсорний вузол прокидається на 2 с і засинає на 898 с.

7) Чотири сенсорні вузли взаємодіють за допомогою одного вузла-маршрутизатора. Таким чином, виділений для кожного вузла-маршрутизатора час становить 16 с. З цих 16 секунд 8 секунд припадають на чотири сенсорні вузли, а 8 секунд для уникнення колізії даних (2 с після передачі даних кожного сенсорного вузла). Вузли-маршрутизатори збирають і передають дані відповідних сенсорних вузлів на головний вузол маршрутизатора протягом 16 с, а потім переходять в сплячий режим. Отже, кожен вузол маршрутизатора прокидається на 16 с і спить протягом 884 с.

8) Головний вузол маршрутизації збирає отримані від чотирьох вузлів маршрутизації дані і передає ці дані вузлу-координатору протягом 64 с. Після передачі головний вузол-маршрутизатор переходить у сплячий режим на 836 с.

Висновки

Запропоновано і практично реалізовано енергоефективну систему моніторингу кліматичних умов вирощування сільськогосподарських культур, включаючи температуру повітря, вологість повітря та вологість ґрунту. Апаратне забезпечення системи підібрано з урахуванням критеріїв енергоефективності з метою мінімізації енергоспоживання різних вузлів БМС. Крім того, пропонуються два способи зниження енергоспоживання вузлів датчика, маршрутизатора та головного маршрутизатора, а саме використання режиму роботи “сон/пробудження” і алгоритму “мінімізації надлишкових даних”. Використання цих засобів дозволяє зменшити час активної роботи компонентів системи та їх енергоспоживання, а також ефективність системи в цілому, що підтверджують результати виконаного дослідження. Економія електроенергії збільшується порівняно з традиційним режимом роботи, оскільки датчики вмикаються тільки при необхідності. Крім того, мінімізовано обмін даними між компонентами системи. Запропонована система дозволяє здійснювати збір даних для підтримки прийняття рішень і може бути використана при автоматизації процесів зрошення на сільськогосподарських полях. Використання енергоефективної технології сприятиме високопродуктивному та сталому землеробству.

Література

1. *Prasant Kumar Pattnaik, Raghvendra Kumar, Souvik Pal, S. N. Panda, IoT and Analytics for Agriculture (Studies in Big Data Book 63), 2020.*
2. *Agus Kurniawan, Digi Xbee3 ZigBee 3 Development Workshop Kindle Edition, 2019.*
3. *GAO ZE HUA SUN WEN SHENG, Things - the architecture. a wireless communication protocol standards (RFID. NFC. LoRa. NB-IoT. WiFi. Zi), 2020.*

УДК 004.75

Ст. викладач Дробязко І. П., магістрант Туркін М. П.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

АНАЛІЗ ФРЕЙМВОРКІВ ДЛЯ РОБОТИ З ВЕЛИКИМИ ДАНИМИ ПРИ ВИРІШЕННІ ЗАДАЧ NLP

Abstract

*Iryna P. Drobiazko, senior lecturer; Myhaylo P. Turkin, student
Analysis of Big Data frameworks for NLP*

In this paper modern Big Data frameworks for processing data on a cluster are analyzed. A comparison in terms of performance, accessibility and usability for textual data analysis is made. Optimal frameworks were chosen. The choice is justified.

Вступ

Аналіз даних в сфері роздрібної торгівлі є однією з важливих складових успішного функціонування бізнесу. Зокрема, під час аналізу даних виникають задачі NLP, а саме: обробки відгуків клієнтів, пошуку схожих товарів у конкурентів за їх описом, встановлення залежностей між товарами (сабститутивність та комплементарність). Це сприяє кращому оцінюванню як власного асортименту, так і позицій на ринку [1]. Проте обсяги інформації, що потребують обробки, є дуже великими та постійно зростаючими. Це зумовлює необхідність використання для обробки даних програмного забезпечення, що працює паралельно на кластері комп'ютерних серверів, часто в хмарному середовищі. Існує багато фреймворків для розподіленої обробки даних на кластері. Вони відрізняються підходами до передачі даних та оптимізації алгоритмів обробки.

Тому при рішенні задач аналізу текстових даних виникає необхідність вибору фреймворку, який забезпечуватиме високу продуктивність та надаватиме інструменти для роботи з задачами NLP.

Постановка задачі

Задача полягає в аналізі існуючих рішень з метою визначення оптимального фреймворку з точки зору продуктивності, доступності та

зручності для аналізу текстових даних на прикладі задачі класифікації товарів за їх описом.

Термінологія

NLP (від англ. Natural Language Processing) – обробка природної мови

UDF (від англ. user-defined functions) – функції, що визначаються користувачем

Датафрейм (від англ. DataFrame) - двовимірна (рядки і стовпці) структура даних

Огляд фреймворків для задач NLP на кластері

Apache Hadoop є фреймворком на базі Java з відкритим вихідним кодом, який надає засоби пакетної обробки та зберігання даних на групі апаратних машин, об'єднаних у кластери. Hadoop створений для надійних, масштабованих та розподілених обчислень [3]. В той же час, він може використовуватися як файлове сховище загального призначення. Цей фреймворк здатний зберігати і обробляти петабайти інформації.

Проте, він не має інструментів для роботи з текстом. Всі алгоритми мають бути реалізовані вручну з використанням парадигми MapReduce. Також не підходить для потокової обробки та обробки в реальному часі. Тому, не зважаючи на високу відмовостійкість та масштабованість, Hadoop не є зручним інструментом для аналізу текстових даних. Проте, його можна використати для зберігання даних.

Apache Spark – це фреймворк пакетної обробки з покращеною обробкою потоків даних. Завдяки виконанню операцій в пам'яті (на відміну від Hadoop), а також наявності засобів оптимізації порядку обчислень, він є надзвичайно швидкою кластерною обчислювальною системою.

Spark може працювати самостійно, підтримуючи роботу кластеру, або забезпечити послідовну інтеграцію з Hadoop. Фреймворк підтримує такі популярні мови програмування, як Python, R, Java та Scala [2].

Існує дуже популярна бібліотека на базі Apache Spark - Spark NLP [6]. Вона надає широкі можливості роботи з текстом, зокрема, великий вибір сучасних попередньо натренованих моделей для обробки тексту. Також у Spark з'явилися UDF (user-defined functions), які дозволяють застосовувати власні функції до кожної групи даних. Зокрема, можна використовувати сторонні бібліотеки Python. Це дозволяє спростити та розширити можливості обробки текстових даних. Проте, за використання UDF доведеться платити продуктивністю [7].

Перевагами Spark є швидкодія, розширений набір операцій для аналізу даних порівняно із Hadoop, наявність додаткових пакетів з реалізацією алгоритмів машинного навчання, роботи із графами. Також, це достатньо поширена та стабільна технологія, тому багато компаній в сфері обробки даних та розподілених обчислень підтримують її та забезпечують розвиток.

Spark має окремі недоліки, зокрема складність опанування, налаштування та впровадження, оскільки належне обслуговування вимагає розуміння обчислювальних парадигм та внутрішньої роботи Spark.

Apache Flink – це фреймворк з відкритим вихідним кодом, який підходить як для пакетної, так і для потокової обробки даних. Він краще придатний для кластерних середовищ. Фреймворк засновано на концепції потоків. Його називають “4G для великих даних”. Він працює набагато швидше, ніж Hadoop [4].

Основна його перевага полягає в обробці кожної події в режимі реального часу і забезпеченні невеликої затримки та високої пропускну здатності. В той же час Spark, використовуючи мікро-пакети, може забезпечити обробку лише в наближеному до реального часу режимі.

Ця технологія може бути оптимальним вибором для online learning, коли дані оновлюються дуже швидко і система повинна одразу реагувати на ці зміни. Наприклад, при рішенні типової задачі для онлайн роздрібних торговців або маркетплейсів, де асортимент та ціни конкурентів змінюються часто, а потік клієнтів дуже великий. Проте, у загальному випадку, ця технологія не має переваг над Apache Spark при обробці тексту.

Dask – фреймворк для паралельних обчислень з відкритим вихідним кодом, яка була випущена в 2015 році, тому є відносно новою в порівнянні зі Spark. Також, на відміну від Spark, одним з оригінальних принципів проектування, прийнятих при розробці Dask, було зробити бібліотеку максимально зручною для розробників на Pandas в задачах аналізу даних [5]. Зважаючи на це, додано підтримку паралельного виконання операцій над Pandas DataFrames. Також, є можливість запускати на кластері алгоритми популярної бібліотеки для машинного навчання scikit-learn. Dask дозволяє зручно працювати в багатомашинному кластері з обсягами даних в декілька терабайт. Dask не має бібліотек, призначених для роботи з текстом, проте має вбудовані бібліотеки для задач класифікації та кластеризації. Також він може працювати як обгортка з функціями користувача і виконувати їх паралельно на всіх машинах кластера (подібно до Apache Spark UDF).

Переваги Dask полягають в тому, що його легко налаштовувати та масштабувати. Вбудована підтримка Pandas DataFrames спрощує роботу для

спеціалістів з досвідом роботи з Pandas. Проте, комерційна підтримка Dask значно менша, ніж у Spark.

Ray – ще один проєкт, метою якого є спрощення розподілених обчислень. Ray складається з двох основних компонентів: Ray Core, який є фреймворком розподілених обчислень, та Ray Ecosystem, який фактично є низкою бібліотек для конкретних завдань, які постачаються разом з Ray [5].

Ray схожий на Dask тим, що дозволяє користувачеві використовувати код на Python, для паралельної обробки даних на всіх машинах кластера. Однак, на відміну від Dask, Ray не намагається імітувати API Pandas. Він лише надає інтерфейс для виконання коду на Python паралельно. Це робить його універсальним фреймворком, який можна використовувати для створення та запуску розподілених додатків будь-якого типу. З іншого боку, це призводить до відсутності зручних інструментів для задач NLP та вбудованих примітивів для розбиття даних на пакети.

Таблиця 1

Результати аналізу фреймворків

Фрейм-ворки	Характеристики фреймворків		
	Простота налаштування та використання	Наявність бібліотек для NLP	Сфера застосування
Hadoop	Складно налаштувати, парадигма MapReduce	Немає	Загальна
Spark	Складно налаштувати, є реалізація датафреймів	Spark NLP або використовуючи UDF	Загальна
Flink	Складно налаштувати, є реалізація датафреймів	Використовуючи UDF	Робота з потоками даних, IoT
Dask	Дуже легко налаштувати кластер через SSH, Pandas-like датафрейми	Використовуючи UDF	Data Science
Ray	Легко налаштувати кластер вручну, використовується як обгортка	Інтеграція з іншими бібліотеками	Розподілені обчислення

Проте Ray інтегрується з бібліотеками, які мають інструменти для роботи з текстом (HuggingFace, SpaCy), що дозволяє використовувати його для прискорення тренування NLP моделей.

Результати порівняльного аналізу фреймворків наведено у таблиці 1.

Висновки

В роботі проведено порівняльний аналіз відомих сучасних фреймворків для обробки великих обсягів даних на кластері, який показав, що для аналізу текстових даних оптимальним вибором є фреймворки Apache Spark та Dask.

Apache Spark - найбільш поширений на сьогодні фреймворк, що має високу надійність та добре інтегрується з багатьма системами. Бібліотека Spark NLP є однією із найпоширеніших NLP бібліотек, що використовується на підприємствах. Вибір Spark є обґрунтованим для великих обсягів даних і масштабних проєктів.

Для швидкого масштабування існуючих рішень доцільним є вибір фреймворку Dask, оскільки він легко налаштовується та може бути використаний як обгортка до існуючого Python коду. Крім того, реалізація датафреймів буде звичною для фахівців, що працювали з Pandas.

Література

1. Плуа Katsov, The theory and practice of enterprise AI: Recipes and Reference Implementations for Marketing, Supply Chain, and Production Operations (2022) 213–230
2. About Spark - Databricks: URL: <https://www.databricks.com/spark/about>
3. Hadoop: URL: <https://www.databricks.com/glossary/hadoop>
4. Stream Processing Platforms: URL: <https://scramjet.org/stream-processing-platforms>
5. Spark, Dask, and Ray: Choosing the Right Framework: URL: <https://www.dominodatalab.com/blog/spark-dask-ray-choosing-the-right-framework>
6. State of the Art Natural Language Processing: URL: <https://nlp.johnsnowlabs.com/>
7. Jules S. Damji, Brooke Wenig, Tathagata Das, and Denny Lee, Learning Spark Lightning-Fast Data Analytics (2020) 336-337

УДК 004.35

**Студент Єршихін С. В., к.т.н., ст. викладач Коляда К. В.,
асистент Радченко К. О.**

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

КОМПОНЕНТИ АПАРАТНОГО МОДУЛЯ ДЛЯ КОМПЛЕКСНИХ СИСТЕМ МОНІТОРИНГУ ЯКОСТІ ПОВІТРЯ

Abstract

**Serhii Yershykhin, student; Kostiantyn Koliada, senior lecturer, PhD;
Kostiantyn Radchenko, assistant**

Components of the hardware module for complex air quality monitoring systems

In this paper an overview of the budget options for sensors used in hardware modules of complex air quality monitoring systems is presented. The selection of necessary sensors is proposed. The comparative analysis of different options is fulfilled. The way for sensors' integration into systems is proposed as well.

Вступ

Проблема забрудненого повітря є однією з найбільш нагальних у сучасному світі. Її вирішення починається з постійного моніторингу якості повітря. Для цього необхідно мати достатню кількість відповідних апаратних засобів, оснащених датчиками, що аналізують різні показники та складові повітря. На жаль, у сучасному світі більшість таких апаратних засобів є дорогими та використовуються локально лише великими компаніями у великих містах. Для покращення моніторингу повітря необхідно знайти бюджетні альтернативи апаратних компонентів для збільшення покриття території та покращення якості моніторингу повітря.

Постановка задачі

Задача полягає в пошуку бюджетних альтернатив компонентам апаратного модуля систем моніторингу якості повітря для зменшення витрат на модуль при збереженні якості вимірювань, що дозволить збільшити бази

даних про стан та якість повітря за рахунок збільшення кількості систем, що виробляються.

Термінологія

Моніторинг – періодичні безперервні, довгострокові спостереження, оцінки і прогнозу змін стану явищ та процесів.

Датчик, давач, сенсор – вимірювальний пристрій у вигляді конструктивної сукупності одного або декількох вимірювальних перетворювачів величини, що вимірюється і контролюється, та котрий виробляє вихідний сигнал, зручний для дистанційного передавання, зберігання та використання у системах керування і має нормовані метрологічні характеристики.

Огляд бюджетних рішень

Перед тим як розглянути варіанти бюджетних компонентів апаратного модуля комплексних систем моніторингу якості повітря необхідно спочатку визначитися зі списком показників [1], що будуть вимірюватися в запропонованій системі, а також вимоги до датчиків.

Серед показників, що будуть вимірюватися, найбільш важливими є [1]:

- озон O_3
- діоксид азоту NO_2
- діоксид сірки SO_2
- чадний газ CO
- дрібні частинки PM_{10} та $PM_{2.5}$
- бутан C_4H_{10}
- метан CH_4
- пропан C_3H_8

Саме вимірювання цих показників необхідно взяти за основу роботи апаратного модуля будь-якої системи моніторингу якості повітря.

Серед основних вимог до датчиків можна виділити наступні:

- точність вимірювань
- бюджетність
- малогабаритність
- багатофункціональність (наприклад, можливість використання одного датчика для декількох вимірювань)
- простота підключення

Сформувавши список необхідних показників та вимоги, можна обирати сенсори, що будуть відповідати заданим вимогам. В нашому випадку було обрано п'ять високоефективних датчиків, які задовольняють усі вимоги та можуть як найкраще проводити моніторинг якості повітря, залишаючись бюджетними та компактними:

- Shinyei PPD42 для розпізнавання дрібнодисперсних часточок в повітрі – PM2.5 та PM10 (пил)
- Газовий сенсор MQ2 для виявлення пропану, метану та бутану
- Газовий сенсор MQ9 для розпізнавання в повітрі чадного газу (CO)
- Газовий сенсор MiCS-2714 для моніторингу рівня NO2 (діоксид азоту)
- Газовий сенсор MiSC-2614 для виявлення озону в повітрі

Розглянемо кожен датчик більш детально.

Перший сенсор Shinyei PPD42 японського виробництва, працює на методі розсіювання світла та постійно виявляє тверді частинки, що потрапляють у повітря. Вихід імпульсу, що відповідає концентрації частинок на одиницю об'єму отримується оригінальним методом детектування, який заснований на принципі розсіяного світла, подібному до лічильника частинок. Резистор генерує вихідний струм й частинки, що потрапили в зону детектору, та проходять через точку зондування, тим самим розсіюють світло, а рецептор отримує дане світло через лінзу й перетворює його на імпульсний сигнал [2]. Імпульс на одиницю часу є пропорційний концентрації твердих частинок, що концентруються в атмосферному повітрі. Через непросту, але точну технологію, датчики мають доволі високу ціну, але їх присутність та роль в даній системі моніторингу є надзвичайно важливою. Розмір частинок, що можуть бути розпізнані Shinyei PPD42 становить понад $1.0\mu\text{m}$, що є доволі точним та чітким показником при розпізнаванні PM2.5 та PM10. Напруга живлення датчика 5 В постійного струму та діапазон робочої температури та вологості до 45°C та менше 95% відповідно. Вивід даних відбувається за допомогою широтно-імпульсної модуляції.

Наступний датчик газу MQ2-Gas є датчиком широкого спектру газів, тому може виявити виток пропану, бутану, метану або водню, також диму та різноманітних випарень [3]. Даний сенсор побудований на базі напівпровідникового газоаналізатора MQ2. Давач має два виходи: аналоговий та цифровий, а також заземлення. Принцип роботи полягає в тому, що на логічний вихід датчик видає аналоговий сигнал, пропорційний вмісту газів в навколишньому середовищі, а напруга на аналоговому виході змінюється в залежності від концентрації різних домішок в атмосфері. Чутливість даного сенсора налаштовується потенціометром та становить близько 10000 ppm, а

діапазон робочої температури та вологості до 22°C та менше 70% відповідно. Значення опору MQ2 є різницею до різних видів та концентрацій газів у повітрі, тому при точному вимірюванні необхідно визначати правильну точку сигналізації. Ще однією перевагою MQ2 є широка доступність і популярність на вітчизняному ринку, а також якість в передачі показників за невисоку ціну.

Газовий датчик MQ9-Gas має надзвичайно високу чутливість до чадного газу (CO). Чутливим матеріалом даного датчика є SnO₂ (діоксид олова), який існує в повітрі в невеликій кількості [4]. Вихідним результатом є аналоговий сигнал, пропорційний вмісту вуглеводних газів. Регулювання напруги відбувається за допомогою ШІМ-сигналу, а управління через транзистор. Провідність MQ9 стає вищою з підвищенням концентрації шкідливих газів у повітрі. Чутливість MQ9 до CO становить до 1000 ppm, що є доволі високим показником. Датчик можна використовувати на вулиці, адже він працює навіть при мінусовій температурі та з вологістю повітря до 70%. Основна перевага – це висока точність до розпізнавання CO, а також довготривала цілісність при недорогій ціні.

Датчик, що допомагає виявити в повітрі NO₂, має назву MiCS-2714 [5]. Даний сенсор працює в умовах від -30°C до 85°C та до 60% відносної вологості. Коефіцієнт чутливості становить відношення 0,25 ppm NO₂ на концентрацію всього повітря. Перевагою даного сенсора є швидкий відгук, який відбивається за короткий проміжок часу (менше 10с), малогабаритність та точність моніторингу заданих газів навіть у важкодоступних місцях при малій концентрації. Конструкція сенсора складається з переробленої діафрагми та вбудованого резистору, який виступає чутливим елементом. Концентрація кількості газів у повітрі співвідносна опору, саме так і виконується вимір.

Останнім датчиком є MiCS-2614 O₃ Sensor, що визначає кількість озону в повітрі. Датчики, які здатні аналізувати присутність O₃ зазвичай дороговартісні, але надзвичайно точні. Для систем моніторингу повітря підходить саме MiCS-2614 O₃ Sensor, якісні та точні виміри якого мають перевагу над ціною політикою. Датчик видає аналогову напругу, яка швидко перетворюється вбудованим 12-розрядним перетворювачем АЦП MCP3201 [6]. Виявлення концентрації озону відбувається шляхом вимірювання чутливості резистору під час активної роботи датчика та становить майже 1000 ppb.

Усі запропоновані датчики є всепогодними, малогабаритними та широкодоступними, що дозволяє використовувати їх у компактних апаратних модулях для моніторингу якості повітря як на вулиці, так і в приміщеннях.

Усі вони також мають просту схему підключення, що робить їх сумісними з багатьма варіантами сучасних модулів.

Висновки

При порівнянні витрат на закупівлю запропонованих датчиків та наявних на ринку рішень, можна побачити суттєвий вигреш у затратах на запропоноване рішення при сталій якості вимірювань показників забрудненості повітря, що дає змогу побудувати апаратний модуль для комплексної системи моніторинг якості повітря із меншим бюджетом та без втрат в якості та точності вимірювань.

Література

1. Toxic Risk / Characteristics of toxic gases [Електронний ресурс] / Режим доступу: <https://www.crowcon.com/resources/talking-gas/what-is-gas/toxic-risk/characteristics-of-toxic-gases/> – Дата доступу: звернення 15.10.2022.
2. Датчик твердих частинок Shinyei PPD42 [Електронний ресурс] / Режим доступу: <https://www.shinyei.co.jp/stc/eng/products/optical/ppd42nj.html> – Дата доступу: звернення 05.10.2022.
3. Сенсор газу MQ2-Gas [Електронний ресурс] / Режим доступу: <https://static.chipdip.ru/lib/184/DOC001184457.pdf> – Дата доступу: звернення 05.10.2022.
4. Сенсор газу MQ9-Gas [Електронний ресурс] / Режим доступу: https://www.smart-prototyping.com/image/data/2_components/sensors/100996%20MQ-9/MQ9.pdf – Дата доступу: звернення 07.10.2022.
5. Сенсор газу MiCS-2714 [Електронний ресурс] / Режим доступу: https://eu.mouser.com/datasheet/2/18/1107_Datasheet-MiCS-2714-1144844.pdf – Дата доступу: звернення 07.10.2022.
6. Датчик озону MICS-2614 [Електронний ресурс] / Режим доступу: <https://datasheet.octopart.com/MICS-2614-E2V-datasheet-8615830.pdf> – Дата доступу: звернення 08.10.2022.

УДК 519.876.5

Д.т.н, професор Зайцев В.Г., студент Стабровський В.К.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

МОДЕЛЮВАННЯ ВИКОНАННЯ ЗАДАЧІ В СИСТЕМІ РЕАЛЬНОГО ЧАСУ

Abstract

Volodymyr Zaitsev, prof., PhD; Vladyslav Stabrovky, student
Simulation of task performance in a real-time system

This work concerns the creation of a task performance model. Petri nets have been studied as a mathematical model for creating a model of task performance. The paper presents a model that allows simulating the performance of one task.

Вступ

Задача моделювання системи реального часу є важливою складовою при розробці таких систем. Це дозволяє перевірити працездатність системи та алгоритмів планувальника задач, обрати найкращі варіанти алгоритмів та виправити виявлені помилки. Однією з вимог при моделюванні системи реального часу, є створення моделі виконання задачі.

Постановка задачі

Задача полягає в розробці моделі мережі Петрі, яка моделює процес виконання однієї задачі, та яку можна буде використовувати для моделювання роботи множини задач, де планувальник задач буде обирати поточну задачу, яка виконується.

Теоретичні відомості

Система реального часу – це система для якої час виконання задач є дуже важливою характеристикою, та ефективність роботи забезпечується алгоритмом планування задач. Тому важливою частиною при розробці системи реального часу є розробка алгоритму планувальника задач. Для перевірки працездатності та обираючись найкращого алгоритму виконується моделювання роботи системи реального часу.

Задача це деякий набір операцій, які виконують логічно завершену функцію. Модель виконання задачі полягає в тому, що задача має час виконання в деяких умовних одиницях.

Мережа Петрі це математичний об'єкт, який дозволяє змоделювати роботу динамічної дискретної системи. Модель мережі складається з позицій, переходів, маркерів, які знаходяться в позиціях, дуг, що з'єднують позиції та переходи. Якщо в кожній позиції, що з'єднані з певним переходом дугами, достатня кількість маркерів, то перехід спрацює. При цьому в тих позиціях зникає певна кількість маркерів, що залежить від кількості дуг до переходу, та з'являються в кожній позиції, з якими дугами з'єднаний перехід, кількості маркерів дорівнює кількості дуг. На рис. 1 зображено приклад мережі Петрі, де кола (P1, P2, P3, P4) це позиції, прямокутні риси (T1, T2) це переходи, чорні точки це маркери, стрілками помічені дуги.

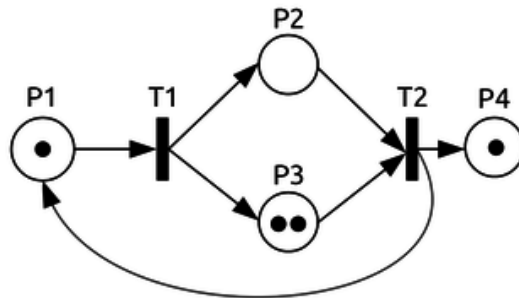


Рис. 1. Мережа Петрі

Модель роботи задачі

Модель виконання задачі повинна містити інформацію про час, який необхідний для виконання задачі, стани задачі, такі як задача готова до виконання, виконується, закінчила виконуватися. Мережа Петрі, яка дозволяє змоделювати виконання задачі, наведена на рис. 2.

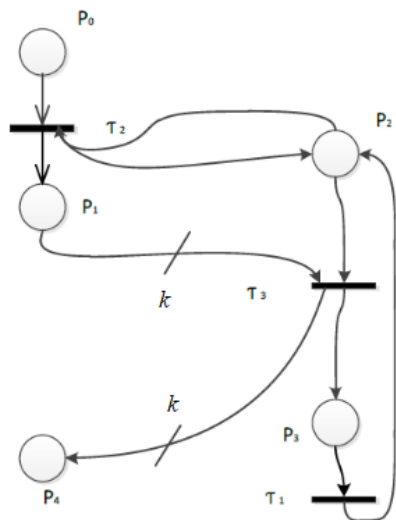


Рис. 2. Мережа Петрі, яка моделює виконання задачі

Параметр k містить значення кількості тактів, яких необхідно для виконання задачі. Спрацювання переходу T1 показує, що дана задача

почала виконуватися. Перехід T2 спрацьовує при кожному такті роботи системи. Перехід T3 спрацьовує коли задача виконала всі умовні операції. Позиція P0 містить маркери, кількість яких вказує на час, що має задача на виконання. P1 показує проміжний результат виконання задачі. Якщо P2 містить маркер, то ця задача почала виконуватися. Якщо позиція P3 містить маркер, то дана задача готова до виконання, або вже завершила своє виконання. Якщо P4 містить маркери, то дана задача завершила роботу.

Дану мережу Петрі можна представити у вигляді матриць M^- та M^+ розмірами $m \times n$, де m – кількість переходів, n – кількість позицій. Матриця M^- показує скільки та в яких позиціях зникне маркерів при спрацюванні кожного переходу. M^+ показує скільки та в яких позиціях з'явиться маркерів при спрацюванні переходів. Для даної мережі Петрі значення матриць M^- та M^+ наведені у табл. 1 та табл. 2 відповідно.

Таблиця 1

Матриця M^-

	P0	P1	P2	P3	P4
T1	0	0	0	1	0
T2	1	0	1	0	0
T3	0	k	1	0	0

Таблиця 2

Матриця M^+

	P0	P1	P2	P3	P4
T1	0	0	1	0	0
T2	0	1	1	0	0
T3	0	0	0	1	k

Загальну матрицю M можна отримати за формулою:

$$M = M^+ - M^- \quad (1)$$

Матриця отримана формулою (1) наведена у табл. 3, та позначає зміни кількості маркерів в позиціях при спрацюванні переходів.

Таблиця 3

Матриця M

	P0	P1	P2	P3	P4
T1	0	0	1	-1	0
T2	-1	1	0	0	0
T3	0	$-k$	-1	1	k

При кожному такті мережі виконується спроба запуснути послідовність переходів T1, T2, T3, що може призвести до зміни маркування, яке обчислюється за формулою:

$$\mu_{i+1} = \mu_i + V_j \times M, \quad (2)$$

де μ - значення маркування, V_j – вектор переходу, в якому усі значення дорівнюють 0 окрім j , яке дорівнює 1, j – номер переходу, який намагається спрацювати. Якщо нове маркування містить від’ємні значення, то перехід не спрацювала і маркування слід залишити незмінним.

У табл. 4 наведено приклад роботи моделі задачі, де за початкове маркування взято $\mu = [k\ 0\ 0\ 1\ 0]$.

Таблиця 4

Робота моделі задачі

Номер такту	Перехід	Початкове маркування	Результат запуску	Спрацювання переходу	Нове маркування
1	T1	$k\ 0\ 0\ 1\ 0$	$k\ 0\ 1\ 0\ 0$	+	$k\ 0\ 1\ 0\ 0$
	T2	$k\ 0\ 1\ 0\ 0$	$k-1\ 1\ 1\ 0\ 0$	+	$k-1\ 1\ 1\ 0\ 0$
	T3	$k-1\ 1\ 1\ 0\ 0$	$k-1\ -k+1\ 0\ 1$ k	-	$k-1\ 1\ 1\ 0\ 0$
2	T1	$k-1\ 1\ 1\ 0\ 0$	$k-1\ 1\ 2\ -1\ 0$	-	$k-1\ 1\ 1\ 0\ 0$
	T2	$k-1\ 1\ 1\ 0\ 0$	$k-2\ 2\ 1\ 0\ 0$	+	$k-2\ 2\ 1\ 0\ 0$
	T3	$k-2\ 2\ 1\ 0\ 0$	$k-2\ -k+2\ 0\ 1$ k	-	$k-2\ 2\ 1\ 0\ 0$
...					
$k-1$	T1	$2\ k-2\ 1\ 0\ 0$	$2\ k-2\ 2\ -1\ 0$	-	$2\ k-2\ 1\ 0\ 0$
	T2	$2\ k-2\ 1\ 0\ 0$	$1\ k-1\ 1\ 0\ 0$	+	$1\ k-1\ 1\ 0\ 0$
	T3	$1\ k-1\ 1\ 0\ 0$	$1\ -1\ 0\ 1\ k$	-	$1\ k-1\ 1\ 0\ 0$
k	T1	$1\ k-1\ 1\ 0\ 0$	$1\ k-1\ 2\ -1\ 0$	-	$1\ k-1\ 1\ 0\ 0$
	T2	$1\ k-1\ 1\ 0\ 0$	$0\ k\ 1\ 0\ 0$	+	$0\ k\ 1\ 0\ 0$
	T3	$0\ k\ 1\ 0\ 0$	$0\ 0\ 0\ 1\ k$	+	$0\ 0\ 0\ 1\ k$

Висновок

Отримана модель дозволяє змоделювати виконання однієї задачі. В подальшому на основі цієї моделі створюється модель виконання декількох задач, яка вже використовується в моделюванні системи реального часу, де планувальник задач обирає в якій саме задачі виконати один такт.

Література

1. Сулейманова А.М. Системы реального времени: учебное пособие/ Уфимск. гос. авиац. техн. ун-т.– Уфа, 2004.– 292 с.
2. Питерсон Дж. Теория сетей Петри и моделирование систем. – М.: Мир, 1984. – 264 с.

УДК 519.688

К.т.н., доцент Клятченко Я.М., студент Шевченко Д.С.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

ПОРІВНЯЛЬНИЙ АНАЛІЗ ВЕРСІЙ WORLD WIDE WEB

Abstract

*Yaroslav M. Klyatchenko, assoc. prof., PhD; Dmytro Shevchenko, student
Comparative analysis of World Wide Web versions*

This work is devoted to versions of the Internet. A comparative analysis of three versions of the World Wide Web was conducted. The best approach, concepts and technologies for creating modern web applications are defined.

Вступ

В сучасному світі інтернет є невід'ємною частиною життя кожної людини. Сьогодні неможливо уявити будь-яку сферу бізнесу, навчання, надання послуг, або реклами без використання веб-ресурсів таких, як веб-додатки або веб-сторінки, соціальні мережі, месенджери тощо. Але інтернет не був таким завжди, і значно змінився з часів створення першої веб-сторінки. З того часу минуло тридцять два роки, 20 грудня 1990 року Тім Бернерс-Лі, британський вчений з Європейської організації з ядерних досліджень (CERN), запустив перший у світі веб-сайт. Він був присвячений опису нової технології глобального обміну інформацією - "World Wide Web" або "Всесвітня павутина" [1].

Постановка задачі

Метою даної роботи є проведення порівняльного аналізу версій World Wide Web, зокрема Web 3.0, щоб визначити її переваги та з'ясувати можливості використання для створення сучасних веб-додатків.

Термінологія

World Wide Web (WWW) — це інформаційна система, що забезпечує доступ до документів та інших веб-ресурсів через Інтернет.

Web 1.0, 2.0, 3.0 - версії **World Wide Web**, що представляє собою сукупність різних підходів та технологій.

Аналіз основних принципів та концепцій

Web 1.0 відноситься до першого етапу еволюції World Wide Web. Це були веб-сторінки, що містили в основному статичний контент і розміщувались на веб-серверах провайдерів або безкоштовних веб-хостингах. Епоха Web 1.0 тривала приблизно з 1990 по 2000 рік [2].

Web 2.0 - це вдосконалена версія Web 1.0. Зміни стосувалися не модифікації будь-якої технічної специфікації, а зміни способу розробки та використання веб-сторінок. Взаємодія між користувачами та діалог у віртуальній спільноті стали основним принципом та покращенням нового покоління веб-ресурсів. Це поточна версія Інтернету, з якою знайомі сучасні користувачі. Web 2.0 є стандартом за замовчуванням. Активний розвиток цієї версії тривав з 2000 по 2010 рік [2]. Якщо перша версія являє собою статичну мережу з лінійною інформацією, то друга версія - це динамічна мережа, що складається із нелінійної інформації, в якій не має обмежень, і крім цього її можна редагувати. Основними проблемами даної мережі є компромісна безпека та конфіденційність, цензурування даних та централізація контенту.

Web 3.0 представляє собою нову ітерацію або фазу еволюції Інтернету. Розвиток Web 3.0 розпочався в 2010 році і триває зараз [2]. Web 3.0 побудовано на основних концепціях децентралізації, відкритості та більшої зручності для користувачів.

Порівнюючи всі 3 версії, можна сказати, що Web 1.0 - мережа лише для читання, Web 2.0 - для читання та запису, а Web 3.0 - для читання, запису та виконання. Для третьої версії всі дані є децентралізованими, тобто копії даних зберігаються на різних серверах, і ніхто не має контролю над ними. Децентралізовані додатки ніколи не припиняють роботу завдяки відсутності центрального сервера, який міг би зупинитися. Дані в них розподілені між вузлами, при цьому всі вузли діють незалежно, і якщо один з них зупиниться, інші продовжать роботу в мережі [3]. Web 3.0 - являє собою інфраструктуру із декількох основних технологій: Blockchain, Semantic Web, Machine Learning та Internet of things. Кожна із цих технологій є невід'ємною частиною мережі зі своєю роллю в екосистемі, яка дозволяє створювати системи, додатки, сервери нового покоління.

Blockchain - це технологія розподіленої бази даних. Блокчейн-інформація записана в ланцюг блоків, які підключені в строго визначеному порядку. Системи, засновані на Blockchain, здатні працювати без централізованого управління та довіри між учасниками [3]. Рішення в такій системі приймаються не центральними вузлами, а шляхом голосування всіх учасників системи. Для створення та підтримання блокчейну не потрібні

потужній сервери, достатньо створити мережу із звичайних комп'ютерів. Поведінка учасників регулюється консенсусним протоколом — набором правил для створення та запису блоків у реєстрі системи. Розподілений реєстр є основою децентралізованої веб-інфраструктури, де веб-програми можуть обмінюватися інформацією напряду та з усіма учасниками на рівних умовах. Завдяки децентралізації можна вирішити проблеми непрозорості веб-сервісів, цензури у мережі та конфіденційності особистих даних, що є надважливою проблемою в поточному Web 2.0. Передача даних в блокчейн-мережі проходить миттєво, тому головна сфера його використання це фінанси. Криптовалюта, яка доступна вже зараз, керується децентралізованою платіжною системою повністю автоматизовано, тобто без внутрішнього або зовнішнього адміністрування. Підробити або скасувати запис в такій системі неможливо, або дуже витратно. Всі транзакції виконуються анонімно, відомі тільки номери рахунків, а не особисті дані.

Semantic Web або семантична павутина - наступна концепція, що вирішує проблему перетворення всієї інформації в мережі в зрозумілу для машин. Для реалізації семантичної павутини, всій інформації в мережі необхідно присвоїти метадані, тобто інформацію про інформацію. Ключовий елемент в реалізації семантичної павутини - RDF (Resource Description Framework). Це модель опису інформації через спеціальні зрозумілі машині твердження - триплети, що складається з трьох частин: «суб'єкт», «предикат» та «об'єкт» [4]. За допомогою цього вирішується проблема опису будь-якого об'єкта в мережі: веб-додатку, зображення, аудіозапису, людини.

Machine Learning або Машинне навчання — це система методів, які використовують комп'ютерні алгоритми для вирішення проблем без прямих інструкцій. Алгоритм вчиться виконувати певне завдання. Він аналізує набір даних і самостійно виявляє шаблони, які потім використовує для виконання завдань [2]. Така концепція стає фундаментальною для Web 3.0, так як це вирішує проблему аналізу великої кількості даних, що присутня в мережі нового покоління.

Концепція **Internet of things або Інтернет речей** описує мережу багатьох пристроїв, які мають датчики та підключення до Інтернету, що взаємодіють один з одним і ними можна керувати віддалено [2]. Інтернет речей поєднує світ веб-додатків зі світом запрограмованих розумних фізичних об'єктів: автомобілів, будинків, телевізорів. Для Web 3.0 такі пристрої є не тільки ще одним засобом доступу до мережі, а й постійними джерелами даних про фізичний світ. За допомогою Інтернету речей в мережі накопичуються детальні дані, що можна використовувати в потрібній для

користувачів сфері, наприклад для пошуку найкращого маршруту, вирахування статистики, урбаністичних досліджень.

Додатки нового покоління становлять собою програмне забезпечення, яке не обов'язково використовує всі концепції Web 3.0, але мінімум одну із технологій. Основним прикладом є додатки для виконання операцій з криптовалютами, для аналізу даних за допомогою штучного інтелекту, голосові асистенти для виконання розумного пошуку або управління системою розумного дому, побудови 3D-моделей ландшафту тощо.

Висновки

Після виконаного аналізу можна зробити висновок, що використовувати концепції версій Web 1.0 та Web 2.0 недоцільно для створення сучасних веб-ресурсів, так як в них існує багато проблем, що є критичні для користувача. Доцільніше використовувати більш новітні підходи. Використання найсучасніших концепції та технології Web 3.0, вирішує проблеми безпечної авторизації та приватності, децентралізації додатків та онлайн-платформ, розумного пошуку та пристроїв, доступу до інформації. Прогрес робить перехід від поточного Web 2.0 до Web 3.0 можливим та неминучим. Створення децентралізованих веб-додатків, розумних пошуковиків та реалізація Інтернету речей, допоможе прискорити цей процес.

Література

1. Перший у світі веб-сайт. [Електронний ресурс] – Режим доступу : <https://babel.ua/texts/23307-28-rokiv-tomu-zapustili-pershiy-u-sviti-veb-sayt-mi-pokazali-yak-viglyadav-bi-thebabel-v-1990-roci>
2. What is Web 1.0, 2.0, and 3.0 and How They Compare. [Електронний ресурс] – Режим доступу : https://www.simplilearn.com/what-is-web-1-0-web-2-0-and-web-3-0-with-their-difference-article#what_is_web_30
3. Web 3.0 explained. [Електронний ресурс] – Режим доступу : <https://www.investopedia.com/web-20-web-30-5208698>
4. Transformation to Web 3.0. [Електронний ресурс] – Режим доступу: <https://hacken.io/insights/web-3-0-transformation-explained-cybersecurity-implications/>

УДК 004.051

К.т.н. доц. Клятченко Я.М., студент Шепель К.С.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

МОДИФІКОВАНИЙ АЛГОРИТМ SB-РЕНДЕРИНГУ

Abstract

Klyatchenko.Y.M.,doc., PhD; Kostyantyn Shepel, student
Modified SB-rendering algorithm

This paper concerns the task of modifying the old SB-rendering algorithm to simplify and increase the efficiency of use. The classical algorithm is studied and discussed. The modified SB-rendering algorithm with automated input of initial data is proposed. The comparative analysis of efficiency of both the classical and the modified algorithms is fulfilled. The ways for further research are proposed as well.

Вступ

Растрові графічні зображення широко використовуються у сфері дизайну. Кожен оброблюваний елемент на полотні являє собою одиницю візуальної комунікації з користувачем. Концепт-художники та сг-ілюстратори користуються елементами текстур, шаблонними та абстрактними зображеннями, набагато частіше ніж реалістичними. Стилізація зображення надає можливість дизайнеру під час редагування додавати інші компоненти. З цієї точки зору фотографічне зображення менш гнучке.

Існують алгоритми рендерингу з використанням відповідної стилістики, але більшість з них потребують значної взаємодії з користувачем. Для отримання результату stroke-based рендерингу [1] необхідно вручну вказувати положення накладаючих текстур та їх кількість. Окрім цього, в деяких випадках вхідний масив даних додатково заповнюється інтенсивністю кольору та орієнтацією кожного окремого мазка на полотні. Очевидно, що такий підхід з введенням багатьох значень достатньо складний для користувача, оскільки для кожного зображення потрібно задавати нові початкові дані. Через значний час обробки зображення процес динамічного (real-time) рендерингу неможливий. В даній статті пропонується модифікація існуючого алгоритму SB-рендерингу, яка дозволяє автоматизувати процес синтезу зображення та значно зменшити складність обробки з боку користувача.

Постановка задачі

Задача полягає в модифікації алгоритму SB-рендерингу з ручним введенням даних таким чином, щоб мінімізувати час обробки та коректно виконати рендеринг зображення, досягнувши умов відповідної стилістики.

Термінологія

SB-рендеринг – процес синтезу зображення за допомогою компонування рендер-міток [1].

Real-time рендеринг – процес синтезу декількох зображень в секунду [3].

Фільтр Гауса – техніка фільтрації за допомогою якої виконується розмиття зображення [4].

Фільтр Собела - дискретний диференціальний оператор за допомогою якого обчислюється наближене значення градієнта яскравості графічного зображення [2].

Опис алгоритму

Для виконання SB-рендерингу потрібно згенерувати текстури, наприклад мазки пензля, по всій площині оригінального зображення.

Позначення:

c_x, c_y – центр траєкторії по якій виконується замальовування.

l – довжина лінії

r – радіус мазку

$\alpha = \text{const}$ – орієнтація на площині

Z_i – лічильник генерації центрів траєкторії

Зазначимо, що з даним лічильником при $1.5 \leq i \leq 2.0$ буде виконане повне покриття всієї області.

Поточний колір C для мазку у відповідній точці (c_x, c_y) буде:

$$C(c_x, c_y) = a_1 + a_2 c_x + a_3 c_y + a_4 c_x c_y$$

де a_1, a_2, a_3, a_4 – коефіцієнти інтерполюючої функції [3].

Нанесення мазків виконується завдяки функції $\text{random}()$, для того щоб уникнути просторової когерентності.

Важливою складовою процесу є видалення частини лінії, для співпадання з початковою версією зображення. На відміну від більш простих технік, де область видалення мазка на полотні задається користувачем або допоміжним зображенням границь кольору, повний алгоритм, включаючи відсікання кольору, виконує наступне:

1. Використовуючи початкове зображення генерується матриця інтенсивності кольорів:

$$M(x, y) = \frac{30r + 59g + 11b}{10}, \text{ де } r, g, b \text{ – значення у точці } x, y, 0 \leq \{r, g, b\} \leq 255.$$

2. Для зменшення кінцевого шуму по точкам виконується розмиття (фільтр Гаусса [4]). Потрібно враховувати, що чим інтенсивніше розмиття тим менше дрібних деталей. Значення інтенсивності вказується за замовчуванням відповідно до критеріїв стилізації.

3. Підраховується значення градієнту (E_x, E_y) у кожній точці генерації траєкторії, після чого виконується фільтрація Собела [2].

$$\text{Sobel}(x,y) = H(E_x, E_y)$$

4. Мазок буде збільшуватися з центру (c_x, c_y) доти, доки не досягне кінцевого радіусу або границі зображення.

Якщо має місце:

$\text{Sobel}(x,y) < \text{Sobel}(x+\sin(\alpha), y+\cos(\alpha))$ (тобто по напрямку збільшення мазку), границя вважається знайденою.

5. Штрих зображується з кінцевими точками (x_1, y_1) та (x_2, y_2). Кольору штриха присвоюється колір початкового зображення в центрі штриха.

Висновки

При виконанні запропонованого алгоритму SB-рендерингу, на відміну від початкового, отримуємо значну автоматизацію обробки даних під час відтворення зображення. Початкова інформація про кількість мазків, напрямок замальовування, положення на площині, колір у точках, границі промальовування не потрібно змінювати кожен раз для нового зображення. Вони або автоматично підраховуються (положення, колір, межі), або встановлюються за замовченням (орієнтація, кількість рухів).

Оскільки відображення мазку виконується раніше перевірки границі промальовування, можна побачити, що відбиток на полотні не видаляються у кінцевих точках. Даний факт означає, що незважаючи на те, як щільно оточена траєкторія граничними пікселями, у відповідній точці, як мінімум, буде один відбиток. При збільшенні радіусу відбитку збільшується стилізація картини (через товщину пензля). При виконанні замальовування без рандомізації вибору (тобто за лічильником), зникає ефект замальовування “від руки”. Зменшуючи показник ядра Гауса близько до нуля, та збільшуючи радіус ми значно розмиваємо кінцеве зображення. Якщо ж не виконувати функцію видалення зайвих частин мазків, результат буде без чітко відмальованих границь силуетів відносно вхідного зображення.

У подальшому було б доцільно модифікувати даний алгоритм, автоматично прораховуючи напрямок руху пензля на площині, та зайнятися питанням оптимізації для виконання real-time рендерингу.

Література

1. Dunlavey M. R. Efficient polygon-filling algorithms for raster displays / M. R. Dunlavey// ACM Transaction Graphics 2(4)., 1983. - P. 264-275.
2. Haeberli P. E. Paint by numbers: Abstract image representations Computer Graphics / P. E. Haeberli,1990. - P. 207–214.
3. Jain A. K. Fundamentals of Digital Image Processing / A. K. Jain //UpperSaddleRiver, NJ: Prentice-Hall, 1989. – P. 173-176.
4. Jain A. K. Ramesh, Rangachar Kasturi, and Brian Schunck. Machine Vision. /A. K. Jain // Mc Graw-Hill, Inc. NewYork.,1995. – P. 17-34.

УДК 519.688

К.т.н., доцент Клятченко Я.М., студент Шоломіцький І.В.

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

МОДИФІКОВАНИЙ АЛГОРИТМ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ НА ЗОБРАЖЕННЯХ

Abstract

Yaroslav M. Kliatchenko, assoc. prof., PhD; Ivan Sholomitskyi, student
Modified CNN image recognition algorithm

This paper concerns the task of data augmentation. The modified algorithm with the data augmentation such as flipping and scaling is proposed. The comparative analysis of efficiency of both the standart and modified algorithms is fulfilled. The drawbacks and positive sides of the algorithm are discussed as well.

Вступ

Задача розпізнавання об'єктів є однією з базових в області комп'ютерного бачення. До неї зводиться чимало інших задач та вона широко використовується в комп'ютерній графіці.

Існує багато алгоритмів, які дозволяють обробити зображення за скінченну кількість кроків [1-2]. У даній статті пропонується модифікація існуючого алгоритму, яка використовує аугментацію [3] дозволяє підвищити результативність розпізнавання об'єктів нейронною мережею.

Постановка задачі

Задача полягає в модифікації існуючого алгоритму розпізнавання об'єктів таким чином, щоб збільшити точність результатів, використовуючи одну і ту ж саму базу даних зображень.

Термінологія

Аугментація – це додавання нових даних шляхом модифікації вже існуючих даних, на основі яких відбувається тренування нейронної мережі [3].

Задачею аугментації за заданим набором вхідних даних називається задача визначення та виконання потрібних модифікацій зображення, що

будуть використовуватися як дані, на основі яких відбувається тренування нейронної мережі [3].

Опис алгоритму

Усі згорткові нейронні мережі мають схожу архітектуру. Така мережа містить шар вхідних даних, декілька прихованих шарів та шар вихідних даних. В згорткових нейронних мережах кожне зображення подається як масив значень пікселя(рис 1). У використовуваній нейронній мережі, функцією активації є ReLU.

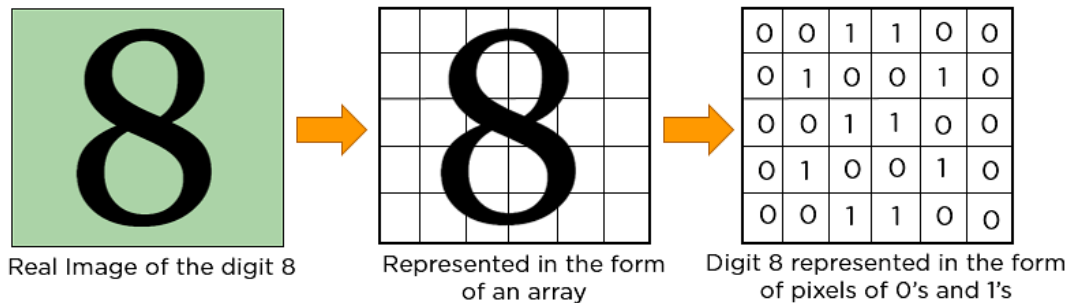


Рис 1. Подання зображення у нейронній мережі

Алгоритм згорткової нейронної мережі можна описати наступним чином[1]:

1. Зображення, у вигляді масиву пікселів передається до згорткового шару, що виконує операцію згортки. В результаті отримуємо згорткову карту.
2. Дана карта застосовується до функції активації ReLU, в результаті чого отримується зрізана карта ознак.
3. Зображення далі оброблюється прихованими шарами, які виконують функцію пулінгу та ідентифікують частини зображення. Приховані шари також виконують функції, на основі яких отримуються дані передбачення, що використовуються у вихідному шарі[4]
4. Вихідний шар отримує дані з прихованих шарів та надає передбачення нейронної мережі.

Проблема полягає у тому, що на пікселі зображення можуть впливати технічні фактори, такі як низька якість камери, що додає шум до зображення або освітлення. Для вирішення даної проблеми потрібно більше вхідних даних, на основі яких нейромережа буде ідентифікувати об'єкти. Такі зображення можуть не бути присутні у базі даних, з якою працює нейронна мережа. Тому, модифікація алгоритму полягає у додаванні аугментації вже існуючих зображень. Є декілька способів аугментації зображення:

- Поворот зображення
- Зміна розширення зображення
- Додавання шуму до зображення
- Віддзеркалення зображення

В запропонованій модифікації алгоритму, перед обробкою зображення віддзеркалюються, та змінюють розширення:

1. Зображення, у вигляді масиву пікселів передається до згорткового шару, що виконує операцію згортки. В результаті отримуємо згорткову карту.
2. Дана карта застосовується до функції активації ReLU, в результаті чого отримується зрізана карта ознак.
3. Зображення далі оброблюється прихованими шарами, які виконують функцію пулінгу та ідентифікують частини зображення. Приховані шари також виконують функції, на основі яких отримується дані передбачення, що використовуються у вихідному шарі.
 - 3.1 Зображення змінює розширення
 - 3.2 Зображення віддзеркалюється
 - 3.3 Дана модифікація зображення оброблюється, нейронна мережа тренується на основі модифікованих зображень
4. Вихідний шар отримує дані з прихованих шарів, що обробили оригінальне зображення та аугментовані варіанти та надає передбачення нейронної мережі.

Після отримання нових даних потрібно провести аналіз отриманих результатів. При використанні аугментації, збільшується точність(precision), що розраховується за формулою:

$$\frac{Truepositive}{Truepositive + Falsepositive} \quad (1)$$

Також збільшується згадування(recall), що розраховується за формулою:

$$\frac{Truepositive}{Truepositive + Falsenegative} \quad (2)$$

Висновки

При порівнянні запропонованого алгоритму зі звичайним алгоритмом отримуємо підвищення результатів коректної обробки

зображення нейронною мережею. Обидва алгоритми тестувалися на однакових наборах даних і на одному й тому самому комп'ютері.

Слід зазначити, що модифікований алгоритм вимагає більше ресурсів для проходження нейронною мережею. Це є основним недоліком даної модифікації. Збільшення варіантів зображення відповідно збільшує кількість часу, який потрібний нейронній мережі для обробки бази даних. А зміна розширення зображення вимагає більшої кількості згорток, що будуть проведені у прихованих шарах.

Даний алгоритм доцільно використовувати при потребі у більш точних результатах, за умови коли збільшення вимог до ресурсів не є проблематичним

Література

1. CNN Acchitectures [Електронний ресурс]. – 2022. – Режим доступу: <https://www.simplilearn.com/tutorials/deep-learning-tutorial/convolutional-neural-network>
2. Evaluating Deep Learning Models: The Confusion Matrix, Accuracy, Precision and Recall [Електронний ресурс]. – 2020. – Режим доступу: <https://blog.paperspace.com/deep-learning-metrics-precision-recall-accuracy/>
3. What is Data Augmentation in a CNN? Python Examples [Електронний ресурс]. – 2020. – Режим доступу: <https://nnart.org/what-is-data-augmentation-in-a-cnn/>
4. Prediction using Neural Networks [Електронний ресурс]. – 2020. – Режим доступу: <https://www.expressanalytics.com/blog/neural-networks-prediction/>

УДК 004.05

Аспірант Коваленко О.П., студент Галицький Д.

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

РОЗРАХУНОК КІЛЬКОСТІ ПРОЦЕСОРІВ У БАГАТОШИННІЙ ВІДМОВОСТІЙКІЙ БАГАТОПРОЦЕСОРНІЙ СИСТЕМІ

Abstract

Olena Kovalenko, student PhD, Danylo Halytskyi, student

Calculation of the number of processors in a multibus fault-tolerant multiprocessor system

This paper considers fault-tolerant multiprocessor systems that have several buses. The dependence of the maximum possible number of failures during which the system is still working on the system parameters is analyzed. Ensuring fault tolerance is one of the methods of increasing the reliability of control systems, and the following makes it possible to choose the system configuration at the development stage in such a way that the number of faults during which the system still works is maximal.

Вступ

Сучасні системи управління складними об'єктами мають підвищені вимоги в області організації роботи з високими показниками надійності та швидкодії. Зупинки у роботі складних систем, що викликані несправностями у системі управління, іноді призводять до серйозних наслідків, наприклад, таких, як затримка у виробництві або втрата прибутку [1]. На сьогодні інтенсивними темпами розвиваються складні системи, зокрема багатошинні багатопроцесорні відмовостійкі системи (БВС). Одним із основних факторів, який потрібно враховувати при розробці таких систем, є надійність. Існує декілька шляхів забезпечення потрібного рівня надійності. Одним з таких є підхід до оцінки надійності на основі степені відмовостійкості системи. Варто відмітити, що в інженерній практиці надійність систем управління визначається ймовірністю безвідмовної роботи за певний час. Надійність об'єкта часто порівнюють до недопустимості відмов в роботі системи управління. В даному випадку термін надійності можна розглядати як властивість об'єкта зберігати робоздатність протягом певного часу. Поняття надійності в загальноприйнятому розумінні – це комплексна властивість,

яка обумовлена типом об'єкта, умовами його експлуатації, а також має в собі такі поняття, як відмовостійкість, довгостроковість тощо [2].

Постановка задачі

Задача полягає в розрахуванні кількості відмов, при яких система залишається працездатною, в залежності від параметрів системи.

Термінологія

Відмовостійкістю називають властивість системи зберігати робоздатність при наявності відмов [3].

Опис методу

Позначення:

- N - загальна кількість процесорів
- k - кількість шин
- n_{imin} - мінімальна кількість процесорів на шині для забезпечення працездатності
- n_i - кількість процесорів на кожній шині
- n_{ij} - кількість процесорів між двома шинами
- m - число відмов, при яких система залишається працездатною

У даній роботі розглядається багатопроцесорна багатOSHинна відмовостійка система, що складається з k шин і N процесорів. Процесори даної БВС виходять з ладу з однаковою ймовірністю $p_i = p$.

Процесори на шинах розподіляються таким чином, що на кожній шині є мінімальне число процесорів для забезпечення робоздатності системи n_{imin} . При цьому на кожній шині є стале число процесорів, однакове для кожної шини n_i .

БВС виходить з ладу при появі $m + 1$ відмов. Розглянемо крайній випадок, коли система ще працює і виконує всі функції. Відповідно маємо наступне:

$$\begin{cases} m = n_i - n_{imin} \\ m = (k - 1) \cdot n_{ij} - 1 \end{cases} \rightarrow \begin{cases} n_i = m + n_{imin} \\ n_{ij} = \frac{m+1}{k-1} \end{cases} \quad (1)$$

В той же час загальна кількість процесорів БВС є сумою усіх процесорів на шинах і усіх процесорів, що зв'язують шини, і це дорівнює:

$$N = n_i \cdot k + C_k^2, \text{ де} \quad (2)$$

$$C_k^2 = \frac{k!}{2! \cdot (k-2)!} = \frac{k \cdot (k-1) \cdot (k-2)!}{2 \cdot (k-2)!} = \frac{k^2 - k}{2} \quad (3)$$

Тепер підставимо (1) та (3) у формулу (2).

Шляхом простих перетворень знайдено наступну формулу для числа відмов $m > 1$:

$$m = \frac{(2 \cdot N + 2 \cdot n_{imin} + 1) \cdot k - (2 \cdot n_{imin} + 1) \cdot k^2 - 2 \cdot N}{3 \cdot k^2 - 3 \cdot k} \quad (4)$$

Приклад.

Розглянемо систему з такими параметрами:

$$N = 62 \quad n_{imin} = 3$$

Підставимо початкові дані у формулу (4):

$$m = \frac{131 \cdot k - 7 \cdot k^2 - 124}{3 \cdot k^2 - 3 \cdot k}$$

Обчислимо число відмов в залежності від кількості шин:

$$k = 2 \rightarrow m = 18$$

$$k = 3 \rightarrow m = 11$$

$$k = 4 \rightarrow m = 8$$

$$k = 5 \rightarrow m = 5$$

$$k = 6 \rightarrow m = 3$$

Висновки

Задача розрахунку числа відмов, при яких система залишається працездатною, вирішена аналітичним шляхом. В майбутньому, використовуючи схожий підхід, можливо для аналізу задач вибору кількісного складу в залежності і від інших показників системи, а саме надійності та швидкодії.

Література

1. *Konesev S.G., Khazieva R.T.* Modern problems of science and education, 2015, no. 1-1. Available at: <http://www.science-education.ru/ru/article/view?id=17558> (accessed February 22, 2020).

2. *Lavrishcheva E.M., Pakulin N.V., Ryzhov A.G., Zelenov S.V.* Proceedings of ISP RAS, vol. 30, no. 3, 2018, pp. 99–120. DOI: 10.15514/ISPRAS-2018-30(3)-8.
3. *ДСТУ 2500-94.* Основні норми взаємозамінності. Єдина система допусків та посадок. Терміни та визначення. Позначення і загальні норми, за адресою http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=55979

УДК 004.04

К.т.н., доцент Марченко О.І., студент Єрмоленко Д.В.

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

**СПОСІБ ПОРІВНЯННЯ АБСТРАКТНИХ
СЕМАНТИЧНИХ ДЕРЕВ ПРОГРАМ НАПИСАНИХ
LISP-ПОДІБНИМИ МОВАМИ**

Abstract

Oleksandr I. Marchenko, assoc. prof., PhD; Denys Yermolenko, student
Technique for comparing abstract semantic trees of programs written in LISP-like languages

This paper concerns the task of comparing abstract semantic trees of programs written in LISP-like languages. The key features of abstract semantic trees of LISP-like languages are analyzed. An efficient technique of detecting the "same", "unique" and "moved" fragments in compared abstract semantic trees by using the modification of the LCS finding algorithm and by hashing trees is proposed. The ways for further research are offered as well.

Вступ

Задача порівняння абстрактних семантичних дерев програм є однією з тих задач, що дозволяє аналізувати зміни на структурному рівні в різних версіях кодової бази програми, виконувати пошук плагіату тощо. Наявні способи порівняння використовують різноманітні модифікації алгоритму пошуку TED (tree edit distance), бо класичний алгоритм пошуку TED не вміє знаходити переміщені фрагменти програми, а його операція визначення “перейменованого” вузла не враховує семантичного чи синтаксичного сенсу. Також треба зауважити, що пошук переміщених фрагментів у загальному випадку є NP-повною задачею [1], а тому наявні способи використовують різноманітні спрощення чи емпіричні підходи, але вони пристосовані та враховують специфіку саме тих мов програмування для яких були створені [2].

Оскільки популярність використання функціональних мов програмування зростає, то розробка таких особливих способів пошуку переміщених фрагментів програм для LISP-подібних мов є задачею актуальною.

Постановка задачі

Задача полягає в розробці способу порівняння абстрактних семантичних дерев з урахуванням особливостей LISP-синтаксису з метою виявлення переміщених фрагментів та, як результат, отримання дерева з маркуванням вузлів категорій “unique”, “same”, “moved” та списків відповідних цим категоріям елементів верхнього рівня без повного перебору усіх можливих варіацій піддерев.

Термінологія

Алгоритм пошуку LCS (longest common subsequence) – це алгоритм пошуку послідовності, що є підпослідовністю декількох послідовностей.

Алгоритм TED (tree edit distance) – алгоритм пошуку коефіцієнта подібності дерев. Під час його роботи також відбувається маркування вузлів дерев як «deleted», «inserted», «relabeled».

Абстрактне семантичне дерево AST (abstract semantic tree) – це дерево, яке формується з абстрактного синтаксичного дерева та вузли якого зберігають семантичну інформацію.

S-вираз - спосіб запису даних у вигляді або *атому*, або *лісового списку*, де *лісовий список* – це список чийі елементи записуються всередині круглих дужок, а *атом* – це все те що не є лісовим списком.

Віртуальний список – це лісовий список, якого не має в текстовому вигляді LISP-програми, але семантично його наявність присутня.

Def-s-вираз – це s-вираз, що визначає якусь основну структурну одиницю програми: функцію, метод, глобальну змінну тощо.

Емпіричний режим – режим процесу порівняння піддерев, при якому піддерева вважаються однаковими без детальної перевірки їх ідентичності, якщо збігаються їх хеші та інші додаткові параметри.

Хешування дерева – це розрахунок хешу, який буде зберігатися в корені даного дерева та значення якого розраховується із хешів піддерев чи хешів термінальних вузлів й обов’язково з врахуванням порядку.

Опис способу

Програми на LISP-подібних мовах записані за допомогою s-виразів й s-вирази є одним із варіантів запису деревоподібних даних в текстовому вигляді. AST програми записаної s-виразами ніколи не буде мати атомів не в термінальних вузлах, а всі батьківські вузли є коренями звичайних чи віртуальних лісових списків. Це показано на рис. 1.

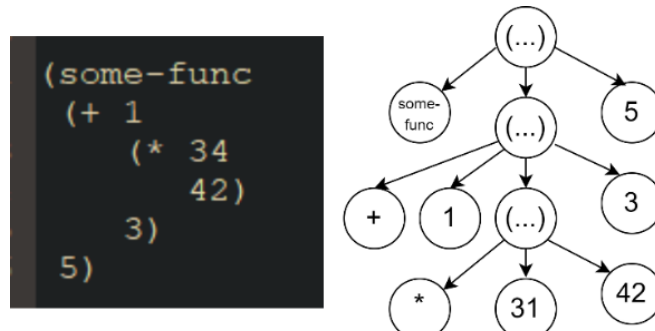


Рис. 1. Фрагмент програми мовою LISP та його AST

Опишемо використані категорії вузлів AST LISP-подібних мов та їх маркування на прикладі пари def-s-виразів def-v1 та def-v2.

Категорія “same”. Відповідний вузол є однаковим в def-v1 та def-v2 та знаходиться на однаковому ярусі їх дерев та що найголовніше в однаковій підпоследовності «same» вузлів на тому ж ярусі цих дерев.

Категорія “unique”. Відповідний вузол є унікальним для def-v1 чи def-v2. В залежності від застосування способу його, наприклад, для def-v1 його можна маркувати як “deleted”, а для def-v2 як “added”.

Категорія “moved”. Відповідний вузол та дерево коренем якого він є був переміщений з однієї частини дерева mod-def-ver1 в інше місце вже в mod-def-verf2.

Запропонований спосіб порівняння полягає у виконанні чотирьох етапів:

- 1) підготовка вхідних AST;
- 2) формування пар def-s-виразів для порівняння між собою;
- 3) знаходження “same” вузлів у відповідній парі порівняння;
- 4) виявлення ”moved” піддерев й “unique” вузлів.

Підготовка вхідних AST

На цьому етапі розраховуються додаткові параметри піддерев та термінальних вузлів, а саме хеші та кількість вузлів у піддеревих. Хеші мають розраховуватися з врахуванням того, що AST є впорядкованими деревами, при цьому складність даного процесу буде $O(n)$, де n – кількість вузлів. При формуванні хешів обов’язково має враховуватися семантичний тип відповідних вузлів. Це пов’язано з тим, що одні й ті самі s-вирази, які мають однаковий вигляд на синтаксичному рівні, в різних контекстах будуть мати різний семантичний сенс. Наприклад $(f\ 2)$ в $(let\ ((f\ 2)))$ та $(f1\ (f\ 2))$: у першому прикладі ми в локальному контексті присвоюємо символу f число 2, а в другому випадку маємо вже виклик функції f з аргументом 2.

Формування пар def-s-виразів для порівняння між собою

Реалізація даного етапу залежить від цілей використання способу та обраного LISP-діалекту, для якого він буде виконуватися. Розглянемо приклад def-s-виразів, що мають однакові імена, для мови Common Lisp. Якщо у файлі v1 визначено функції f1, f2, f3, а у файлі v2 визначено функції f2, f3, f4, то пари для порівняння між собою сформулюються з функцій f2 та f3, а f1 та f4 будуть занесені до відповідних списків «unique» цих файлів.

Знаходження “same” вузлів у відповідній парі порівняння

Для пошуку “same” вузлів, пропонується модифікація алгоритму пошуку LCS. Сам алгоритм пошуку LCS має застосування в програмах, що порівнюють файли на рівні рядків, але його легко модифікувати для роботи з деревами. Реалізація модифікації алгоритму пошуку LCS детально описана в [3]. Як результат маємо AST для def-v1 та def-v2, де ми отримуємо «коріння» із «same» вузлів, що можна побачити на рис. 2, де кольором виділено «same» вузли.

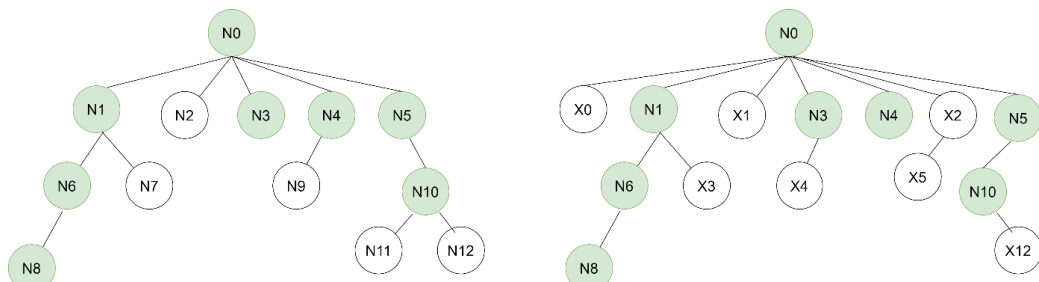


Рис. 2. Приклад результату роботи знаходження “same” вузлів

Виявлення “moved” піддерев й “unique” вузлів.

На даному етапі ліс піддерев з def-v1, що не були позначені як “same” вузли додаються до списку maybe-moved-from-v1, у якому піддерева мають знаходитись у порядку спадання за кількістю вузлів у піддеревах, а відповідний ліс з def-v2, обходиться будь-яким чином, й будується хеш-таблиця maybe-moved-in-v2-ht, де ключами є хеші з вузлів, а значенням є список вузлів, що мають цей хеш. Опишемо алгоритм пошуку «moved» вузлів, який починається після підготовки.

1. Якщо список maybe-moved-from-v1 не пустий, то забираємо перше піддерево pattern-tree із нього, та шукаємо його хеш в maybe-moved-in-v2-ht і якщо знаходимо, то виконуємо пункт 2, якщо ні, то пункт 3. Якщо список уже пустий, то алгоритм закінчує свою роботу.

2. У отриманому списку піддерев з цим хешем проводимо пошук піддерева, який не має маркування «moved» та який або точно чи емпірично збігається з pattern-tree. Якщо такий елемент було знайдено, то маркуємо його та pattern-tree як «moved», та переходимо до пункту 1, а якщо не знайдено, то до пункту 3.

3. Оскільки *pattern-tree* не було виявлено як переміщеного, то проводимо з ним операцію «*decay*» або «розкладання»: кореневий вузол *pattern-tree* позначаємо як «*unique*» та розкладаємо його на піддерева, які додаємо до списку *maybe-moved-from-v1* так, щоб його властивості не змінилися (був відсортований). Після чого переходимо до пункту 1.

Таким чином пошук відбувається від найбільших фрагментів до найменших, тобто атомів. Причина цього у тому, що пріоритетнішим є знаходження саме найбільших переміщених піддерев.

Ще однією важливою особливістю запропонованого способу пошуку є те, що ми не виконуємо повного перебору усіх можливих варіацій піддерев, які могли б бути виділені із заданого дерева як можливий переміщений фрагмент, а беремо до уваги лише вже наявні піддерева, яких у дереві з *n*-вузлами, відповідно буде теж *n*.

Висновки

Запропоновано спосіб, який враховує особливості AST LISP-подібних мов під час виконання їх порівняння. Ефективність цього способу забезпечується завдяки виконанню операції хешування AST, маркуванню вузлів AST всього лише на три види, обмеженню на вибір та форму можливих переміщених фрагментів, встановленню оригінальної пріоритетності фрагментів для пошуку та використанню хеш-таблиці під час їх пошуку. Напрямами подальших досліджень можуть бути розширення цього способу на можливість виявлення перейменувань та врахування цього факту при знаходженні переміщених фрагментів.

Література

1. *F. Magniez and M. de Rougemont*. Property testing of regular tree languages. *Algorithmica*, 49(2):127–146, 2007.
2. *Hashimoto, Masatomo & Mori, Akira*. (2008). Diff/TS: A Tool for Fine-Grained Structural Change Analysis. *Proceedings - Working Conference on Reverse Engineering, WCRE*. 279 - 288. 10.1109/WCRE.2008.44.
3. *Єрмоленко, Д. В.* Засіб порівняння версій програм на мові LISP з використанням абстрактного семантичного дерева : дипломний проєкт ... бакалавра : 123 Комп'ютерна інженерія / Єрмоленко Денис Вадимович. – Київ, 2021. – 75 с.

УДК 004.42

К.т.н., доцент Марченко О.І., студент Колпаков П.С.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

АНАЛІЗ ІСНУЮЧИХ СПОСОБІВ ОПТИМІЗАЦІЇ ПЕРЕДАЧІ ОДНОТИПНИХ JSON-ДОКУМЕНТІВ В КОМП'ЮТЕРНИХ МЕРЕЖАХ

Abstract

Oleksandr I. Marchenko, assoc. prof., PhD; Pylyp Kolpakov, student
The analysis of existing techniques for transferring JSON-documents with the same structure

This paper contains the analysis of optimization techniques for transferring JSON-documents with the same structure. JSON-format is a text format and it is redundant a lot. There are some techniques to deal with JSON redundancy, but they are not good enough in JSON compression or do require a lot of programmers work to implement. The new technique has good compression rate and almost no code changes to implement.

Вступ

Швидка передача великих обсягів інформації часто є тим самим “пляшковим горлечком”, що обмежує ефективність програмних продуктів. Загальноприйняті та широкоживані рішення нерідко є застарілими та непридатними до оновлення або виконання оптимізації. Розробники вимушені витратити свій час або на ситуативну оптимізацію передачі великих обсягів інформації, або ж на адаптацію до нових, але не таких простих та звичних рішень.

Постановка проблеми

У даній роботі розглянутий конкретний формат даних — JSON, і конкретна підмножина випадків — велика кількість однотипних/схожих документів. Задача полягає в стисканні інформації (шляхом переформатування, архівування чи будь-яким іншим шляхом) зі збереженням прийнятної швидкодії, або навіть з її покращенням.

JSON формат для передачі даних є важливим, оскільки в сучасній ІТ-індустрії є розповсюдженим використання REST-архітектури. Вона найчастіше передбачає обмін JSON-запитами, структура яких зводиться до невеликої множини більш-менш подібних класів (при тому, що сам формат JSON жодних класів не передбачає).

Мережевий обмін JSON-повідомленнями має немало серйозних недоліків, проблемних місць, “пляшкових горлечок”, й, відповідно, простору для покращення.

Відзначимо основні недоліки способу передачі даних на основі JSON-повідомлень.

1. JSON-формат є текстовим, а текстові формати не відзначаються компактністю.

2. JSON-документ не має класів, типізації, тому, відповідно, інформація щодо типу об'єктів, переліку полів тощо — передається в кожному відповідному повідомленні (документі), що зазвичай є надлишковим.

3. JSON-документ є динамічно типізованим, тому процес зчитування даних потребує додаткової логіки обробки (додаткових обчислень).

З іншої сторони, зазначимо переваги, які бажано зберегти при розробці нових способів передачі даних або вдосконаленні існуючих.

1. JSON-формат є надзвичайно поширеним, й підтримується усіма основними програмами, платформами, фреймворками для розробки веб-застосунків, веб-сервісів.

2. JSON-документ не має класів, типізації, тож відповідно розробка програмного забезпечення не потребує подекуди тривалого етапу визначення кінцевого формату повідомлень (документів у б. д. тощо).

3. JSON-формат є текстовим, тому процес налагодження програми частково полегшується.

4. JSON-формат не вимагає ніяких конкретних технологій або протоколів для передачі.

Термінологія

Формат JSON— широкоживаний текстовий формат даних, описаний у стандарті RFC8259 [1]

JSON-документ, JSON-повідомлення — дані у форматі JSON, представлені у текстовому вигляді об'єктом JSON (*object*, згідно стандарту)

Порівняльний аналіз існуючих рішень

Порівнюємо рішення зазначеної проблеми за основними критеріями: ступінь надлишковості повідомлень, простота та швидкість кодування та декодування повідомлень, простота впровадження технології, безпека даних, збереження основних переваг JSON-формату, наявність додаткових переваг, особливостей, обмежень.

1. Використання бінарних аналогів JSON-формату

Найбільш очевидним вирішенням проблеми надлишковості текстового формату є використання формату бінарного. У порівнянні з текстовим він дає наступні переваги:

- для комп'ютера бінарний формат є більш зручним для збереження та зчитування;

- текстовий формат є певною мірою надмірним, має більший розмір, а бінарний практично завжди є коротшим.

І такі бінарні формати для JSON-документів дійсно існують: BSON, Binary Ion, CBOR, UBJSON, *MessagePack*.

Принципово ці формати нічим один від одного не відрізняються. Зазначимо два основні моменти, спільні для них:

- вони зберігають структурну інформацію документів (імена полів, їх вкладеність);
- вони мають додаткову інформацію щодо конкретного типу значення (однобайтне число, двобайтне, рядок, булеве значення, масив тощо).

В цілому, використання бінарного формату даних є виправданим для випадків, коли питання швидкості запису/зчитування документу є суттєвим. Розмір окремого повідомлення при цьому зменшується, але це зменшення не є принциповим, оскільки ми ніяк не позбавляємось від динамічної типізації та зберігаємо рядкові імена полів.

2. Стискання JSON-повідомлень шляхом архівації

Найвідомішими утилітами, що дозволяють швидко стискати файли довільного формату, є `gzip` та `brotli`. Перша використовує для архівації алгоритм Deflate [2], а друга — його покращену модифікацію [3].

Постає питання — чи можна використовувати ці утиліти для оптимізації обміну однотипними JSON-документами?

Алгоритми сімейства LZ*[4] (Deflate та його варіації входять до цього сімейства) стискає дані шляхом виділення повторюваних рядків (заміни повторюваних даних їх на посилання на запис у так званому “словнику”) та шляхом використання кодів Гафмана. І якщо коди Гафмана дійсно здатні показати ефективність для чи не будь-яких текстових даних, то виділення повторюваних рядків може значно зменшити обсяг тільки у випадку дійсно надлишкових даних, з багаторазовими повторами одних і тих самих записів.

Алгоритми сімейства LZ* у якості “словника” повторюваних записів використовують так зване “ковзаюче вікно” — звичайний буфер з попередніми даними. Цей буфер є окремим для кожного файлу або документу, тож, відповідно — чим менший файл, тим менший буде словник і тим гірше файл буде стискатися цими алгоритмами.

JSON-повідомлення у типовій REST-архітектурі дуже рідко мають великий розмір, й відповідно, не повинні мати значну надлишковість. Структурні елементи JSON-формату, що мають бути спільними для усіх документів, є повторюваними. Але ці елементи є дуже короткими (1-3 символи), і їх скорочення навряд чи можливо.

Можна дійти наступного висновку: використання розглянутих алгоритмів стискання задля пришвидшення обміну JSON-повідомлення потребує певних додаткових обчислень (в теорії — відчутних), але при цьому виграш від стиснення даних може бути зовсім незначним. Алгоритми не накладають додаткових обмежень на дані, не потребує значних змін у програмному коді (оскільки може працювати на рівні протоколу) але доцільність використання архівації все одно є суперечливою.

3. Системи віддаленого виклику процедур: gRPC та Apache Thrift

Розглянемо тепер альтернативний до архітектури REST підхід, а саме — так звані RPC (Remote Procedure Call, [5]). Такий спосіб комунікації дозволяє програмі звертатись до процедур на іншому сервері як до своїх локальних. Однією з ключових задач, що при цьому потребує вирішення, є задача передача даних: сервери можуть використовувати різні фреймворки, мови програмування, операційні системи тощо, але структури даних мають бути однаковими в усіх випадках. Потрібен універсальний інтерфейс та універсальний спосіб передачі даних.

Розглянемо спосіб передачі даних за допомогою gRPC. Передача даних при такому способі реалізується за допомогою Protocol Buffers — бінарного формату серіалізації даних, за структурою подібних JSON/XML [6]. Розробник явно описує усі структури даних (об'єктні моделі, запити, відповіді) у прототипі, власноруч присвоюючи кожному полю власний унікальний ідентифікатор. Спеціальний компілятор **protoc** на основі описаного прототипу генерує програмний код для потрібної мови програмування, для потрібної операційної системи. Користувач, в свою чергу, може звертатися до серверу за допомогою звичайних програмних процедур, що виглядають так само як і локальні.

Розглянемо переваги способу передачі даних за допомогою gRPC порівняно до типової REST-архітектури.

1. Використовується бінарний формат даних без додаткової структурної інформації, що дозволяє значно зменшити об'єм даних, що передається, й відповідно збільшити швидкість.

2. gRPC є технологією з відкритим програмним кодом та з вільною ліцензією Apache 2.0.

3. Явний опис структур даних працює аналогічно статичній типізації в мовах програмування. Це захищає від зайвих помилок та конкретизує відповідність до доменної моделі [7].

4. Бінарний формат даних не є чутливим до імен полів (на відмінну від JSON/XML).

5. Додавання нових полів та/або структур даних не руйнує сумісність — неоновлені клієнти будуть працювати без помилок.

Але варто зазначити й недоліки цього способу передачі даних.

1. Від розробника вимагається додаткова робота — явний опис усіх структур даних.

2. Впровадження технології gRPC у готовому сервері з REST-архітектурою потребує значного переписування програмної логіки. За деяких обставин витрати на впровадження можуть не окупитися.

Тож, можна підсумувати: gRPC є потужною технологією, яка претендує на заміну типовому обміну текстовими JSON-документами (притаманними REST-архітектурі), і дозволяє значно пришвидшити обмін даними між серверами. Але, з іншого боку, впровадження цієї технології потребує немалих зусиль від розробників та, певним чином, обмежує сферу використання gRPC.

Слід зазначити, що існує багато подібних технологій, наприклад Apache Thrift, яка є більш ранньою за gRPC. За принципом роботи вона

практично не відрізняється, тому докладний опис цієї технології наводити не будемо.

Висновки

На основі зробленого аналізу можна стверджувати, що жоден з існуючих способів стискання інформації не дозволяє достатньою мірою вирішити задачу ефективного стискання великої кількості однотипних/схожих документів зі збереженням прийнятної швидкодії. Спостерігається дві ситуації: або спосіб надає незначне стискання (бінарні формати, алгоритми стиснення довільних даних), або спосіб задля свого впровадження змушує відмовитись від зручної типової REST-архітектури з динамічною типізацією (gRPC потребує явного опису усіх можливих типів).

Показано, що з метою зменшення обсягу даних, що передаються, доцільним є використання бінарного формату даних, який не буде містити додаткову структурну інформацію (структура об'єктів, імена полів). Але задля збереження основних переваг типової REST-архітектури варто уникнути явного опису структур даних для кожного повідомлення.

Тому є актуальною розробка нового способу передачі великих обсягів інформації, який, з однієї сторони, вирішує задачу оптимізації передачі JSON-документів за обсягом даних (для прискорення процесу їх обміну), а з іншої сторони — не вимагає явного опису програмістом структур даних повідомлень та зберігає прийнятну швидкість передачі.

Література

1. The JavaScript Object Notation (JSON) Data Interchange Format [Електроний ресурс] – Режим доступу: <https://www.rfc-editor.org/rfc/rfc8259>
2. RFC 1951: DEFLATE Compressed Data Format Specification version 1.3 [Електроний ресурс] – Режим доступу: <https://www.rfc-editor.org/rfc/rfc1951>
3. RFC 7932: Brotli Compressed Data Format [Електроний ресурс] – Режим доступу: <https://www.rfc-editor.org/rfc/rfc7932>
4. LZW (Lempel–Ziv–Welch) Compression technique — GeeksforGeeks [Електроний ресурс] – Режим доступу: <https://www.geeksforgeeks.org/lzw-lempel-ziv-welch-compression-technique/>
5. Bruce Jay Nelson (May 1981). Remote Procedure Call. PARC CSL-81-9 (Also CMU-CS-81-119). Xerox Palo Alto Research Center. PhD thesis.
6. Protocol Buffers Version 3 Language Specification | Google Developers [Електроний ресурс] – Режим доступу: <https://developers.google.com/protocol-buffers/docs/reference/proto3-spec>
7. Evans, Eric Domain-Driven Design: Definitions and Pattern Summaries. Domain Language Inc., 2006, p. 3.

К.т.н., доцент Марченко О.І., студент Хомутник Д.Ю.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

ПОРІВНЯЛЬНИЙ АНАЛІЗ СПОСОБІВ ОПИСУ РЕСУРСІВ ХМАРНОЇ ІНФРАСТРУКТУРИ

Abstract

Oleksandr I. Marchenko, assoc. prof., PhD; Dmytro Khomutnyk, student

Analysis and comparison of cloud infrastructure resources description techniques

This paper presents the analysis and comparison of differences between a description technique, proposed by authors, and existing tools for cloud infrastructure description. The comparison is made on a specific example of cloud infrastructure. Main advantages of the proposed technique are described. Results of a survey of programmers that shows differences in usage of both techniques are presented.

Вступ

Сьогодні існує багато засобів для автоматизації створення хмарної інфраструктури. Такі засоби надають як самі провайдери (CloudFormation у AWS [1], Azure Resource Manager у Microsoft Azure [2], Deployment Manager у Google Cloud Provider [3]), так і сторонні корпорації (Ansible у RedHat [4], Chef у Progress [5], Terraform у HashiCorp [6]). Через конкуренцію на ринку хмарних технологій, кожен провайдер надає ресурси, які функціонально майже не відрізняються, але відмінні від інших за способом опису. Через цю відмінність, усі засоби автоматизації вимагають повної деталізації описуваних ресурсів та взаємозв'язків між ними.

У статті [7] авторами був запропонований високорівневий спосіб опису ресурсів хмарної інфраструктури (проблемно-орієнтована мова опису), який відрізняється від існуючих більш високим та наближеним до користувача рівнем опису ресурсів хмарної інфраструктури з генерацією скриптів мовою Terraform і дозволяє спростити процес опису та знизити вимоги до кваліфікації розробника.

У даних тезах підкреслюються найбільш важливі особливості запропонованого способу на більш комплексному прикладі, а також виконується порівняльний аналіз запропонованого способу зі способом, який використовує мову Terraform.

Постановка задачі

Задача полягає в проведенні порівняльного аналізу запропонованого авторами способу опису хмарних ресурсів [7] з існуючим способом на основі мови Terraform: виділення ключових відмінностей між ними та опис переваг і недоліків.

Термінологія

Хмарні ресурси — ресурси, які фізично розташовані на потужностях хмарних провайдерів, підключених до мережі Internet.

Провайдер хмарних послуг/сервісів — компанія, що надає у використанні користувачам хмарні ресурси.

Віртуальна приватна хмара — Virtual Private Cloud (VPC) — ресурс, що об'єднує в одну логічну групу інші ресурси [8].

Зона доступності — Availability zone — ізольовані дата-центри, що розміщені в специфічних регіонах [9].

Розподільувач навантаження — Load Balancer — ресурс, що надає можливість розподілення навантаження по декільком віртуальним серверам [10].

Мережевий шлюз — Internet Gateway — ресурс, що надає доступ усіх ресурсів в підмережі до мережі Internet [11].

Порівняльний аналіз

Цільовою мовою трансляції запропонованого авторами способу опису хмарних ресурсів [7] було обрано мову Terraform через те, що ця мова є скриптовою і має найбільш гнучкі конструкції опису ресурсів хмарних інфраструктур серед існуючих способів опису, а також через її гнучкість у роботі з різними провайдерами, яка дозволить у майбутньому розширювати список провайдерів, платформи яких будуть підтримуватись при трансляції опису запропонованим способом. Ресурси різних хмарних провайдерів один в один мапуються у ресурси мови Terraform. Тому, проводячи порівняльний аналіз зі способом, що використовує мову Terraform, ми фактично порівнюємо наш спосіб опису з усіма способами, що надаються провайдерами хмарних ресурсів [1 - 3].

Для порівняння запропонованого способу з цільовою мовою трансляції Terraform, авторами пропонується конкретний приклад хмарної інфраструктури (рис. 1) та виконується показ на цьому прикладі ключових відмінностей зазначених способів.

Хмарна інфраструктура даного прикладу складається з одного VPC, у якому дві приватні підмережі розподілені по різним зонам доступності. Основний ресурс — віртуальний сервер — використовує автоматичне масштабування (автоматичне створення копій за необхідності) у цих двох

підмережах для підвищення доступності користувацького коду, що виконується на сервері, у разі збільшення навантаження на систему або відключення однієї з зон доступності.

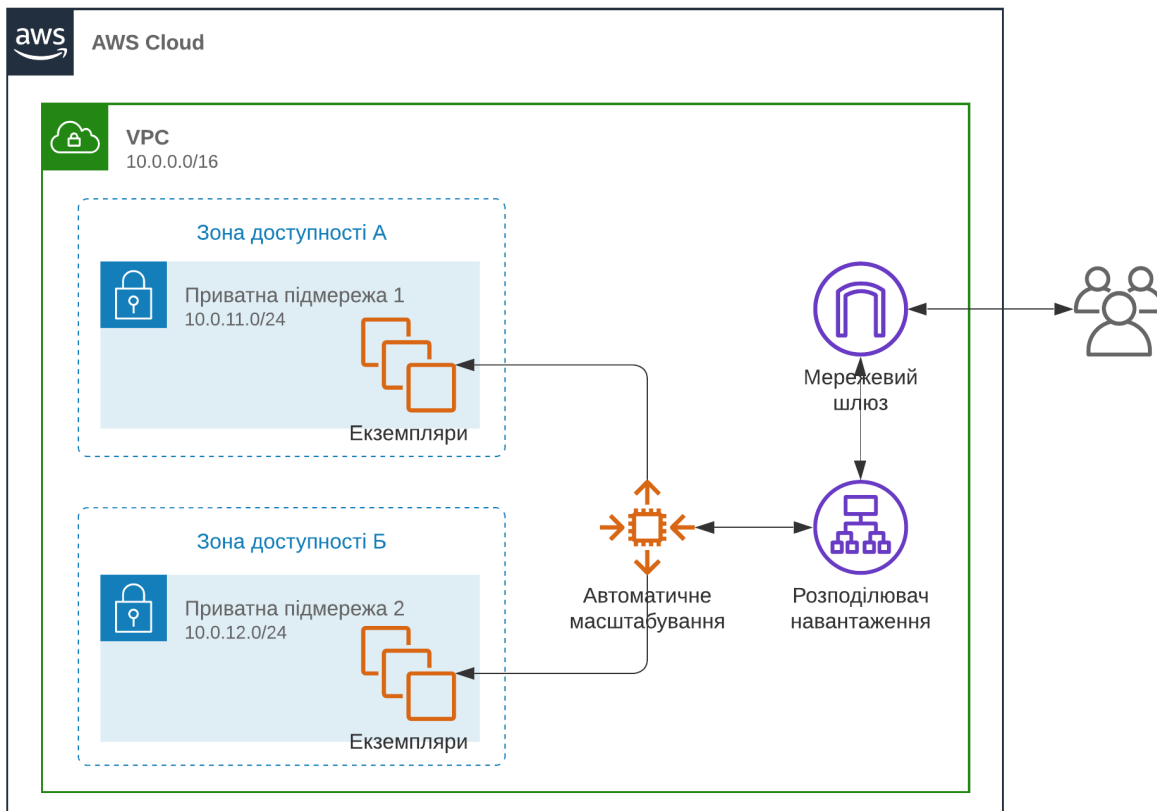


Рис. 1. Приклад хмарної інфраструктури

Для розподілення запитів між декількома копіями віртуального серверу, у VPC розміщений розподільвач навантаження — єдина пряма точка доступу для користувачів з мережі Internet. Для доступу з мережі та до мережі також необхідно створити мережевий шлюз, якому буде присвоєно публічну IP-адресу та який буде перенаправляти пакети з та до розподільвача навантаження.

Опис інфраструктури з прикладу запропонованим способом наведено на рисунку 2.

Відзначимо основні відмінності запропонованого способу порівняно до способу на основі мови Terraform.

1. Конструкції визначення взаємозв'язків між ресурсами. Запропонований спосіб визначає належність ресурсів один до одного за допомогою вкладеності.

2. Конструкції оголошення підмереж всередині VPC. Запропонований спосіб дозволяє скорочений вигляд опису підмереж з автоматичним розподіленням по різним зонам доступності.

```

VPC MainVPC {
  cidr_block    = "10.0.0.0/16"
  region       = "eu-north-1"

  Subnet Subnet1 = "10.0.11.0/24"
  Subnet Subnet2 = "10.0.12.0/24"

  EC2 RestService {
    name           = "example-rest-service"
    instance_type  = "t3.micro"
    image          = "ubuntu"

    autoscaling {
      min_size           = 1
      max_size           = 2
      desired_capacity   = 1
      subnets           = [Subnet1, Subnet2]
    }

    networking {
      ingress = http
      egress  = all
    }
  }
}

```

Рис. 2. Опис інфраструктури з прикладу запропонованим способом

3. Конструкції оголошення групи автоматичного масштабування з розміщенням екземплярів віртуальних серверів в створених підмережах. Запропонований спосіб дозволяє вказати лише бажану кількість одночасно запущених екземплярів та підмережі, в яких вони будуть створюватись. Весь шаблонний код буде згенеровано автоматично.

4. Конструкції оголошення розподільвача навантаження та мережевого шлюзу. Для розподілення мережевого трафіку між створеними екземплярами віртуальних серверів необхідно використовувати розподільвач навантаження з мережевим шлюзом. Запропонований спосіб автоматично згенерує необхідні для цього ресурси.

5. Конструкції налаштування зв'язку по мережі Internet — запропонований спосіб дозволяє використовувати ключові слова для найбільш розповсюджених налаштувань доступу вхідного та вихідного трафіку.

Для підтвердження заявлених переваг було проведено опитування серед програмістів різного досвіду і знань у сфері хмарних технологій. Програмісти, що були опитані, склали опис інфраструктури двома способами — мовою Terraform і запропонованим авторами способом — і при цьому заміряли час, який вони на це витрачали. В якості результатів опитування фіксувався час, який розробники витратили на опис

інфраструктури одним та другим способом, а також кількість помилок, які були зроблені при цьому.

Висновки

Авторами було проведено порівняльний аналіз запропонованого способу опису ресурсів хмарної інфраструктури зі способом на основі мови Terraform. Було представлено такі переваги використання запропонованого способу:

- спрощення опису інфраструктури для програмістів будь-якого рівня знань з хмарних засобів;
- зменшення часу опису інфраструктури;
- зменшення кількості можливих помилок в описі ресурсів та їхніх взаємозв'язків при виконанні цього опису.

Всі перелічені переваги є наслідком використання високорівневих абстракцій та автоматичної генерації шаблонного коду, запропонованих авторами. Користувач може кількома рядками описати лише необхідні йому деталі, а весь шаблонний код буде згенерований автоматично.

Один з перспективних напрямів розвитку запропонованого способу опису хмарних ресурсів полягає в підтримці генерації коду мовою Terraform для засобів різних провайдерів. Оскільки провайдери надають багато схожих ресурсів, то використання єдиного опису для різних провайдерів значно спростить перехід від одного провайдера до іншого з мінімальними трудовитратами.

Література

1. AWS CloudFormation [Електронний ресурс] – Режим доступу до ресурсу: <https://aws.amazon.com/cloudformation/>.
2. Azure Resource Manager Overview [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/overview/>.
3. Google Cloud Deployment Manager [Електронний ресурс] – Режим доступу до ресурсу: <https://cloud.google.com/deployment-manager/docs/>.
4. Ansible Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.ansible.com/>.
5. Chef Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.chef.io/>.

6. Terraform Documentation [Електронний ресурс] // Terraform – Режим доступу до ресурсу: <https://www.terraform.io/docs>.
7. Хомутник Д. Ю. Високорівневий спосіб опису ресурсів хмарної інфраструктури [Електронний ресурс] / Д. Ю. Хомутник, О. І. Марченко // № 48: Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – 2022. – Режим доступу до ресурсу: <https://doi.org/10.36910/6775-2524-0560-2022-48-18>.
8. Amazon Virtual Private Cloud (Amazon VPC) [Електронний ресурс] – Режим доступу до ресурсу: <https://aws.amazon.com/vpc/>.
9. Availability Zone [Електронний ресурс] — Режим доступу до ресурсу: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html#concepts-availability-zones>.
10. Load Balancer [Електронний ресурс] — Режим доступу до ресурсу: <https://aws.amazon.com/elasticloadbalancing/>.
11. Internet Gateway [Електронний ресурс] — Режим доступу до ресурсу: https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Internet_Gateway.html.

Асист. Молчанов О.А., магістрантка Государська А.О.

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

АНАЛІЗ І СТРУКТУРУВАННЯ КОНТЕНТУ

Abstract

Oleksii A. Molchanov, assistant; Hosudarsa Anastasiia, student

Analysis and structure of the content

This paper concerns the task of content analysis. Methods of content analysis and structuring are studied and discussed. A comparative analysis of several methods was carried out. The ways for further research are proposed as well.

Вступ

Аналіз контенту являє собою дослідження комунікаційних артефактів і документів, прикладом яких можуть слугувати зображення, тексти різних форматів, відео, аудіо тощо. Контент-аналіз використовується соціологами для досліджень закономірності в комунікації в систематичний і повторюваний способи [1].

Можна виділити таку ключову перевагу в використанні контент-аналізу, як неінвазивний характер, що надає йому перевагу над збором відповідей на опитування чи симуляцією соціального досвіду. В різних навчальних дисциплінах практики контент-аналізу відрізняються, але всі вони містять в собі такі складові, як спостереження за текстами чи артефактами або систематичне читання. Текстам або артефактам може бути присвоєно мітки (або коди). Таким чином можна вказати на наявність фрагментів, що мають значення для дослідження [2][4].

Завдяки систематичному позначенню вмісту тексту, дослідники мають змогу кількісно проаналізувати шаблони вмісту, використовуючи статистичні методи, або значення вмісту в текстах, використовуючи якісні методи. Все частіше в аналізі контенту використовуються комп'ютери для автоматизації маркування (або кодування) документів. Частота слів, довжина документів та інші описові дані можуть бути надані простими обчислювальними методами. Класифікатори машинного навчання можуть значно збільшити кількість текстів, які можна позначити, але наукова корисність цього є предметом дебатів. Крім того, доступні численні комп'ютерні програми автоматизованого аналізу тексту (САТА), які

аналізують текст на заздалегідь визначені лінгвістичні, семантичні та психологічні характеристики.

Постановка задачі

Задача полягає в дослідженні існуючих методів аналізу контенту та впровадженню ідей для його покращення.

Опис алгоритмів

Інструмент збору даних, який використовується в аналізі контенту, – це кодова книга або схема кодування. У якісному контент-аналізі кодову книгу створюють і вдосконалюють під час кодування, тоді як у кількісному контент-аналізі кодову книгу необхідно розробити та попередньо перевірити на надійність і валідність перед кодуванням [2].

Книга кодів містить докладні інструкції для кодувальників, а також чіткі визначення відповідних концепцій або змінних, які потрібно кодувати, а також присвоєні значення. Відповідно до чинних стандартів належної наукової практики, кожне дослідження контент-аналізу має надавати свою книгу кодів у додатку або як додатковий матеріал, щоб забезпечити відтворюваність дослідження.

На сервері Open Science Framework (OSF) Центру відкритої науки багато книг кодів досліджень контент-аналізу доступні у вільному доступі за допомогою пошуку «codebook». Крім того, База даних змінних для аналізу вмісту (DOCA) надає відкритий доступ до архіву попередньо протестованих змінних і створених книг кодів для аналізу вмісту. Заходи з архіву можуть бути прийняті в майбутніх дослідженнях, щоб забезпечити використання високоякісних і порівнянних інструментів.

DOCA охоплює, серед іншого, заходи щодо аналізу вмісту вигаданих засобів масової інформації та розваг (наприклад, заходи щодо сексуалізації у відеоіграх), медіа-контенту, створеного користувачами (наприклад, заходи щодо онлайн-мови ненависті), і ЗМІ та журналістики (наприклад, заходи щодо використання фотознімків у пресі, що висвітлює сексуальне насильство над дітьми, а також заходи персоналізації у висвітленні виборчої кампанії).

Контент-аналіз є дослідницьким інструментом, що використовується для визначення присутності певних слів, тем або концепцій у певних якісних даних (тобто тексті). Використовуючи контент-аналіз, дослідники можуть кількісно оцінити та проаналізувати присутність, значення та зв'язки певних слів, тем або понять. Наприклад, дослідники можуть оцінити мову, яка використовується в новинній статті, щоб знайти упередженість [2]. Потім дослідники можуть зробити висновки про

повідомлення в текстах, автора, аудиторію та навіть культуру та час, у якому було створено текст.

Джерелами інформації можуть бути інтерв'ю, відкриті запитання, нотатки польових досліджень, розмови або буквально будь-яка комунікативна мова (наприклад, книги, есе, дискусії, газетні заголовки, виступи, ЗМІ, історичні документи). Одне дослідження може аналізувати різні форми тексту в своєму аналізі. Щоб проаналізувати текст за допомогою контент-аналізу, текст має бути закодований або розбитий на категорії коду для аналізу (тобто «коди») [1]. Після того, як текст буде закодовано в категорії кодів, коди можна далі класифікувати в категорії, щоб ще більше узагальнити дані. Нижче наведено три різні визначення контент-аналізу.

- Визначення 1: «Будь-яка техніка для створення висновків шляхом систематичної та об'єктивної ідентифікації особливих характеристик повідомлень». (з Холсті, 1968)

- Визначення 2: «Інтерпретаційний і натуралістичний підхід. За своєю природою це як спостереження, так і розповідь, і воно меншою мірою покладається на експериментальні елементи, які зазвичай асоціюються з науковими дослідженнями (надійність, валідність і можливість узагальнення) (з Етнографії, спостережливих досліджень і нарративного дослідження, 1994-2012).

- Визначення 3: «Метод дослідження для об'єктивного, систематичного та кількісного опису явного змісту спілкування». (з Берельсона, 1952)

Правила використання аналізу контенту:

- Визначте наміри, фокус або комунікаційні тенденції особи, групи чи установи

- Опишіть ставлення та поведінкові реакції на спілкування
- Визначити психологічний чи емоційний стан осіб чи груп
- Розкрийте міжнародні відмінності змісту спілкування
- Виявіть закономірності у змісті спілкування

- Попередньо протестуйте втручання або опитування перед запуском

- Проаналізуйте фокус-групові інтерв'ю та відкриті запитання, щоб доповнити кількісні дані

Існує два загальних типи контент-аналізу: концептуальний аналіз і реляційний аналіз. Концептуальний аналіз визначає наявність і частоту понять у тексті. Реляційний аналіз розвиває концептуальний аналіз далі, досліджуючи зв'язки між поняттями в тексті. Кожен тип аналізу може призвести до різних результатів, висновків, інтерпретацій і значень [1].

З появою звичайних обчислювальних засобів, таких як ПК, популярність комп'ютерних методів аналізу зростає. Відповіді на відкриті запитання, газетні статті, маніфести політичних партій, медичні записи чи систематичні спостереження під час експериментів – усі вони можуть піддаватися систематичному аналізу текстових даних. Завдяки тому, що вміст комунікації доступний у формі машинозчитуваних текстів, вхідні дані аналізуються на частоти та кодуються в категорії для створення висновків.

Комп'ютерний аналіз може допомогти з великими електронними наборами даних, скорочуючи час і усуваючи потребу в кількох кодувальниках-людинах для встановлення надійності між кодерами. Однак для аналізу вмісту все ще можна використовувати програмістів-людей, оскільки вони часто можуть краще вибирати нюанси та приховані значення в тексті. Дослідження показало, що кодери-люди здатні оцінювати ширший діапазон і робити висновки на основі прихованих значень.

Мета контент-аналізу — «читати між рядків». Він має на меті визначити відповіді на запитання, де текст на щось натякає і не обов'язково є явним. Контент-аналіз — це дослідження, яке може аналізувати людське спілкування, те, як люди планують своє життя, що люди знають про щось і як на щось реагують.

Контент-аналіз став альтернативою традиційним запитам ЗМІ, які тоді використовувалися для дослідження громадської думки. Контент-аналіз використовує методи вивчення даних, зображень, друкованого тексту, звуків, соціальних медіа, статей, книг, журналів та Інтернету – головним чином для того, щоб зрозуміти, що мають на увазі люди, що вони дозволяють і що інформація, яку вони передають, говорить бізнесу чи суспільства в цілому. Контент-аналіз формує місток між кількісними та якісними методами дослідження, де можуть бути розглянуті деякі організаційні питання, які дуже важко вивчити, такі як поведінка організації, людські ресурси та проблеми клієнтів.

Аналізуючи присутність певних слів і тексту в певних якісних даних, зв'язок між словами та зображеннями, дослідники можуть зробити висновки щодо багатьох життєво важливих аспектів, таких як аудиторія, поведінка, культура та рівень задоволеності. Джерела даних для контент-аналізу переважно бувають двох типів:

1. Офлайн Офлайн-контент-аналіз базується на книгах, журналах, есе, інтерв'ю, дослідницьких нотатках, відкритих запитаннях і довідниках. Вибірка з офлайн-джерел представлятиме весь всесвіт. Однак у багатьох випадках офлайн-дані можуть бути застарілими.

2. Онлайн Зі швидким розвитком Інтернету джерела онлайнових даних набули значення. Онлайн-розмови, коментарі в соціальних мережах, огляди продуктів і відгуки клієнтів збираються з найновіших і оновлених посилань, що робить джерело даних більш релевантним.

Висновки

Повертання до обчислювальної соціальної науки вимагає від вчених перегляду того, як виконується аналіз контенту та чи потрібно його адаптувати, щоб залишатися релевантними у мінливому дослідницькому середовищі [3]. Щоб працювати з великими, складними даними, вчені звернулися до обчислювальних форм аналізу контенту та згодом перейшли від написання коду людиною до алгоритмічного способу створення коду.

Поки деякі процеси, такі як розробка протоколу кодування, вимагають лише реконцептуалізація, проте інші процеси, такі як необхідність звірення алгоритму рішення із золотим стандартом, навряд чи взагалі відповідають традиційним рамкам [2]. Алгоритмічний спосіб створення коду дає такі переваги: підвищена ефективність, прозорість.

Такі переваги не можна ігнорувати. Тим не менш, обчислювальні підходи також мають чіткі обмеження [1]. Їх залежність від цифрових матеріалів, які часто є сумнівної якості, та орієнтація на відносно простий й однозначний контент, щоб знизити ризик отримати результат, далекий від реальності, є проблематичними.

Таким чином, щоб контент-аналіз залишався актуальним у повороті до обчислювальної соціальної сфери науки, необхідно розвивати гібридний підхід, який зберігає як контекстну чутливість і валідність, які є центральними складовими для традиційного аналізу контенту, так і велику потужність та надійність обчислень. Тому непоганим рішенням буде розвиток інструментів і фреймворків, які полегшують взаємодію обчислювальних інструментів і людського досвіду.

Література

1. Berelson B. Content Analysis in Communication Research / B. Berelson // New York: Free Press, 1952. – 220 p.
2. Busha Ch., Harter S. Research Methods in Librarianship: Techniques and Interpretation / Ch. Busha, S. Harter // New York: Academic Press, 1980. – 417 p.
3. Krippendorff K. Content Analysis: An Introduction to its Methodology / K. Krippendorff // Beverly Hills: Sage Publications, 1980. – 413 p.
4. de Sola Pool I. Trends in Content Analysis / I. de Sola Pool // Urbana: University of Illinois Press, 1959. – 244 p.

УДК 004.05

К.т.н., асистент кафедри СПіСКС Морозов К.В., студентка
Шевченко А.О.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

СПОСІБ ОЦІНКИ ПОВНОТИ ТЕСТОВИХ НАБОРІВ НА ОСНОВІ ФОРМАЛЬНОГО МЕТОДУ ВИЗНАЧЕННЯ СТАНУ ПРОЦЕСОРІВ БАГАТОПРОЦЕСОРНОЇ СИСТЕМИ

Abstract

Morozov, K.V., Shevchenko A.O.

The method of assessing the completeness of test sets based on a formal method for determining the state of processors of a mutiprocessor system

The paper considers the problem of self-testing of multiprocessor systems using of the Preparat-Metz-Chen model. It concerns the implementation of algorithm and software that can be used to determine the optimal set of tests. In the case, when based on certain test sets an unambiguous result was not obtained it will help to determine an additional test set that is sufficient to establish all the states of the processors in the system. A step-by-step algorithm for simplifying complex expressions and determining the set of redundant tests are described.

Вступ

Сучасна техніка все більше використовує відмовостійкі багатопроцесорні системи (ВБС), зокрема, в якості систем керування. Такі системи зазвичай мають досить високі вимоги до надійності та продуктивності. Однією із задач, що виникає під час побудови ВБС, є організація процесу їх самотестування, тобто визначення стану кожного з процесорів системи, аби вчасно реконфігурувати її для забезпечення подальшого функціонування та якнайефективнішого використання наявних ресурсів. Один з підходів полягає у тестуванні процесорів системи окремим спеціалізованим вузлом. Перевагою такого підходу є відносна простота реалізації процедури тестування. З іншого боку, вищезазначений вузол також може виходити з ладу, стаючи таким чином вузьким місцем у забезпеченні заданого рівня надійності системи, що є суттєвим недоліком. Інший підхід базується на взаємних тестових перевірках процесорів один одним без використання додаткових модулів. Одним з недоліків такого підходу є більша складність організації тестування та інтерпретації результатів перевірок. Саме цьому підходу і присвячена дана робота.

Передбачається, що процес тестування відбувається відповідно до моделі Препарата-Метце-Чена [1], відповідно до якої результатом тестування справним процесором деякого іншого процесора буде 0, якщо останній є справним та 1 – в протилежному випадку. Якщо ж тестуючий процесор є несправним, то може бути отримано як 0, так і 1 незалежно від стану процесора, що тестується.

В [2] був запропонований алгоритм, який дозволяє точно визначити стани усіх процесорів системи, якщо кожний процесор має можливість тестування будь-якого іншого. На практиці далеко не завжди можна організувати тестування кожного процесора усіма іншими. В [3] запропоновано формальний метод, що дозволяє визначити стани усіх процесорів, не накладаючи обмежень на топологію міжпроцесорних зв'язків. При цьому передбачається, що система є M – діагностованою тобто такою, де стани усіх процесорів можуть бути встановлені, якщо з ладу вийшло не більше ніж M процесорів. Метод може бути застосований і до деякого часткового результату, отриманого на основі вже проведений тестів. Однак виконання тестів забирає частину продуктивності процесорів, часом досить велику, а отже їх кількість бажано мінімізувати. В [4] був запропонований підхід, який дозволяє виконувати аналіз множини тестових перевірок до їх виконання, що може бути використано зокрема для визначення оптимального набору тестів, що в свою чергу призведе до зменшення часу обрахунків та навантаження на систему. Крім того, якщо на основі проведення певного набору тестів, однозначного результату не було отримано, відповідно до [3] необхідно виконати додаткові тестові перевірки. Запропонований у [4] підхід може бути використаний для вибору оптимального набору таких перевірок.

Відповідно до [4], кожному тестовому експерименту, який полягає у тестування i -тим процесором j -го процесора, відповідає вираз наступного виду:

	$T_{ij} \triangleq (r_{ij} \oplus x_j) \vee \bar{x}_i \equiv r_{ij} \bar{x}_j \vee \bar{r}_{ij} x_j \vee \bar{x}_i \equiv r_{ij} x_i \bar{x}_j \vee \bar{r}_{ij} x_i x_j \vee \bar{x}_i = 1$	(1)
--	--	-----

Нехай в результаті проведення деякого числа тестових експериментів відповідно до [3], було отримано вираз виду V_K . Якщо не було проведено жодних експериментів приймемо. Розглянемо деяку множину додаткових тестів, кількість яких дорівнює k та котрій відповідає множина U_k виразів типу T_{ij} , для якої побудуємо вираз наступного виду:

	$V_{K+k} \triangleq V_K \wedge \bigwedge_{T \in U_k} T$	(2)
--	---	-----

Перетворимо вираз (2) до наступного виду, видаляючи в процесі перетворення усі кон'юнкції, що містять більшу кількість інверсій ніж M у змінних виду x_q :

	$V_{K+k} = \bigvee_{l=1}^L P_l C_l,$	(3)
--	--------------------------------------	-----

де C_l – елементарна кон'юнкція, що містить усі змінні виду x_q та далі буде зватись конституентою, а P_l – вираз, що містить лише змінні виду r_{ij} та буде називатись коефіцієнтом.

В [4] було доведено, що якщо для будь-яких двох коефіцієнтів (позначимо їх P_a і P_b) виконується , то набору тестових перевірок, що аналізується достатньо для встановлення станів усіх процесорів у системі. Більш того, якщо виконується вищезазначена умова та для деякої комбінації результатів тестових експериментів один з коефіцієнтів P_h приймає ненульове значення, то стан системи відповідає конституенті C_h . Як результат маємо те, що кожному коефіцієнту P_l відповідає деяка конституента C_l , яка в свою чергу відповідає певному стану системи. Під час тестування системи достатньо буде за результатами проведених тестових перевірок встановити коефіцієнт, котрий прийме ненульове значення, що дозволить одразу визначити стан усіх процесорів у системі.

Однією з переваг цього методу є те, що виконання складних перетворень виразу може бути перенесено з етапу експлуатації системи на етап її розробки. Це в свою чергу призводить до зменшення навантаження на процесори під час роботи системи. Розробники можуть оцінити повноту тестового набору, та шляхом перебору комбінацій тестових перевірок обрати мінімальний або близький до нього. Окрім цього складний аналіз результатів на етапі самотестування може бути замінений на просту процедуру пошуку відповідного рядка у таблиці коефіцієнтів.

Постановка задачі

Метод потребує значної кількості складних перетворень виразів, однак вже на етапі розробки. Цей процес є досить трудомістким для виконання вручну, таким чином, актуальною є задача його автоматизації, а саме розробка алгоритму для виконання перетворень та реалізація відповідного програмного засобу.

Опис програмного засобу

На вхід алгоритм приймає наступні параметри: топологія зв'язків процесорів у системі та максимальна допустима кількість несправних процесорів. Також може бути задана множина тестів, які вже були

проведені та їх результати. Якщо тестів попередньо проведено не було, то множина буде порожньою.

Програма матиме два режими роботи. У першому режимі передбачається, що жодних тестових перевірок виконано не було (тобто множина результатів попередньо проведених тестів не задана). У цьому випадку програма аналізуватиме на достатність множину всіх можливих, відповідно до заданої топології зв'язків, комбінацій тестових перевірок. Другий режим відповідає ситуації, коли користувач задав певний набір вже проведених тестів та їх результати. В такому випадку програма аналізує комбінації лише тих тестових перевірок, які ще не були виконані, враховуючи результати проведених тестів за рахунок формування на їх основі виразу V_k .

В обох режимах програма буде складний вираз виду (2). Складність цього виразу збільшується з потужністю заданої множини перевірок. Запропоновано алгоритм [5] спрощення виразів такого виду, що полягає у наступному. По-перше, аналізується вміст перших дужок ($T_l(x_i, x_j)$) та визначається набір змінних, які увійшли до його складу. Далі аналізується наступні дужки (вирази $T_l(x_i, x_j)$ один за одним) та знайдені в їх складі змінні порівнюються з множиною, отриманою з перших дужок. При першому співпадинні пошук припиняється і знайдені дужки розкриваються. Отриманий результат приймається за перші дужки і алгоритм продовжує пошук по тих дужках, що ще залишились. Процес закінчується після того, коли не залишиться більше дужок для розкриття. Запропонований алгоритм є простим у реалізації. В результаті його використання відбувається велика кількість скорочень, оскільки майже завжди після розкриття обраних дужок з'являються кон'юнкції, які містять схожі набори змінних.

В програмі також реалізовано алгоритм визначення множини надлишкових тестів, тобто таких, які не дають додаткової інформації щодо стану процесорів. Після побудови виразу (3), алгоритм розділяє кожну кон'юнкцію на конститuentу C_l та коефіцієнт P_l . Далі всі коефіцієнти аналізуються і серед усіх коефіцієнтів виконується пошук такого результату тесту r_{ij} , який би входив до кожного коефіцієнту. Якщо такий тест був знайдений, алгоритм повідомить про те, що він є надлишковим і може бути виключеним з набору.

Результатом роботи програми у першому режимі буде визначення факту достатності набору тестових перевірок для визначення станів усіх процесорів, а для другого режиму - набір додаткових тестових перевірок. Також будується таблиця відповідності коефіцієнтів векторам стану системи.

Висновки

В роботі проаналізовано деякі наявні підходи до організації самотестування відмовостійких багатопроцесорних систем. Було запропонований спосіб програмної реалізації формального методу, який дозволяє виконувати аналіз набору тестових перевірок, зокрема, здійснювати оцінку його повноти. Реалізований програмний засіб працює у двох режимах в залежності від заданої множини проведених тестів та їх результатів та може бути використаний для визначення оптимального набору тестів або застосований у випадку, коли на основі певних тестових наборів однозначного результату не було отримано. В такому випадку він дозволяє визначити додатковий тестовий набір, достатній для встановлення всіх станів процесорів системи.

Література

1. Preparata F.P., Metze G., Chien R.T. On the Connection Assignment Problem of Diagnosable Systems. IEEE Trans. Electron. Comput. 1967. ES-16. № 6. P. 848–854.
2. Belyavskii V.E., Valuiskii V.N., Romankevich A.M. and Romankevich V.A. Self-Diagnosable Multimodular Systems: Some Estimates of Testing // Autom. Remote Control. 1999. V. 60. No. 8. P. 1179–1183.
3. A. M. Romankevich, K. V. Morozov, Vitaliy A. Romankevich: A Formal Method for Determining the State of Processors in a Multiprocessor System under Testing. Autom. Remote. Control. 2021. № 82(3). P.460-467
4. Generalization of the Formal Method for Determining the State of Processors of a Multiprocessor System Under Testing Romankevich, A.M., Morozov, K.V., Romankevich, V.A. Lecture Notes on Data Engineering and Communications Technologies [this link is disabled](#), 2022, 134, pp. 363–375
5. Романкевич А.М., Морозов К.В., Шуляк А.О. Алгоритм розв'язку булевого рівняння для встановлення стану процесорів системи при її самотестуванні. Інформаційні технології – 2021: зб. тез VIII Всеукраїнської науково-практичної конференції молодих науковців. - 2021. - м. Київ. с.136-137

УДК 519.688

К.т.н., доцент Павловський В.І., студент Денисенко І.В.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

АЛГОРИТМ ФІЛЬТРАЦІЇ ДЖЕРЕЛ ОСВІТЛЕННЯ, ЩО НЕ ПОТРАПЛЯЮТЬ НА ТАЙЛ ЕКРАНУ

Abstract

Volodymyr I. Pavlovskiy, assoc. prof., PhD; Ivan Denysenko, student
Light culling algorithm for tile-based lighting

This paper describes algorithm for efficient light culling for tile-based lighting. Main difference of this algorithm is its higher accuracy in terms of reducing false positives and multithreaded execution, which leads to better performance and less time required for lighting calculations.

Вступ

Відео контент грає важливу роль в сучасному суспільстві. Майже кожна людина щодня споживає його в тому чи іншому вигляді. При цьому, з розвитком технологій, змінювалися і підходи до його створення. Так відносно молодим напрямком створення графічного контенту є 3D комп'ютерна графіка, що відрисовується в реальному часу, що передбачає обчислення освітленості графічної сцени для кожного пікселя екрану – найменшої неподільної частинки екрану.

В такому випадку дуже важливим постає задача оптимізації алгоритмів обчислення освітленості, спрямована на збільшення їх швидкодії, адже обчислювальні спроможності будь-якого комп'ютера обмежені, зважаючи на кількість пікселів досить велика.

Найбільш поширеним алгоритмом обчислення освітленості є розбиття екрану на тайли однакового розміру, де для кожного з них визначається які джерела освітлення потрапляють в даний тайл. Для кожного тайлу відбувається фільтрація джерел освітлення, тобто, джерела освітлення, що не перетинаються з тайлом, відкидаються під час обчислення освітленості даного тайлу. Детально алгоритм тайлового освітлення описано в [1].

Але існуючі на даний момент алгоритми фільтрацій джерел освітлення дають багато хибних позитивних результатів [2], бо не враховують кожен

піксель окремо, а тільки весь тайл загалом. Тобто, в процес обчислення освітленості тайлу потрапляють джерела освітлення, які, насправді, не потрапляють на жоден з пікселів даного тайлу.

Постановка задачі

Суть задачі полягає в створенні алгоритму фільтрації джерел освітлення для кожного тайлу екрану з визначенням перетину джерела освітлення з кожним пікселем тайлу.

Термінологія

Тайл екрану – прямокутна область на екрані, що має глибину (глибина визначається графічною сценою). Зазвичай це 8*8 або 16*16 пікселів в ширину та висоту.

Текстура – структура даних, що розміщується в пам'яті GPU та використовується як двомірний масив.

False Positive – випадок, коли алгоритм повертає хибний позитивний результат.

Опис алгоритму фільтрації джерел освітлення

Представлений нижче алгоритм визначає, чи потрапляє джерело освітлення на кожен з пікселів тайлу. Якщо джерело освітлення не потрапляє на хоч один з пікселів, таке джерело відкидається. При цьому, джерело освітлення може перетинатися з тайлом.

Даний алгоритм виконується після формування карти глибини сцени – текстури, що зберігає нормалізовані значення глибини сцени від 0 (ближче до камери) до 1 (далі від камери). Дана текстура використовується для визначення глибини кожного пікселя тайлу. Значення в даній текстурі позначають глибину відносно Near plane та Far plane – умовних площин, що задають межі зрізаної піраміди огляду (viewing frustum).

Першим кроком роботи алгоритму є розбиття екрану на тайли однакового розміру. Кожен тайл представляє собою зменшену в розмірі зрізану піраміду огляду. Ширина та висота тайлу має підбиратися таким чином, щоб екран міг бути представлений цілим числом тайлів. Схематичне зображення тайлу представлено на рисунку 1. Жовтим кольором позначений viewing frustum, синім – tile frustum.

Наступним кроком є визначення глибини тайлу, в яку потрапили об'єкти графічної сцени. Вона визначається шляхом розрахунку мінімальної глибини та максимальної глибини сцени в даному тайлі і обчислюється завдяки порівнянню значень глибини в кожному з пікселів даного тайлу.

Таким чином, визначається мінімальне та максимальне значення глибини в даному тайлі.

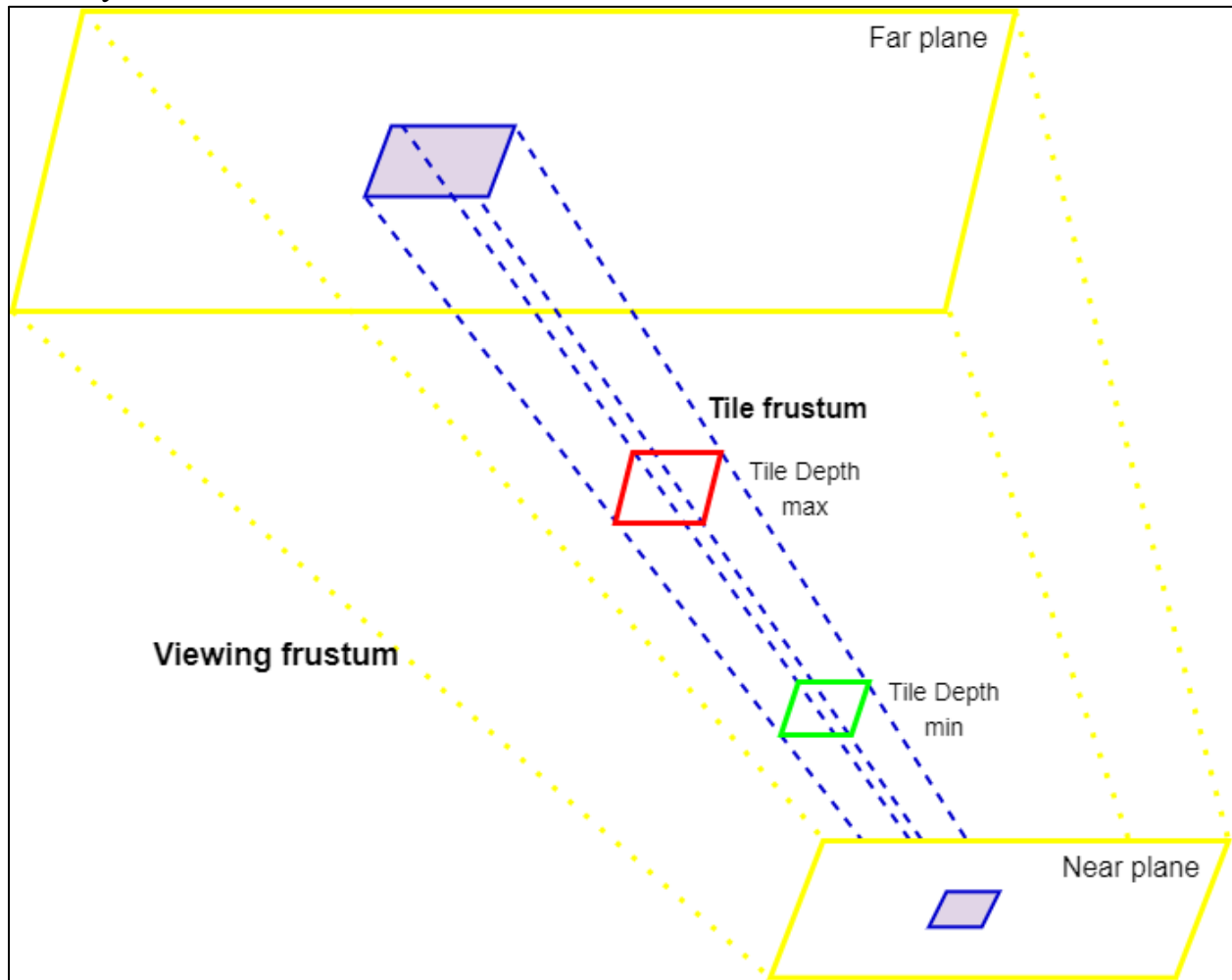


Рис. 1. Схематичне зображення тайлу

Наступним кроком є розбиття даного тайлу на сегменти відносно глибини в межах мінімального та максимального значення глибини тайлу. Схематичне зображення тайлів представлено на рисунку 2.

Далі потрібно сформувати маску глибини тайлу. Маска глибини представляє собою число, де біт з індексом i відповідає сегменту з індексом i . Якщо даний біт має значення 1, то принаймні один піксель тайлу має глибину, що потрапляє в даний сегмент.

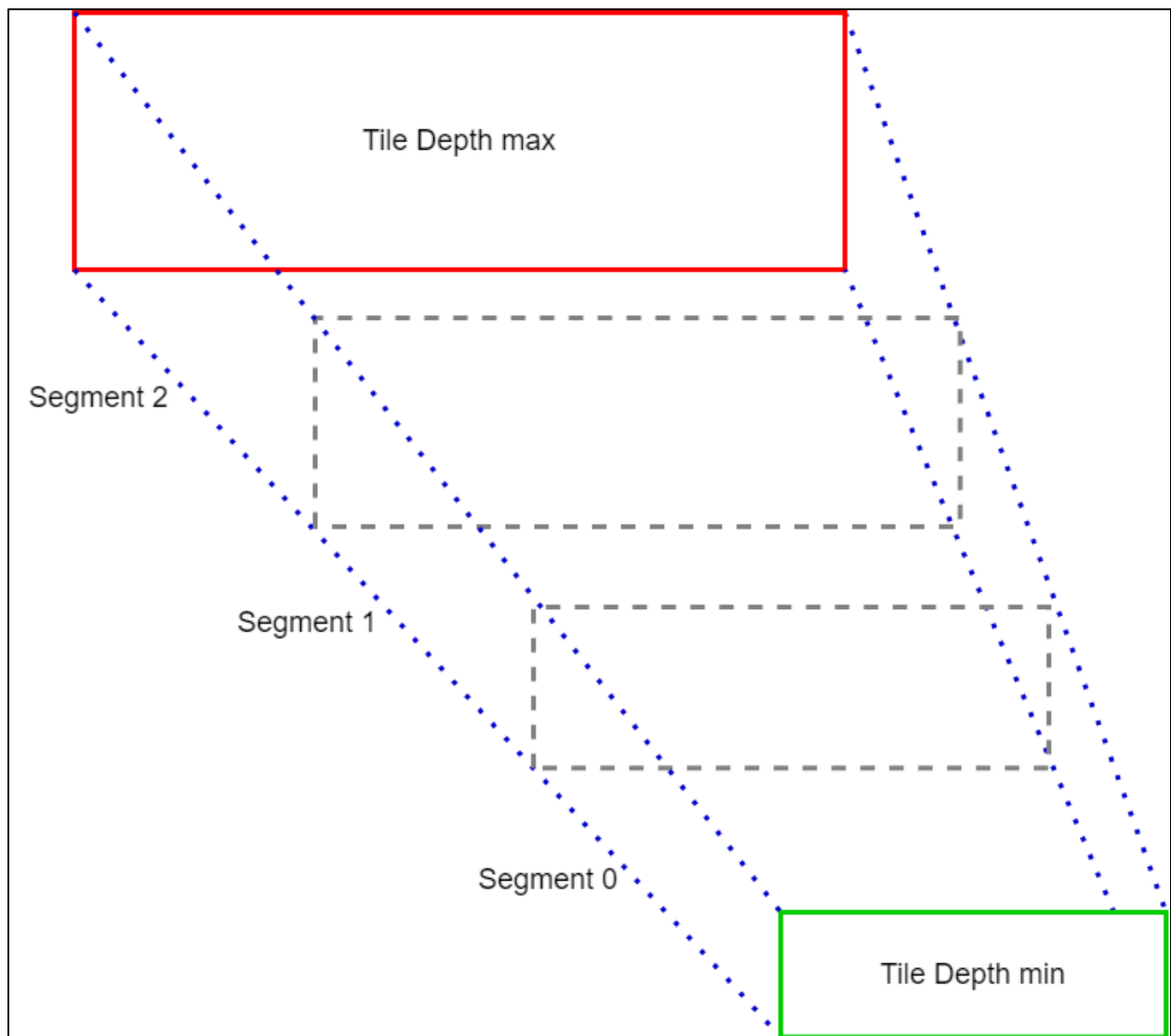


Рис. 2. Розбиття тайлу на сегменти

Далі визначається маска глибини джерела освітлення і порівнюється з маскою глибини тайлу за допомогою операції AND. Якщо результат має значення нуль, то дане джерело освітлення не потрапляє на жоден піксель тайлу.

Для прикладу розглянемо горизонтальний зріз тайлу з шириною в 4 пікселя та одне джерело освітлення, зображеного на рисунку 3.

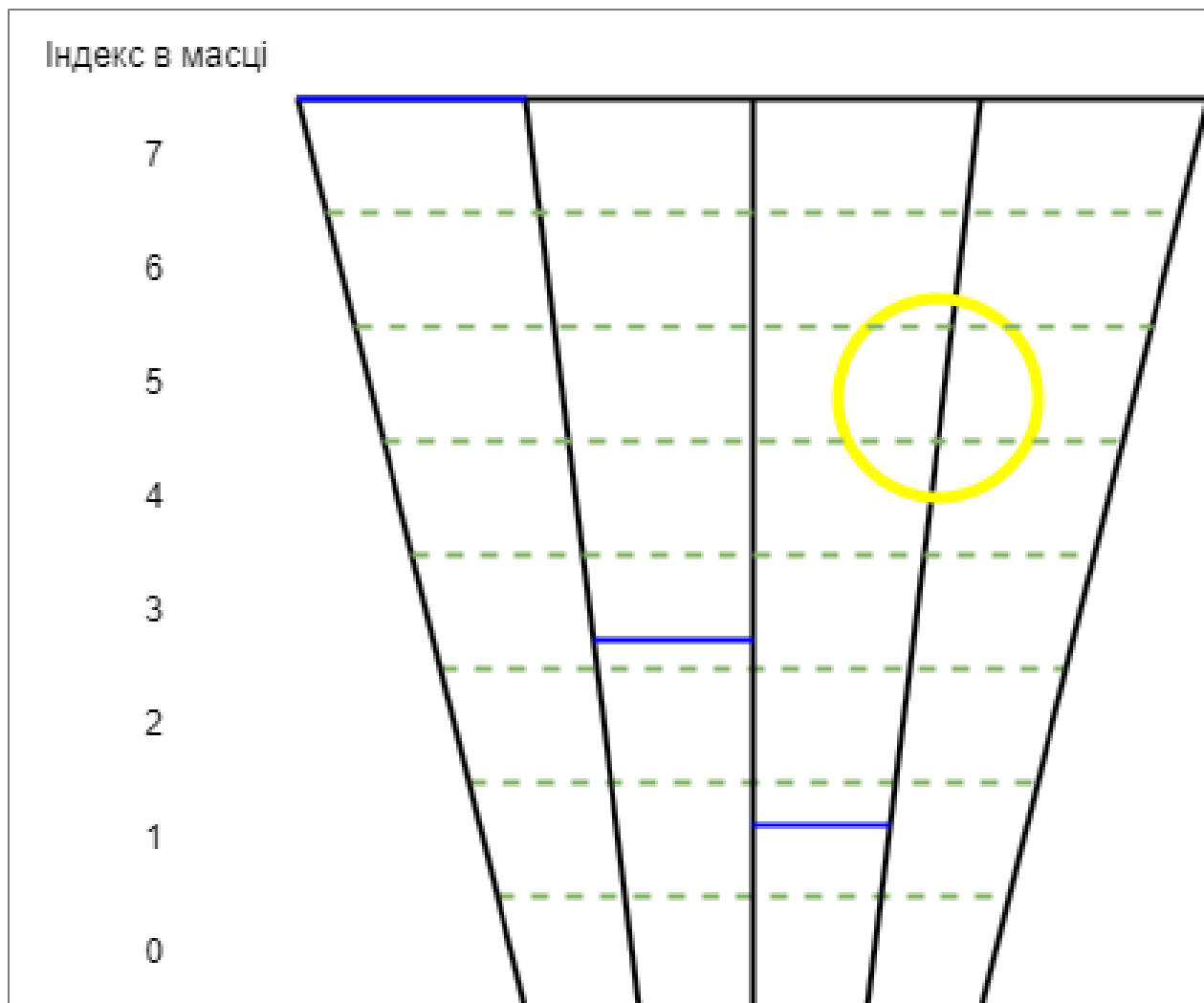


Рис. 3. Горизонтальний зріз тайлу

На рисунку 3 представлений горизонтальний зріз тайлу від мінімального (знизу) до максимального (зверху) значення глибини. Справжні значення глибини для кожного пікселя позначені синім кольором. Джерело освітлення має жовтий колір.

Визначимо, чи потрібно враховувати дане джерело під час розрахунку освітлення для даного тайлу.

Розрахунок маски глибини даного тайлу дає результат 10001011, адже пікселі мають значення глибини, що потрапляє в сегменти з індексами 0, 1, 3 та 7. Маска глибини джерела освітлення, що позначений жовтим кольором, буде 01110000, бо джерело освітлення потрапляє в сегменти з індексами 4, 5 та 6. Наступним кроком є побітова операція І між цими масками: $10001011 \text{ And } 01110000 = 00000000$. Результатом є нуль, отже, дане джерело освітлення хоч і потрапляє в даний тайл, проте, не потрапляє в жоден з

пікселів тайлу, а, значить, має бути відкинуте під час розрахунку освітлення для цього тайлу.

Головною перевагою даного алгоритму є можливість його паралельного виконання для кожного пікселя окремо, виконуючи операції AND та OR атомарно, що може суттєво зменшити час виконання даного алгоритму.

Також даний алгоритм дозволяє з більшою точністю відфільтровувати джерела освітлення, що потрапляють в тайл, проте, не потрапляють на жоден з пікселів. Це дозволяє зменшити об'єм обчислень під час розрахунку освітленості сцени.

Висновки

Запропонований алгоритм фільтрації джерел освітлення дозволяє з більшою точністю визначати, які з джерел освітлення не потрапляють на тайл екрану, що зменшує час виконання обчислень освітленості графічної сцени. Також даний алгоритм можна виконувати паралельно для кожного пікселя, що зменшує час його виконання.

Література

1. [Harada et al12] Takahiro Harada, Jay McKee, Jason C Yang, “Forward+: Bringing Deferred Lighting to the Next Level”, Eurographics 2012.
2. [Harada 12] Takahiro Harada, “Culling for Forward+”, Technical Briefs, SIGGRAPH Asia 2012

К.т.н., доцент Павловський В.І., студент Ковезюк А.О.

Національний технічний університету України
«Київський політехнічний інститут імені Ігоря Сікорського»

АЛГОРИТМ СИМВОЛЬНОЇ ПРОЦЕДУРНОЇ ГЕНЕРАЦІЇ КОНТЕНТУ ЗА ТЕКСТАМИ ПРИРОДНОЇ МОВИ

Abstract

Volodymyr I. Pavlovskiy, assoc. prof., PhD; Anrii O. Koveziuk, student
Symbolic procedural content generation algorithm by natural language text

Content generation for computer systems is widely used in modern software, but most widespread algorithms involve arithmetic algorithms a computation – such as different special noises, or random selection from a set of variants, so symbol-oriented approach is generally unknown. This work takes an approach of using discrete values – symbols, and rules – relations between symbols to provide realistic, immersive, seamless content. To support such algorithm, supporting algorithm that creates rulesets from user provided input and presenting view of generated data. To define the ruleset, natural-language processing algorithm is used, to display output – tiled 2D game engine.

Вступ

Сучасні програмні засоби широко використовують методи процедурної генерації інтерактивного контенту – моделей, аудіо, сценаріїв. Якість процедурно згенерованого контенту визначається наступними параметрами:

- логічна зв'язаність згенерованих елементів між собою та з негенерованими компонентами,
- незалежність від деталей імплементації. Внутрішнє представлення даних не повинно впливати на результат генерації.

На практиці поширення набули алгоритми процедурної генерації ландшафту з використанням шумів (Шум Перліна, симплексний шум, шум Ворлея) [1], фрактальні алгоритми [2], алгоритми емуляції фізичних процесів. Сценарії генеруються шляхом симуляції, аудіо створюється методами штучного інтелекту [3] Такі алгоритми генерують одноманітний результат, і важко піддаються контекстним модифікаціям. У даній статті пропонується метод символної процедурної генерації абстрактних моделей, що оперує дискретними величинами.

Постановка задачі

Задача полягає у створенні алгоритму символної процедурної генерації моделі даних, яка може бути використана для генерації контенту, із використанням тексту на природній мові як джерела логічних визначень.

Термінологія

Модель – граф, який є представленням логічних відношень між визначеними об’єктами. Вершини графу містять *теги* – символи що визначають окремий об’єкт, ребра визначають відношення між об’єктами.

Словник – список речень природньої мови у формі синтаксичного дерева.

Опис алгоритму

Алгоритм процедурної генерації приймає початкову модель, що містить початкову інформацію, та *словник*. Початковий стан моделі може бути однаковим для різних словників.

Результатом роботи алгоритму є модифікований граф моделі (рис. 1), що відображає відношення, виражені у словнику. Словник містить речення у

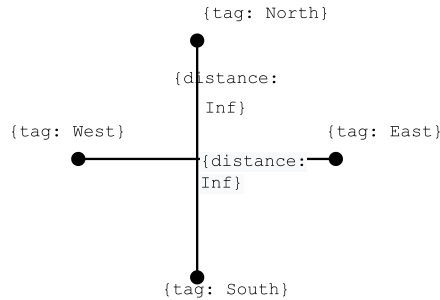


Рис. 1. Приклад вхідної моделі – граф, ребра якого визначають відношення між вузлами

формі синтаксичного дерева – графу, листи якого містять символи, а вузли – зв’язки між ними (рис. 2).

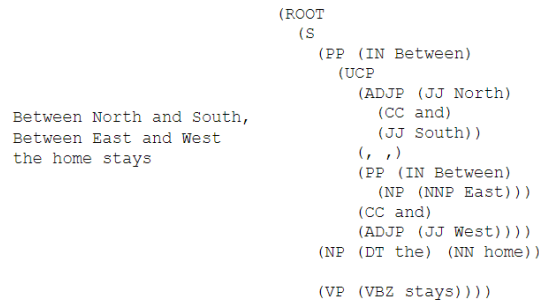


Рис. 2. Структура речення, представлена у вигляді синтаксичного дерева

Символи є дискретними неарифметичними значеннями. Для модифікацій моделі достатньо двох операцій додання - вузла та ребра (табл. 1).

$$\begin{aligned}
 & add(G_a, N) \rightarrow G_b, N \in G_b \\
 & connect(G_a, A, B), A \in G_a, B \in G_a \rightarrow G_b, \{A, B\} \in G_b
 \end{aligned}$$

Таблиця 1. - Визначені операції над графом

Як комбінації цих двох операцій встановлюються базові визначення відношення простору, часу, приналежності (рис.3). Таким чином виконується вимога ізоляції моделі від вхідних даних, а саме ізоляція моделі від вхідного словника, який не повинен визначати операції з графом.

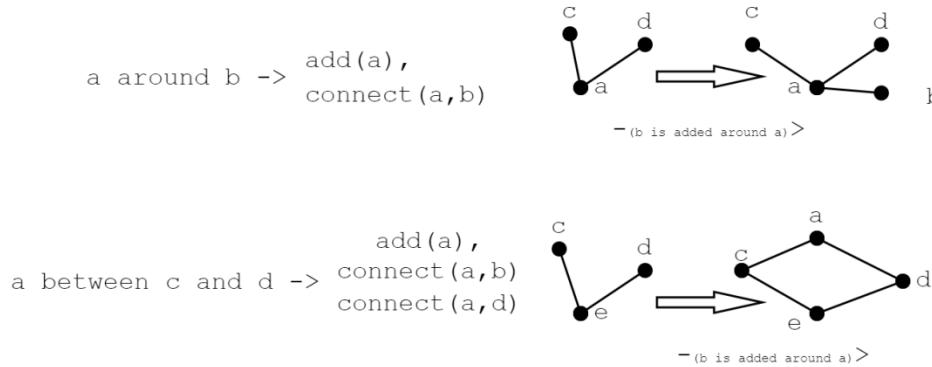


Рис 3. Приклад базових операцій та відповідних модифікацій графа

Під час обробки синтаксичного дерева із словника, створюються нові визначення операцій та виконуються операції модифікації моделі.

Кожне дерево складається із не-термінальних вузлів, що визначають синтаксичну структуру речення, і термінальних вузлів, які містять інформацію про зміст речення. Синтаксична структура речення складається із наступних частин (табл. 2):

Таблиця 2

Синтаксичні елементи речення

Символ	Значення	Приклад
S	речення	<i>the man walked</i>
NP	іменникове словосполучення	<i>a dog</i>
VP	дієслівне словосполучення	<i>saw a park</i>
PP	прийменникові фрази	<i>with a telescope</i>
Det	визначник	<i>the</i>
N	іменник	<i>dog</i>
V	дієслово	<i>walked</i>
P	прийменник	<i>in</i>

Оскільки природні мови є контекстно-залежними, контекст речення повинен визначатися і використовуватися при обробці речення. Для отримання контексту найбільш доцільним є видалення із структури речення іменникового словосполучення. Для досягнення такого результату, нащадки кожного вузла, які містять іменникове словосполучення, обробляються першими, і повертають контекст для обробки інших частин речення – так визначається порядок обходу.

Під час обробки синтаксичного дерева кожен синтаксичний елемент перетворюється на список операцій за окремим алгоритмом.

Обробка речення як синтаксичної структури

Речення є кореневою вершиною синтаксичного дерева, і може містити багато предикативних одиниць, кожна із яких має власний контекст і повинна оброблятися окремо. Члени речення сортуються спочатку за ознакою відношення до певної предикативної одиниці, потім за контекстом: (NP, VP > N, V > PP, P).

Обробка іменникового словосполучення

Іменникове словосполучення (рис. 4) є джерелом контексту об'єкту, часу, приналежності і повертає модифікований контекст, не виконуючи операцій над моделлю. Наприклад, для речення «*In their public lectures they have even claimed that the only evidence that Khufu built the pyramid is the graffiti found in the five chambers*», для виділеного іменникового словосполучення, повертається контекст, але модель не модифікується:

```
process_noun_phrase(np: (NP (DT the) (CD five) (NNS chambers))) ->
{Context: {target: {"house", "first"}, {"house", "second"}, {"house",
"third"}, {"house", "forth"}, {"house", "fifth"}}
```

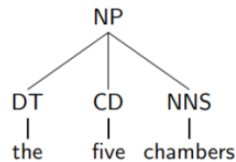


Рис. 4. Приклад структури іменникового словосполучення

Обробка дієслівного словосполучення

Дієслівне словосполучення є джерелом контексту зв'язку, стану об'єктів, може містити іменникові словосполучення і модифікує модель. Операції над моделлю виконуються під час зміни контексту.

Визначення нових правил

Під час обробки речення неможливо обійтися заздалегідь заданими визначеннями, під час проходження по реченню створюються нові операції. Операції виділяються із іменникових словосполучень і являють собою дерева виклику ф-цій, які визначаються через примітивні операції.

Результати роботи

Для перевірки роботи алгоритму була реалізована генерація оточення для простого опису будинку. Початковими даними для алгоритму є текст: «*The building is square shaped. The house is fully surrounded by stylish gardens. The house is equipped with an average kitchen and one small bathroom*» і початкова модель з визначеннями простору та вкладеності: `space:{inf, big, average, small}, {building:{house}, (inside: space)}, shape:{round, square}, spacial_releation:{inside, outside}, {room: {kitchen, bathroom}, (inside:building)}`

Результатом роботи алгоритму є модифікована модель (рис. 5, рис. 6)

```
{{space_1{inf}, square, kitchen, bathroom, space_3{average}, space_2{medium}, space_4{small}},  
{{space_1, space_2}(inside), {space_2_square}(is), {square, house}(contains), {bathroom,  
house}(is_part_of),{kitchen, house}(is_part_of), (bathroom, space_4)(has_size),(bathroom,  
space_3)(has_size),
```

Рис. 5. Текстове представлення моделі-результату

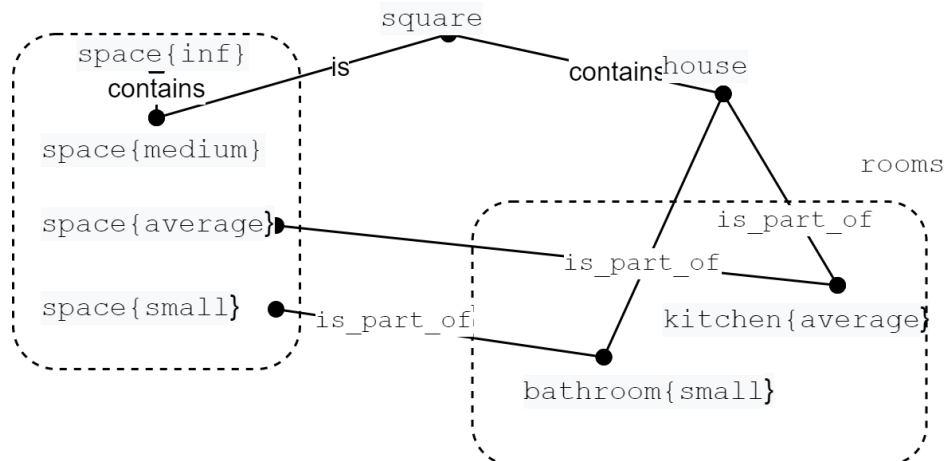


Рис. 6. Графічне представлення моделі-результату

Висновки

Символьна процедурна генерація здатна створювати моделі, якісно кращі за моделі, згенеровані арифметичними методами. Методи символної генерації здатні створювати логічні, неочевидні моделі і можуть використовувати текст на природній мові як джерело інформації про модель.

У подальшому є сенс провести роботу над швидкодією алгоритму для обробки більш складних моделей та додати підтримку інших мов, наприклад, української.

Література

1. G. B. Thomas Rose, "Algorithms and Approaches for Procedural Terrain Generation - A Brief Review of Current Techniques," 2016.
2. T. D. C. J. G. A. Krista Bird, 2013. [Online]. Available: https://web.williams.edu/Mathematics/sjmillier/public_html/hudson/Dickers_on_Terrain.pdf.
3. T. A. Tanya X. Short, Procedural Storytelling in Game Design, 2019.

К.т.н., доцент Павловський В.І., студент Янечко А.С.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

ДОДАТОК 3 ПЕРЕТВОРЕННЯ SDR КОНТЕНТУ В HDR З ВИКОРИСТАННЯМ МОДИФІКОВАНОГО АЛГОРИТМУ

Abstract

Volodymyr I. Pavlovskiy, assoc. prof., PhD; Andrii Yanechko, student
Application for transforming SDR content to HDR using modified algorithm

This paper provides comparison between SDR and HDR and describes modified algorithm for transforming SDR content to HDR. The main modification is adding new step to the algorithm, the inverse tone mapping curve. The inverse tone mapping curve allows to control limits of white and black tones for each frame, which leads to better content transformation.

Вступ

На сьогоднішній день відео та аудіо контент споживає майже кожен користувач смартфона, комп'ютеру чи телевізору. Особливо це стосується відео зображень, якість яких має задовольняти зростаючий попит постійно розширюючогося кола споживачів та є критичним при вирішенні складних науково-технічних, медичних і оборонних задач. Відповідно сфера виробництва дисплеїв стрімко розвивається, в напрямку покращення якості зображень.

На сьогодні основним форматом представлення зображень на дисплеї є SDR формат. Однак його властивості не завжди відповідають зростаючим вимогам, а можливості дисплеїв постійно поліпшуються. Тому в останні роки для представлення зображень на дисплеї пропонується HDR-формат, властивості якого дозволяють відтворювати зображення у кращій кольоровій гаммі, з більшою яскравістю та з більшою кількістю деталей. Порівняльна характеристика SDR та HDR форматів наведена у таблиці 1.

Відповідно все більше і більше нових візуальних пристроїв отримують повноцінну підтримку відображення HDR контенту, тому актуальним є завдання в трансформації існуючих SDR зображень в HDR формат.

Таблиця 1

Порівняння SDR та HDR

	SDR	HDR
Максимальна яскравість	100 кд/м ²	1000 кд/м ²
Коефіцієнт контрастності	1200:1	20000:1
Глибина кольору	8 біт	10 біт
Стандартний простір кольорів	Rec. 709	Rec. 2100
Кількість основних кольорів (відтінків)	256	1024
Максимальна роздільна здатність	4К	4К

На сьогодні вже є ряд алгоритмів трансформації існуючих SDR зображень в HDR формат, але більшість компаній, які займаються поліпшенням якості контенту, з комерційних цілей приховують власні модифікації, проте всі вони не забезпечують повної коректності перетворень, деякі краще перетворюють темні ділянки, а деякі світлі.

Постановка задачі

Модифікація алгоритму перетворення SDR контенту в HDR, що забезпечує корегування граничних меж світлих та темних тонів, використовуючи зворотну криву тонального перетворення та створення на його основі додатка, це дозволить корегувати трансформацію зображень під кожен фрейм окремо.

Термінологія

HDR (High Dynamic Range) – набір параметрів зображення, яке характеризується підвищеною яскравістю, більшою кольоровою гамою та контрастністю [1].

SDR (Standard Dynamic Range) – набір параметрів зображення, які на сьогодні вважаються стандартними для якісного відображення зображень [1]. На сьогоднішній день майже весь відео контент знаходиться у SDR форматі.

Опис алгоритму перетворення та його модифікації

На теперішній час однозначного алгоритму для абсолютно коректного перетворення SDR зображення у HDR не існує, адже усі математичні перетворення це спроба так чи інакше покращити результат трансформації, а більшість етапів було додано з часом, з розумінням, що той чи інший етап перетворення суттєво впливає на фінальне зображення [2].

Загальний алгоритм перетворення SDR в HDR зображений на рисунку 1 і складається з чотирьох основних пунктів:

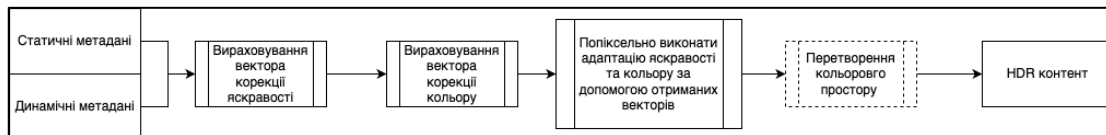


Рис. 1. Загальний алгоритм перетворення SDR в HDR

1. З наданих динамічних та статичних метаданих, вираховується вектор відображення яскравості;
2. З наданих динамічних та статичних метаданих, або підібраних метаданих, вираховується вектор корекції кольорів;
3. Попіксельно виконується операція корекції кольору та яскравості, використовуючи попередньо обчислені в пунктах 1 та 2 вектори;
4. Перетворити кольоровий простір в потрібний, якщо кольоровий простір вхідного SDR зображення не задовольняє вимоги. Наприклад, застосувати перетворення Rec. 709 у Rec. 2100.

На виході отримаємо піксель у HDR форматі з потрібним кольоровим простором.

Найбільш впливовим етапом алгоритму перетворення є вирахування вектору корекції яскравості, адже саме цей вектор впливає на якість деталей в темних та світлих частинах зображення.

Стандартний алгоритм вирахування вектору коефіцієнтів яскравості зображений на рисунку 2.

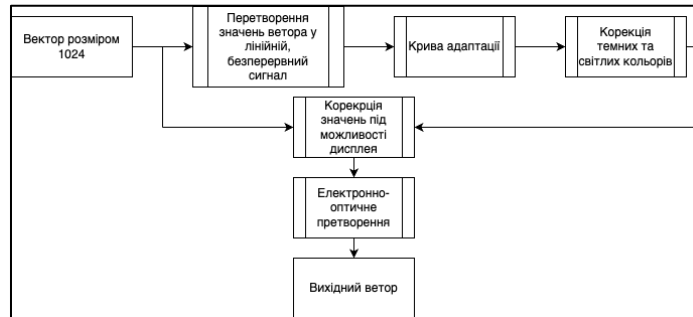


Рис. 2. Стандартний алгоритм вираховування вектору коефіцієнтів яскравості

Запропонована модифікація стандартного алгоритму полягає в тому, що до вираховування вектору корекції яскравості додається ще один етап – зворотна крива тонального відображення, яка зображена на рисунку 3 і виконується після застосування кривої адаптації. Це дозволяє контролювати корекцію світлих і темних тонів та граничні межі для кращої адаптації. Для використання кривої тонального відображення необхідні додаткові метадані:

- коефіцієнт підсилення темних тонів;
- коефіцієнт підсилення диференціального підсвічування;
- коефіцієнт ширини параболічного сегменту.

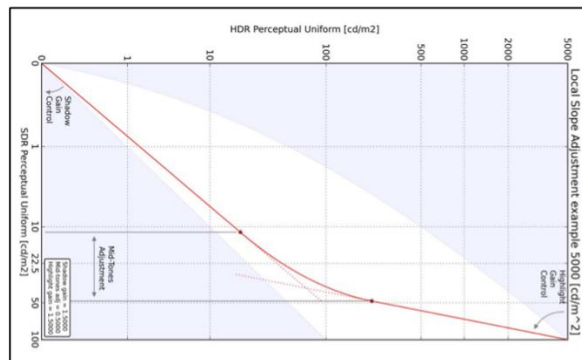


Рис. 3. Зворотна крива тонального відображення

Зворотна крива тонального відображення має три параметри:

1. Коефіцієнт підсилення темних тонів, він визначає яскравість темних ділянок зображення. Передається з метаданими. На рисунку 3 цей коефіцієнт позначений як *shadow gain control*;

2. Коефіцієнт підсилення диференціального підсвічування. Він контролює яскравість у світлих ділянках зображення, також передається разом з метаданими. На рисунку 3 цей коефіцієнт позначений як *highlight gain control*;

3. Ширина параболічного сегменту, відповідає за яскравість значної частини зображення, але це залежить від ширини параболи, тобто від значення коефіцієнта ширини параболічного сегмента, на рисунку 3 цей коефіцієнт позначений як *mid-tone adjustment*. Як правило цей коефіцієнт впливає на більшість ділянок зображення, окрім світлих та темних. Також цей коефіцієнт дозволяє регулювати граничні значення двох інших коефіцієнтів – коефіцієнт підсилення темних тонів та диференціального підсвічування. Після означеного вище етапу виконується стандартний етап корекції світлих та темних кольорів. Модифікований алгоритм зображено на рисунку 4.

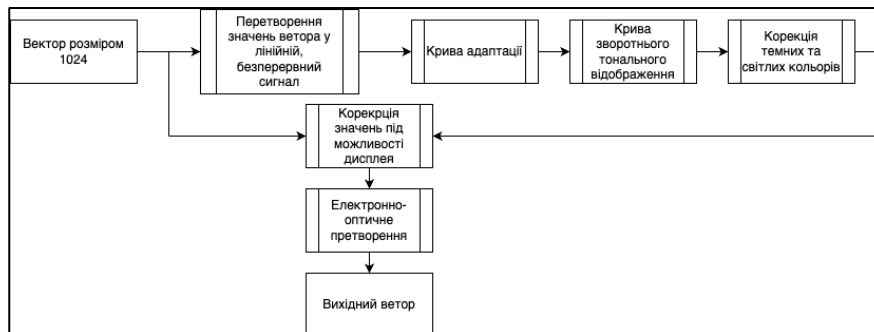


Рис. 4. Модифікований алгоритм трансформації SDR в HDR

Модифікований алгоритм дає змогу покращити відображення деталей на світлих та темних ділянках зображення. Порівняльне зображення трансформоване звичайним алгоритмом та модифікованим зображено на рисунку 5.

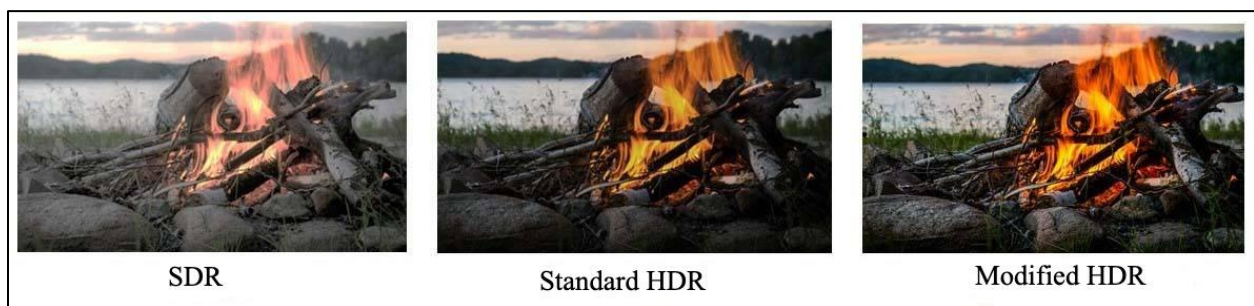


Рис. 5. Порівняння трансформованих зображень

Модифікований алгоритм трансформації SDR зображення в HDR реалізовано у вигляді консольної програми на мові програмування C++.

Висновки

В роботі проведено аналіз модифікації алгоритму перетворення SDR контенту в HDR та наведено порівняльний аналіз.

Запропоновано модифікацію з використанням зворотної кривої тонального відображення, що забезпечує корегування граничних меж світлих та темних тонів, використовуючи зворотну криву тонального перетворення, це дозволить корегувати трансформацію зображень під кожен фрейм окремо. Описаний алгоритм реалізовано у вигляді консольної програми на мові програмування C++.

Література

1. *E. Reinhard, W. Heidrich, P. Debevec.* High Dynamic Range Imaging 2nd edition, 2010
2. *S. Dai, M. Han, Y. Wu, and Y. Gong.* Bilateral back-projection for single image super resolution. In Proc. IEEE International Conference on Multimedia and Expo, 2007.– С. 537-540.
3. *Consumer Television Association.* Standard CTA-861.3-A HDR Static Metadata Extension, 2015

УДК 004.89

К.т.н., доцент Петрашенко А.В., студент Жовнірський Д.М.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

СПОСІБ РОЗПІЗНАВАННЯ КЛЮЧОВИХ ЕЛЕМЕНТІВ ОБЛИЧЧЯ ЛЮДИНИ

Abstract

Andrii V. Petrashenko, assoc. prof., PhD; Dmytro Zhovnirskiy, student
Method of recognizing key elements of person's face

This work is about creating ways to recognize key elements of a person's face. The possibilities and approaches of detecting drowsiness and fatigue with the help of modern technologies are considered and researched. A comparative analysis of the effectiveness of existing algorithms was carried out. The Python programming language was chosen for the implementation of the program. The face and contour point detection databases were developed using the OpenCV computer vision libraries. Landmarks for the main facial features are used using the Face Points pipeline solution created in MediaPipe and the Pattern recognition method. The AR formula was used to determine closed eyes and yawning.

Вступ

Зацікавленість системами, які вирішують технології розпізнавання ключових елементів обличчя людини дуже велика. Людське обличчя стало майже ідеальним ідентифікатором для передачі даних. Розпізнавання обличчя використовують у різних сферах діяльності: в охоронних системах, у кінематографії та 3D моделюванні. Багато соціальних мереж на основі розпізнавання обличчя роблять різні маски та фільтри [1].

Можливість розпізнавання елементів обличчя вирішують велике коло завдань і одне із таких завдань правильно визначати, що відображає людське обличчя. На основі елементів обличчя можна визначити її настрій та стан у якому вона знаходиться. Такі дані будуть необхідні у випадках, які пов'язані з небезпекою для життя, тому це питання так актуальне.

Актуальність розробки даного способу полягає в тому, що вона дозволяє вдосконалити та вивести на новий рівень взаємодію людини з машиною. Існують схожі системи контролю втоми. Новизна результатів отриманих у ході виконання роботи, полягає у підвищенні ефективності процесу класифікації втоми. Людина за комп'ютером не буде змушена постійно дивитися тільки у одному напрямі та знаходитись у одному

положенні, програма використовує кілька методів розпізнавання, для точнішого розпізнавання і контролю стану людини за роботою. Програма допомагає знизити можливість допуску помилок у роботі, зменшує навантаження на CPU за рахунок роботи по проміжках часу та допомагає зменшити навантаження на втомлені очі і зберегти здоров'я людини

Постановка задачі

Основною задачею є дослідження і аналіз існуючих алгоритмів, підходів та методів розпізнавання ключових елементів обличчя людини. Щоб реалізувати поставлену задачу було вирішено створити просту та корисну програму виявлення сонливості та втоми за допомогою MediaPipe.

Термінологія

OpenCV – це бібліотека алгоритмів комп'ютерного зору, обробки зображень та чисельних алгоритмів загального призначення з відкритим кодом Розробляється для Python, Java, Ruby, Matlab, Lua та інших мов [1].

MediaPipe - це фреймворк з відкритим кодом, який допомагає створювати мультимодальні конвеєри машинного навчання. Розробник може створити прототип, не заглиблюючись у написання алгоритмів та моделей машинного навчання, використовуючи наявні компоненти [2].

Face Points – це розроблений за допомогою MediaPipe конвеєр рішень. Складається із двох нейронних мереж, які розпізнають обличчя та дають орієнтири на елементи обличчя.

Шаблонний метод – це метод розпізнавання образів використовуючи базу шаблонів на основі характерних рис та ознак [3].

AR (Aspect Ration) – це формула співвідношення сторін, яка допомагає розрізнити стан очей та губ у реальному часі.

CPU – це центральний процесор комп'ютеру, який виконує функціональну частину, що відповідає за інтерпретації команд: обробку програмного коду, роботу операційної системи.

Опис способу

Щоб створити систему розпізнавання ключових елементів на обличчі людини потрібно:

1. Отримати доступ до камери
2. Алгоритм, який виявляє орієнтири на обличчі людини
3. Алгоритм, який буде на основі орієнтирів, визначати стан людини

Для отримання доступу до камери слід використовувати будь-яку камеру, яка підтримує потокове передавання, тобто веб-камера. Отримане зображення обличчя людини обробляється зібраним конвеєром рішень MediaPipe Face Points та Шаблонним методом розпізнавання на базі даних

отриманих за допомогою OpenCV на мові програмування Python. Далі використовується формула AR, яка буде визначати стан очей та губ у реальному часі.

Виявлення орієнтирів за допомогою зібраного конвеєра рішень MediaPipe Face Points

Використовуючи інструменти у MediaPipe створюємо конвеєрне рішення на основі двох моделей глибокої нейронної мережі, які працюють одночасно. Перша нейронна модель виконує роль детектору, який працює з повним зображенням та обчислює розташування обличчя. Друга використовується для 3D моделі орієнтирів обличчя і прогнозує приблизну 3D поверхню використовуючи регресію. Точно обрізані контури обличчя допомагають зменшити потребу у доповнених даних [2].

За допомогою конвеєру рішень Face Points отримується 68 орієнтирів на обличчі, які можна побачити на рис. 1.

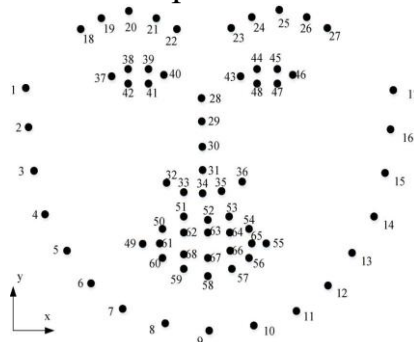


Рис. 1 Отриманні за допомогою конвеєру рішень Face Points 68 орієнтири на обличчі

Для визначення рівня сонливості людини, необхідно використовувати орієнтири, які вказують місце знаходження очей та губ.

- Ліве око: [37, 38, 39, 40, 41, 42]
- Праве око: [43, 44, 45, 46, 47, 48]
- Губи: [61, 62, 63, 64, 65, 66, 67, 68]

Ці точки не координати, вони виконують позначання позиції індексу у вихідному списку, який повертає сітку обличчя. Для очей вибрані орієнтири стають у порядку від P_1 до P_6 , а для губ від P_1 до P_8 , після чого вони використовуються у формулі AR.

Використання формули AR

Щоб виявити стан сонливості необхідно визначити у якому положенні протягом тривалого інтервалу часу знаходяться очі та губи.

Використовуються формули:

$$AR_{EL} = \frac{\|P_2^L - P_6^L\| + \|P_3^L - P_5^L\|}{2\|P_1^L - P_4^L\|} \quad (1)$$

$$AR_{ER} = \frac{\|P_2^R - P_6^R\| + \|P_3^R - P_5^R\|}{2\|P_1^R - P_4^R\|} \quad (2)$$

$$AR_{AVG} = \frac{AR_{EL} + AR_{ER}}{2} \quad (3)$$

$$AR_M = \frac{\|P_2 - P_8\| + \|P_3 - P_7\| + \|P_4 - P_6\|}{3\|P_1 - P_5\|} \quad (4)$$

Формули 1 та 2 повертають скалярні величини лівого та правого ока, які відображають рівень відкриття очей. Якщо $AR_E = 0$, це означає що око закрите, при іншому значенні око відкрите.

Формула 3 використовується для усереднювання двох значення AR_E . Середнє значення AR_{AVG} знаходиться у діапазоні $[0,0, 0,40]$ і швидко зменшується при закритті очей.

Формула 4 повертає скалярну величину губ, що відображає рівень відкриття рота. Якщо $AR_M = 0$, це означає що рот закритий, при іншому значенні рот відкритий.

Використання Шаблонного методу

Шаблонний метод на основі відповідності характерних рис забезпечує високу швидкість роботи і показує високі результати при розпізнаванні об'єктів, що підходить для роботи у реальному часі [3]. Метод використовує для розпізнавання базу даних отриману за допомогою OpenCV у випадку, коли конвеєрне рішення Face Points після кількох спроб не знаходить обличчя. Таке відбувається коли голова повернута у бік, піднята вгору, опущена вниз або людина взагалі відійшла від камери.

У випадку коли відбулося розпізнавання голови повернутої у бік або піднятої до гори програма повторить спробу розпізнавання через деякий час.

Якщо голова розпізнана опущеною, то програма повторить спробу через деякий час, але після кількох повторних розпізнавань у такому самому положенні з'явиться повідомлення з аудіо сигналом.

Коли нічого не вдалося розпізнати, після кількох повторів, програма призупиняє свою роботу і з'являється вікно у якому можна відновити роботу програми або вимкнути її.

Час роботи

Для визначення сонливості людини під час роботи немає необхідності, щоб програма відслідковувала стан очей та губ постійно. Створена програма працює з інтервалом по 5 хвилин на роботу та 5 хвилин у стані спокою. Ця модифікація допоможе зменшити навантаження на CPU.

З моменту коли нормальний стан очей або губ змінюється програма починає відлік часу, який потім порівнює із встановленою нормою і робить

висновок: закриті очі чи просто моргають, було позіхання чи людина розмовляла.

Якщо очі знаходяться у закритому стані більше встановленого часу, то з'являється повідомлення з аудіо сигналом.

У період коли програма працює, підраховується кількість позіхань та закриття очей і якщо значення більші за встановлені норми, з'являється повідомлення з рекомендацією закінчити роботу і йти відпочивати. Після третього повідомлення з'являються великі повідомлення, що закликають йти відпочивати.

Результати отриманої програми

Провівши ряд тестів було отримано статистику точності розпізнавання ключових елементів обличчя людини, яка відображається на рис. 2 та рівень навантаження на CPU на рис. 3.

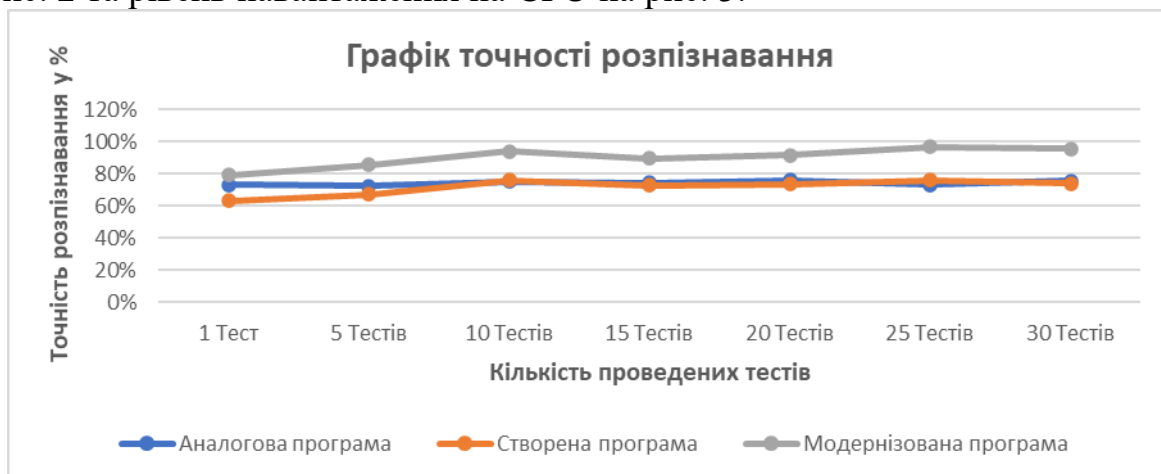


Рис. 2 Графік точності розпізнавання ключових елементів обличчя людини

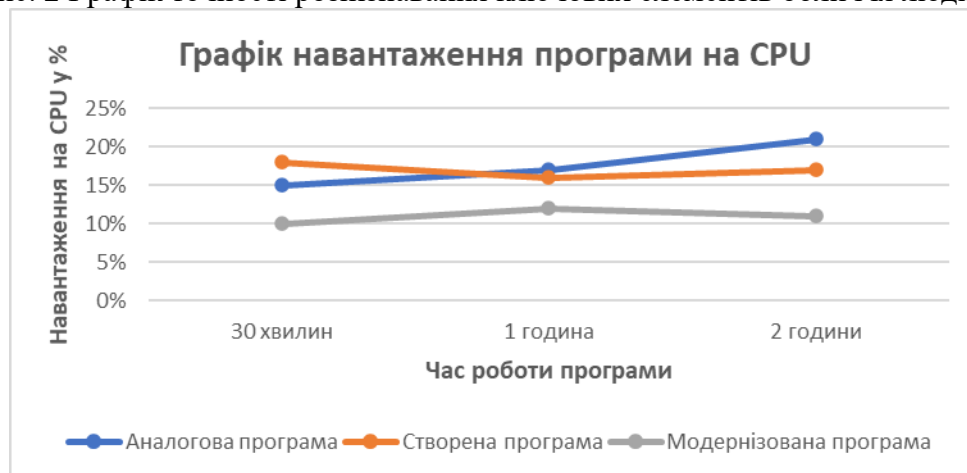


Рис. 3 Графік навантаження програми на CPU

Із графік точності розпізнавання видно, що розроблений конверс рішень MediaPipe Face Points і аналогова програма має схожу точність

розпізнавання. Додавши до конвеєра рішень Face Points використання Шаблонного методу рівень точності значно покращився, що допоможе точніше розпізнавати стан людини за роботою.

Проаналізувавши графік навантаження програми на CPU можна зробити висновок, що по інтервальна робота розпізнавання програми зменшує навантаження і підходить для використання у фоновому режимі роботи.

Висновки

Запропонований спосіб розпізнавання враховує різні положення голови відносно камери. Використовуючи мову програмування Python був створений конвеєр рішень MediaPipe Face Points, який дозволив на основі ключових елементів обличчя встановлювати сонливість людини.

Після тестів було вирішено використовувати додатково Шаблонний метод, що збільшило точність розпізнавання. Це дозволило менше відволікати людину від роботи при незвичайних ситуаціях і точніше визначати рівень сонливості людини.

Введення поінтервальної роботи програми допомогло знизити навантаження програми на CPU, що робить програму майже не помітною для користувача, який використовує програми з великим навантаженням.

Література

1. *Кутовець В.Я.* Розпізнавання образів : навчальний посібник / В. Я. Кутовецький// Миколаїв: Вид-во . ЧНУ ім. Петра Могили, 2017. – 420с.
2. *Kulli.* Introduction to MediaPipe: <https://learnopencv.com/introduction-to-mediapipe/> - March 1, 2022
3. *Жовнірський Д. М.* Комп'ютерні засоби розпізнавання об'єктів на зображенні: дипломний проєкт...бакалавра: 123 Комп'ютерна інженерія/Жовнірський Дмитро Миколайович. – Київ, 2021 – 70 с.

К.т.н., доцент Петрашенко А. В., магістрант Топіха М. В.

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

МЕТАМОРФНЕ ТЕСТУВАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ

Abstract

**Andriy V. Petrashenko, assoc. prof., PhD, Topikha Mykola, master student
*Metamorphic testing of neural networks for object recognition***

The proper testing is required for any software as it could leads for potential bugs and undefined behavior. This should be detected on development stage, otherwise it could leads into production program failure that will chain into negative user experience. To provide more software coverage it should be tested with different technics. This paper describes overall software testing, taking more deeply into machine learning based systems. Metamorphic testing is approach to overcome test oracle, where it is not always possible to define the correct outcomes for test system. The described metamorphic relations for object recognition could also be applied for other subject spheres.

Вступ

Необхідність в надійному програмному забезпеченні досить логічне та очевидно. Ніхто не бажатиме використовувати програмний продукт, якщо він не виконуватиме потреби споживача, або через наявні помилки в продукті може становити загрозу для пристрою взагалі. Ненадійне програмне забезпечення створюється з різних причин, серед них наступні: відсутність комунікації між командами розробки, через мірна складність продукту, не достатня кваліфікація розробників, нереалістичні або невиразні цілі проекту, брак часу для розробки, застарілі засоби для автоматизації тестування, не достатнє тестування продукту, не достатня кількість матеріалу для тестування та інші [1].

Постановка задачі

Завдання полягає у визначенні метаморфного відношення для нейронних мереж розпізнавання об'єктів та генерація подібних зображень з початкового для створення тестових випадків.

Тестування програмного забезпечення

Тестування програмного забезпечення – це процес перевірки програмного продукту на відповідність встановленим критеріям та відсутність дефектів в програмі за допомогою спеціально розроблених тестів. Існуючі методи тестування:

- автоматичне тестування;
- регресивне тестування;
- тестування чорної скриньки;
- тестування білої скриньки;
- тестування сірої скриньки;
- модульне тестування;
- регресивне тестування;
- інтеграційне тестування;
- функціональне тестування;
- системне тестування [2].

Традиційно для тестування програмного продукту необхідно знати точний результат виконання програми для його подальшого порівняння, але для систем, які основані на машинному навчанні не завжди можливо визначити кінцевий результат. Тестування таких систем відноситься до тестування чорної скриньки, коли не відомо про внутрішню структуру додатку, лише відомо, що при вхідних даних отримується певний результат.

Для тестування таких систем створюються великі набори даних, але це не завжди можливо, тому для вирішення цієї проблеми в 1998 році було запропоновано новий метод тестування – метаморфне тестування [3].

Метаморфне тестування

Метаморфне тестування – це тестування програмного забезпечення, яке націлене на подолання проблеми генерації нових тестових випадків та test oracle. Test oracle – це визначення чи узгоджуються фактичні результати з очікуваними, тобто тестовий випадок пройшов перевірку успішно чи ні.

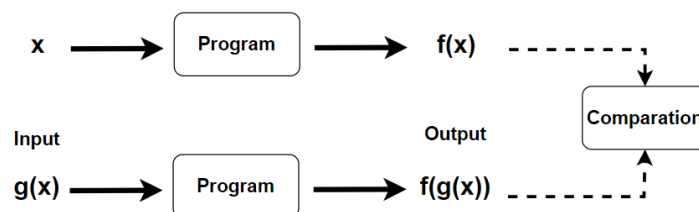


Рис. 1. Схема метаморфного відношення

Метаморфне тестування базується на метаморфних відносинах. Метаморфні відносини – це встановленні відношення між вхідними даними та результатами(рис. 1). Тобто для певних вхідних даних не ставляться певні вихідні дані. Для метаморфного тестування необхідно два або більше наборів вхідних даних з яких отримуються результати, після чого результати порівнюються, якщо поставлені метаморфні відношення виконуються, то тест пройдений успішно [4].

Наприклад, якщо в пошуковій мережі Google задати запит “кіт”, то отримаємо кількість результатів, в даному випадку 8,45млн результатів (рис. 2).

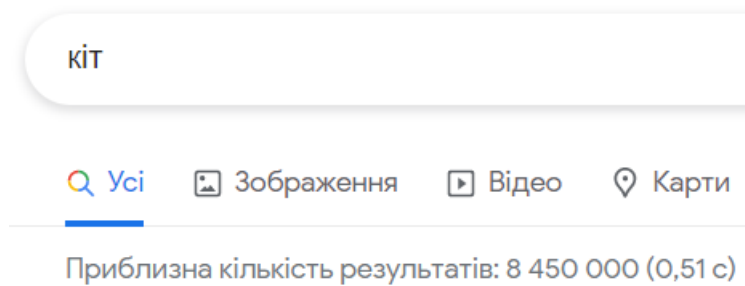


Рис. 2. Кількість результатів за запитом «кіт»

Якщо змінити запит на “чорний кіт” отримаємо зменшену кількість результатів (935тис., рис. 3), оскільки запит вже уточнений.

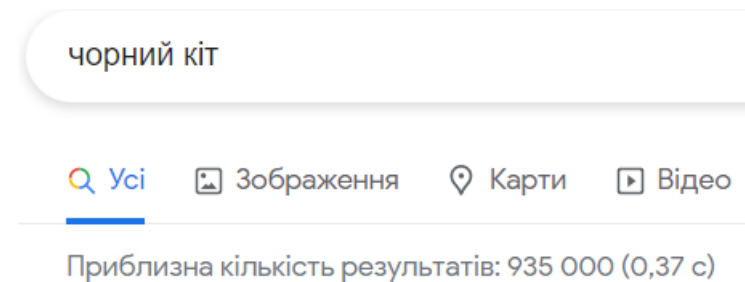


Рис. 3. Кількість запитів за запитом «чорний кіт»

В даному випадку метаморфне відношення – це те що уточнення запиту сприяє зменшенню результатів. Також другий запит це підмножина першого запиту, оскільки чорні коти входять до множини котів з першого запиту [4].

Метаморфне відношення для розпізнавання об'єктів

1. Кількість результатів повинно залишитися без змін, якщо перенести досліджуваний об'єкт на інше місце в межах зображення, без його модифікації.

2. Кількість результатів повинно зменшитися, якщо видалити досліджуваний об'єкт з зображення.

3. Кількість результатів повинно збільшитися, якщо додати до зображення досліджуваний об'єкт без перекриття існуючого.

4. Кількість результатів повинно зменшитися або залишитися без змін, якщо на зображення разом з досліджуваним об'єктом задати наступні зміни або їх комбінації, поворот на 15 градусів за часовою стрілкою, поворот на 15 градусів, затемнення області, висвітлення області, додавання нового об'єкту

Висновки

У статті сформульоване метаморфне відношення для розпізнавання об'єктів дозволяє більш ефективно тестувати системи, які основані на машинному навчанні. Забезпечує можливість збільшення тестових випадків за допомогою видозмінення існуючих зображень без витрачення додаткового часу на пошук нових зображень для тестування.

Література

1. Why Does Software Have Bugs? [Електронний ресурс]. – 2022 . – Режим доступу: <https://www.softwaretestinghelp.com/why-does-software-have-bugs/>.
2. 9 Types Of Software Testing In Software Engineering [Електронний ресурс]. – 2022 . – Режим доступу: <https://theqalead.com/test-management/types-of-software-testing/>.
3. T. Y. Chen, S. C. Cheung, S. M. Yiu, "Metamorphic testing: a new approach for generating next test cases," Department of Computer Science, Hong Kong University of Science and Technology, Tech. Rep. HKUST-CS98-01, 1998.
4. Metamorphic Testing of Machine-Learning Based Systems [Електронний ресурс]. – 2022 . – Режим доступу: <https://towardsdatascience.com/metamorphic-testing-of-machine-learning-based-systems-e1fe13baf048> .

К.т.н., доцент Потапова К.Р., студент Дадиверін В.В.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

ВИКОРИСТАННЯ ТА МОДИФІКУВАННЯ HAAR CASCADE FACE DETECTOR

Abstract

Potapova Kateryna, assoc. prof., PhD; Vitalii Dadyverin, student
About the organization and usage of Haar Cascade Face Detector

This post is a face detector breakdown. Positive and negative aspects of the Haar Cascade Face Detector system are indicated, and aspects that can be useful for improving the work are also suggested. Examples of improvements are given in photographs.

Вступ

Ми живемо в час безмежної цифровізації різного роду процесів суспільної життєдіяльності. З'явилась велика кількість технологій, які дають можливість посилити наш загальний рівень безпеки. Одною з таких технологій є детектор обличчя.

Неоціненно важливим є можливість розпізнавання обличчя людей як і в статичному, так і в динамічному станах, бо це дає змогу використовувати цю технологію у різних сценаріях забезпечення безпеки. Детектор обличчя ефективно використовувати для напівавтоматичного аналізу статичних зображень (фото) або для автоматичної системи розпізнавання у реальному часі задля швидкого реагування на можливий подразник.

Постановка задачі

Оцінка алгоритму Haar Cascade Face Detector та пропозиція аспектів, що можуть збільшити його ефективність.

Оцінка проводиться за такими параметрами:

- Визначення обличчя на фронтальному зображенні;
- Визначення обличчя на зображенні, де обличчя направлене донизу чи повернуте в сторону відштовхуюсь від фронтального положення.

Термінологія

FPS (англ. *frames per second*) — кількість кадрів в секунду на екрані монітора чи телевізора.

AdaBoost (скорочення від *Adaptive Boosting*) — алгоритм машинного навчання.

ОС – операційна система.

RGB (скорочено від англ. *Red, Green, Blue*) — адитивна колірна модель, що описує спосіб синтезу кольору, за якою червоне, зелене та синє світло накладаються разом, змішуючись у різноманітні кольори.

Виклад основного матеріалу

Haar Cascade Face – це алгоритм для виявлення об'єктів, в нашому випадку обличчя людей, на основі машинного навчання.

Машинне навчання полягає в тому, що каскадна функція проходить тренування на великій кількості позитивних та негативних зображень, де позитивні – це ті зображення, на яких присутній об'єкт для пошуку, а негативні – це ті, де цього об'єкту немає.

Це навчання відбувається за наступним алгоритмом :

- Даємо тестові зображення $(x_1, y_1), \dots, (x_n, y_n)$, where $y_i = 0, 1$ для негативного або позитивного результату відповідно;
- Ініціалізуємо значення ваги $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ для $y_i = 0, 1$ відповідно, де m та l – це кількість негативних та позитивних відповідно;
- Далі ми заходимо у цикл $t = 1, \dots, T$:
 1. Нормалізуємо вагу, $w_{1,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}}$, так що w_t – це розподіл ймовірностей;
 2. Для кожної ознаки j , треба «потренувати» класифікатор h_j , який обмежується використання однієї ознаки. Похибка оцінюється з урахуванням $w_t, \epsilon_{1,i} = \sum_i w_i |h_j(x_i) - y_i|$;
 3. Обираємо класифікатор h_j з відповідним найменшим значенням

помилки ϵ_t ;

4. Перераховуємо значення ваги: $w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$,
де $e_i = 0$, якщо x_i класифіковано правильно та $e_i = 1$,
якщо класифікована неправильно та в цьому випадку
$$\beta_t = \frac{\epsilon_t}{1 - \epsilon_t}$$

- Остаточний вирішальний класифікатор буде таким:

$$h(x) = \begin{cases} 1, & \sum_{t=1}^T \alpha_t h_t(x) \geq 0.5 \sum_{t=1}^T \alpha_t \\ 0, & \text{якщо протилежне першому} \end{cases}, \text{ де } \alpha_t = \log \frac{1}{\beta_t}$$

Переносячись від алгоритму у текст, як було сказано раніше навчання проводиться на двох типах зображень і в нашому випадку позитивними є зображення обличчя, а негативними – зображення без граней, це нам дає змогу підготувати класифікатор. Вся основна робота проходить за допомоги так званого «фільтра», використовуючи який вилучаються особливості із поточного зображення, також варто сказати, що ця система є подібною до згорткового ядра.

Ці фільтри представлені на рисунку 1.

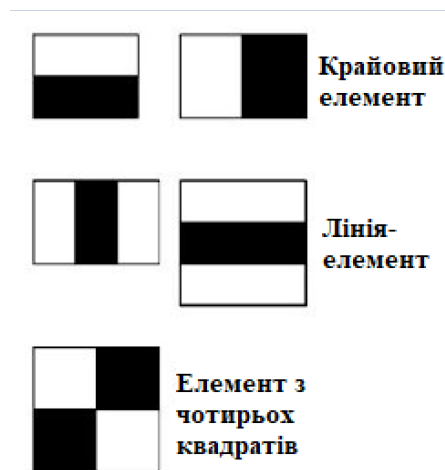


Рис. 1. Вигляд фільтрів Хаар

Представлені вище фільтри накладаються на зображення перевіряючи один сектор (вікно) і далі переходять на інший. Далі для кожного сектору іде обрахунок інтенсивності кожного пікселя за його відповідністю чорному та білому кольорам. Наступним кроком отримується значення витягнутої ознаки, що обраховується як різниця двох сум отриманих на минулому етапі. В найкращому випадку значення ознаки повинно показувати її релевантність.

Далі ми стикаємося з великою кількістю повернутих обрахунків з секторів, які треба оптимізувати. І тут на допомогу приходить концепція Integral Image – це особлива структура даних та алгоритм підрахунку значень у прямокутній підмножині сітки, який спровокує зменшення кількості обчислень для отримання потрібного результату.

На наступному кроці оптимізації ми повертаємося до класифікаторів і завдяки технології Adaboost створюється так званий «сильний» класифікатор з декількох «слабких», а об'єднання проводиться за спільними найкращими характеристиками.

Останнім кроком оптимізації виступає наша каскадність, яка існує для правильного групування ознак за різними стадіями класифікаторів і подальшого їх застосування по черзі у відповідних секторах. Це дає змогу відкинути відповідні сектори і застосування означень для них, а отже це створює прецедент більш економічного використання пам'яті.

Результати і обговорення

Визначаємо системні параметри на яких проводились дослідження:

- Процесор : Intel Core i7 7700HQ – 4 потоки.
- Оперативна пам'ять : 16 Гб.
- Відеокарта : NVIDIA GeForce GTX 1050, 4 Гб.
- ОС : Windows 10.
- Мова програмування : C++.

Каскадна модель - haarcascade_frontalface_alt.xml.

1. Наявна модель дуже добре визначає обличчя на фронтальному зображенні (9 з 10 визначень в поганому випадку), але тут постає питання у якості зображення, якщо воно стає геть жахливим (нижче за 144p), то якість визначення падає, але не критично, залишається 60-70% позитивних результатів.

2. Зображення, де обличчя направлене донизу чи повернуте в сторону визначається у 50-60% випадках і сказати, що це поганий результат не можна, бо каскадна модель орієнтована на здебільшого фронтально направлене обличчя. Завжди треба пам'ятати про кут, на який обличчя відхиляється від фронтальної орієнтації, бо якщо зовсім не видно однієї з трьох головних ознак (очі, губи, ніс), то можливість визначення кратно падає, адже модель на це не розрахована.

Висновки

В даному дослідженні виявлено недоліки представленого алгоритму, які не є критичними при використанні його для заявлених випадків. Варто зазначити що ключову роль грає модель на яку спирається ваш програмний продукт, аналізатор, тощо, бо якщо у регіоні застосування проживає переважно люди монголоїдної чи будь-якої іншої раси, то навчання чи вдосконалення моделі слід проводити на представниках саме цієї раси. Або треба додавати RGB-аналізатор, який буде брати визначені Наар алгоритмом зони і перевіряти чи справедливо для цього сектору умова $R > G > B$ щодо його кольорових характеристик, бо таким способом можна уникнути мінімум 50% помилкових детекцій. Доказ останньому можна побачити на рисунку 2, де червоні квадрати – це Наар, а зелені – RGB-аналізатор.



Рис. 2. Оптимізація помилок

Під час емпіричних досліджень найліпшого результату роботи показала модель з подвійною кількістю позитивів відносно негативів, порівнювалась вона з моделями:

- Позитивів і негативів однакова кількість;
- Негативів більше ніж позитивів у 2 рази.

Зменшити вхідну кількість зображень для навчання можна використовуючи `opencv_createsamples`, який дає можливість накладати позитиви на негативи і повертає позитивне зображення.

Різноманітність розпізнавання дуже сильно збільшує модель, на яку спиратиметься програмний продукт, тому в початковому використанні можна використовувати звичайні згорткові мережі для ліпшої оптимізації, бо, для прикладу, такого роду системи розпізнавання варто встановлювати на маленькі дрони для контролю безпекової ситуації, які повинні бути компактними та максимально фінансово вигідними.

Література

1. Paul Viola and Michael J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004;
2. Rainer Lienhart and Jochen Maydt. An extended set of haar-like features for rapid object detection. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 1, pages I–900. IEEE, 2002;
3. Atharifard A, Ghofrani S. Robust Component-Based Face Detection Using Color Feature. *Proceedings of the World Congress on Engineering*. 2011;II;
4. Singh SKr, Chauhan DS, Vatsa M, Singh R. A Robust Skin Color Based Face Detection Algorithm. *Tamkang Journal of Science and Engineering*. 2003;6(4):227-234.

УДК 519.688

К.т.н., доцент Потапова К.Р., студент Станкевич В.В.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

МОДИФІКОВАНИЙ АЛГОРИТМ ДЕЙКСТРИ ДЛЯ КЕРУВАННЯ АВАРІЙНОЮ ЕВАКУАЦІЄЮ В ПРИМІЩЕННІ В РЕЖИМІ РЕАЛЬНОГО ЧАСУ

Abstract

Kateryna R. Potapova, assoc. prof., PhD; Vladyslav Stankevych, student

Modified Dijkstra algorithm for real-time indoor emergency evacuation management

This paper proposes the improved emergency evacuation algorithm based on time equalization. To optimize the result of partitioning of evacuation zones, a method of group merging is proposed. The dedicated modification of Dijkstra algorithm and its time complexity are proposed. The evaluation of algorithm's performance in comparison to a regular staged evacuation planning algorithm is provided. The steps for further research are proposed as well.

Вступ

З розвитком глобальної урбанізації та ростом складності внутрішньої структури сучасних будівель, аварійна евакуація у приміщенні стає все більш комплексним питанням. У випадку аварійної ситуації та за відсутності ефективної евакуаційної стратегії існує вагома ймовірність стикнутись з великими майновими збитками та жертвами. За критичних обставин, усі люди у приміщенні мають бути евакуйовані, а відповідальні за це особи мають сформуванати найкращу евакуаційну стратегію за короткий час.

Як правило, легко викликати затори навколо аварійних виходів або в коридорах, коли в будівлі багато людей, що евакуюються. Стратегія поетапної евакуації конвертує час стагнації окремих груп, що евакуюються, у час затримки, і тим самим вирішує проблему заторів, але їй бракує оптимізації вибору безпечного виходу. На основі поетапної евакуації, задля підвищення ефективності використання виходів та зміцнення застосовуваності алгоритму у різних сценаріях екстренної евакуації у приміщенні, у даній статті пропонується алгоритм, що базується на вимірюванні часу.

Постановка задачі

Задача полягає у створенні алгоритму евакуації на основі алгоритму Дейкстри, який розподіляє людей, що евакуюються, по виходах у режимі реального часу, враховуючи потенційні конфлікти на шляхах таким чином, аби алгоритм міг бути застосований для будь якого поточного рівня щільності та рівномірності просторового розподілу людей у будівлі.

Термінологія

Алгоритм РППЕ – алгоритм розподіленого та поетапного планування евакуації. Даний алгоритм запозичив елементи евакуаційної стратегії розподілення для закріплення великої кількості людей, що евакуюються, за кожним аварійним виходом, та поетапної евакуаційної стратегії для обчислення шляху евакуації та часу затримки кожної групи.

ЗЧЕ – загальний час евакуації.

Опис алгоритму

Порядок евакуації відповідає принципу «перший зайшов, перший вийшов». Таким чином, час затримки кожної групи в одній зоні обчислюється відповідно до довжини шляху у порядку зростання. Формула часу затримки таким чином виглядатиме:

$$T_{i,d}^z = \begin{cases} 0 & , i = 1 \\ \frac{L_{i-1}-L_i}{V} + \frac{G_{i-1}}{F_{i-1}} + T_{i-1,d}^z & , i = 2, 3, \dots, n \text{ та } T_{i,d}^z > 0 \\ 0 & , i = 2, 3, \dots, n \text{ та } T_{i,d}^z \leq 0 \end{cases} \quad (1)$$

де $T_{i,d}^z$ – часова затримка групи i у зоні z , L_i – довжина шляху для групи i , V – швидкість руху групи, G_i – кількість осіб у групі i , i – група в зоні z , а F_i визначається з наступної формули:

$$F_i = \min \left\{ \{C_{r1,r2} | A_{r1,r2} \in R_i\} \cup C_e^i \right\}, \quad (2)$$

де $C_{i,j}$ – ємність $A_{i,j}$, $A_{i,j}$ – дуга між вузлами i та j , R_i – шлях групи i , C_e^i – ємність виходу e для групи i .

Стратегія одночасної евакуації використовує наступну формулу для розрахунку часу евакуації групи:

$$T_a^i = T_{i,rsp}^z + T_{i,t}^z + T_{i,q}^z, \quad (3)$$

де $T_{i,rsp}^z$ – час реакції групи i , $T_{i,t}^z$ – час евакуації (шлях) групи i , $T_{i,q}^z$ – час у черзі групи i .

Час евакуації кожної зони визначається максимальним часом евакуації груп цієї зони:

$$T^z = \max(T_{i,d}^z + T_{i,t}^z), i = 1, 2, \dots, n \quad (4)$$

Подібним чином, $ЗЧЕ$ визначається максимальним часом евакуації серед усіх зон будівлі:

$$ЗЧЕ = \max(T^z), z = 1, 2, \dots, m \quad (5)$$

$ЗЧЕ$ є одним із ключових факторів для визначення якості евакуаційної стратегії. Аби скоротити $ЗЧЕ$ необхідний обґрунтований метод розподілення з урахуванням щільності та просторового розподілу людей, що евакуюються, і пропускної здатності виходів.

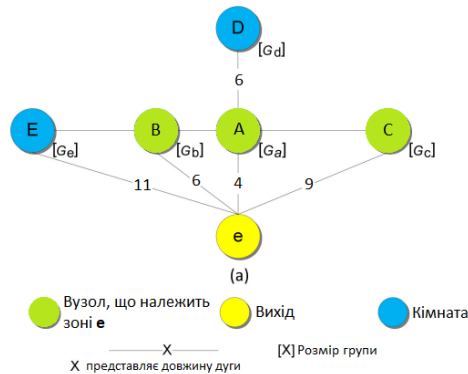


Рис.1 Схема розширення зони e

У процесі евакуаційного розподілення, алгоритм Дейкстри може викликати дисбаланс евакуаційних розділів при наявності відвітвлень. На рис. 1 вузол D – відвітвлення вузла A. За принципом «перший зайшов, перший вийшов», групи B і C розширяються зоною E перед вузлом D. Час евакуації групи D має бути поточним евакуаційним часом зони E. Метод групового злиття використовується для подальшого збільшення балансу розширення зон. Перед злиттям мають бути виконані наступні кроки:

1. Додати невихідні вузли до масиву M, та створити пустий масив B.
2. Додати віртуальний вузол, який з'єднується до усіх виходів, і встановити довжини усіх дуг в 0.
3. Віртуальний вузол вважається початковим «пошуковим» вузлом.
4. Використати алгоритм Тар'яна для пошуку точки зчленування у невідвіданих вузлах, потім помітити відвідані вузли.
5. Якщо точка знайдена, виконати наступний крок, інакше закінчити.
6. Окрім точки зчленування, додати всі вузли у двозв'язному компоненті до масиву B, повернутись до кроку 4.

Якщо в компоненті з подвійним зв'язком є вузли-приймачі, усі групи можуть обрати свої власні виходи, що належать до двозв'язного компонента, щоб вийти, не проходячи крізь вирізану вершину.

Загальні кроки для злиття груп такі:

1. Визначити поточний розширений вузол g та відповідний вихід E

2. Обрати групу i найближчу до виходу E у масиві V .
3. Якщо $T_e \geq \frac{D_i}{V}$, перейти до наступного кроку.
4. Обчислити час затримки групи i оновити час зайнятості виходу E з урахуванням часу евакуації групи i .
5. Видалити групу i з масивів M та V , повернутись до кроку 2.

Коли вузол A розширено, час евакуації групи A фактично є часом евакуації групи D . Подовження груп не збільшуватиме час евакуації поточної зони. Отже, можна використати час евакуації групи D як час зайнятості виходу E .

Алгоритм РППЕ використовує стратегію розподілення «збалансованої евакуації», засновуючись на завантаженості кожного вихода без урахування щільності та просторового розподілу людей в приміщенні та пропускної здатності виходів. Тому запропонований у статті алгоритм орієнтується на час зайнятості виходів. Даний алгоритм виконує розбиття та обчислення часу затримки одночасно, тоді як РППЕ робить одне за іншим. Детальний алгоритм перебігу описується далі.

Вхідні дані: внутрішньобудинкова маршрутна мережа, розташування та пропускна спроможність виходів, кількість і локація гру для евакуації.

Вихідні дані: шлях виходу та час затримки для кожної групи, ЗЧЕ.

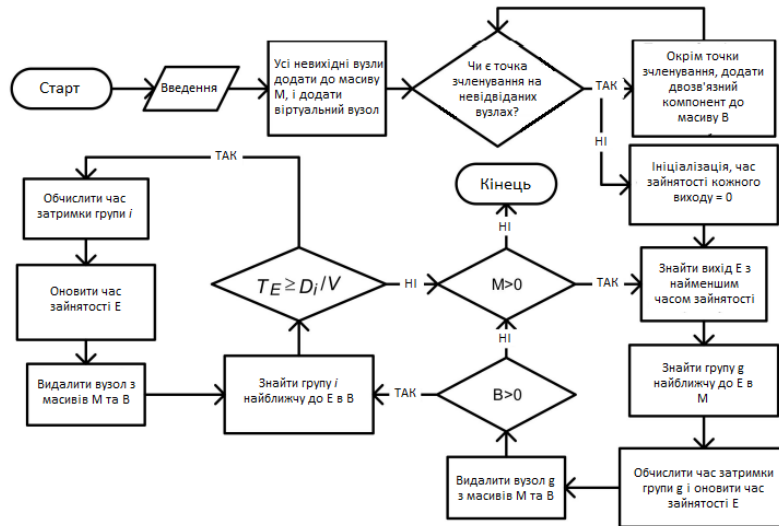


Рис. 2 Блок-схема запропонованого алгоритму

Алгоритм:

1. Операція над вузлами, отримати масиви M і V .
2. Ініціалізувати час зайнятості кожного виходу як 0.

3. Обрати вихід E з найменшим часом зайнятості, потім використати алгоритм Дейкстри для визначення групи g, найближчої до E в M.
4. Обчислити час затримки g, видалити g з масивів M і B, а потім оновити час зайнятості виходу E.
5. Якщо в масиві B є вузли, перейти до наступного кроку, інакше до 7.
6. Виконати об'єднання груп (як у попередньому алгоритмі)
7. Якщо в масиві M є вузли, перейти до кроку 3, інакше до наступного.
8. Обчислити ЗЧЕ

Висновки

Щільність і просторовий розподіл людей, що евакуюються, та пропускна здатність виходів мають вплив на евакуацію в приміщенні. На основі аналізу алгоритму поетапної евакуації, був запропонований новий алгоритм, що базується на вирівнюванні часу. Він може відмінно працювати за різних сценаріїв та значно скоротити ЗЧЕ.

У процесі поділу використовується метод групового злиття, що дає змогу вирішити недолік алгоритму РППЕ – нераціональне розширення евакуаційних зон. Крім того, використовується стратегія «відкладеного очікування», аби забезпечити відсутність конфліктів на шляхах до виходів. Дана стратегія робить евакуацію упорядкованою, безпечною та ефективною.

Даний алгоритм не враховує стан маршрутної мережі у реальному часі. Від пожежі, наприклад, дуги або вузли можуть стати непрохідними. Тому питання поєднання поетапної евакуації з динамічним плануванням шляхів потребує подальших досліджень.

Література

1. Gao, J.; Zhang, J.; He, J.; Gong, J.; Zhao, J.. Experiment and simulation of pedestrian's behaviors during evacuation in an office. *Physica A* **2020**, 545, 123749.
2. Kurdi, H.; Almulifi, A.; Al-Megren, S.; Youcef-Toumi, K. A Balanced Evacuation Algorithm for Facilities with Multiple Exits. *Eur. J. Oper. Res.* **2020**, 289, 285–296.
3. Hamacher, H. W., & Tjandra, S. A. Mathematical modelling of evacuation problems: A state of art. Fraunhofer (ITWM): Fraunhofer-Institut für Techno-und Wirtschaftsmathematik **2001**, 119, 44-252.

УДК 004.4

К.т.н., доцент Потапова К.Р., студент Фещенко І.О.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

ОСОБЛИВОСТІ АПАРАТНО-ПРИСКОРЕНОГО АЛГОРИТМУ РЕНДЕРИНГУ НА ОСНОВІ ТРАСУВАННЯ ПРОМЕНІВ

Abstract

Kateryna Potapova, assoc. prof., PhD; Ivan Feshchenko, student

Features of the hardware-accelerated rendering algorithm based on ray tracing

This paper concerns the task of hardware-accelerated rendering algorithm based on ray tracing without any additional tech supplies. At present, the ray tracing method requires the user to have significant computing power and non-standard hardware solutions, which are only partially being implemented by large manufacturers. This work suggests ways to improve this algorithm.

Вступ

Комп'ютерна графіка використовується у багатьох сферах життєдіяльності сьогодення. За її допомогою можуть створюватися комплексні графічні сцени або графічний інтерфейс користувача будь-якого програмного забезпечення. Одним з найважливіших є методи рендерингу зображення, серед яких метод трасування променів.

На даний час метод трасування променів потребує від користувача значних потужностей обчислювальної техніки і нестандартних апаратних рішень, які тільки частково починають реалізовуватися великими виробниками.

Аналіз існуючих рішень дозволяє зробити висновок, що доступний метод трасування променів має відповідати таким вимогам:

- робота в режимі реального часу;
- забезпечення достатньої кількості кадрів за секунду;
- відсутність потреби у додаткових апаратних рішеннях.

Постановка задачі

Оцінка апаратно-прискореного алгоритму рендерингу на основі трасування променів та запропонування шляхів його вдосконалення.

Виклад основного матеріалу

Метод трасування променів - метод візуалізації для створення зображення шляхом відстеження шляху світла як пікселів у площині зображення та моделювання ефектів його зустрічей з об'єктами на сцені. Ця техніка здатна надати високий ступінь реалізму, більше, ніж типові методи візуалізації, але за більших обчислювальних витрат та навантаження на графічний прискорювач. Це робить трасування променів найкращим чином придатним для випадків, де можна нехтувати відносно тривалим часом для рендерингу, наприклад, у нерухомих комп'ютерних зображеннях та візуальних ефектах у фільмах та телебаченні, але, як правило, більш погано підходять для програм у реальному часі, таких як відеоігри, де швидкість є критичною для рендерингу кожного кадру.

Описаний вище метод можна вдосконалити шляхом симулювання фізичної поведінки світла настільки близько до реального, наскільки це можливо. Такий підхід буде називатися трасуванням шляху, і він є просунутим окремим випадком традиційного трасування променів. Промінь відбивається від предметів до тих пір, поки повністю не буде поглинутий об'єктом або розсіється.

Принцип роботи алгоритму базується на фізичних законах поширення світла. Для цього кожен об'єкт на сцені має матеріал з наступними властивостями:

- Випромінювання енергії (англ. *emittance*) – кількість та довжина хвилі світла, яку випромінює об'єкт.
- Шорсткість (англ. *roughness*) – наскільки сильно промені розсіюються при зіткненні з об'єктом.
- Відбивна здатність (англ. *reflectance*) – кількість та довжина хвилі світла, яку відбиває об'єкт.
- Прозорість (англ. *transparency*) – відношення пропущеного крізь об'єкт світла до відбитого.
- Заломлення (англ. *refraction*) – показник заломлення світла.

На початку алгоритму для кожного пікселя на екрані випускається промінь. Якщо промінь перетинає об'єкт, то, в залежності від властивостей матеріалу, він може відбитися, заломитися або розсіятися.

У випадку відбивання є два його типи: дзеркальне та дифузне.

При дзеркальному відбиванні кут променя, який відбився, буде дорівнювати куту променя, який падає.

При дифузному відбиванні промені, що відбиваються, мають різний напрямок. Це явище пояснюється типом, що світло розсіюється на нерівних поверхнях.

Обчислення напрямку заломлених променів відбувається за допомогою закону Снеліуса. Записується наступним чином:

$$\frac{\sin\theta_1}{\sin\theta_2} = \frac{n_2}{n_1},$$

де θ_1 – кут падіння;

θ_2 – кут відбиття;

n_1 та n_2 – коефіцієнти заломлення двох середовищ.

Формули Френеля (або коефіцієнти Френеля) описують відбиття і пропускання світла (або електромагнітного випромінювання загалом) при падінні на межу розділу між різними оптичними середовищами.

Відповідно до ефекту Френеля, чим більше кут падіння променя, тим інтенсивніше відбитий промінь. Цю особливість можна використати для обробки ситуації, коли одночасно відбудеться відбивання і заломлення. Чим більше буде кут падіння променя, випущений з камери користувача на об'єкт, тим чіткіше буде відзеркалене зображення на поверхні цього об'єкта.

Для цього треба знайти коефіцієнт дзеркального відображення R , який можна використати як значення частини світла, що відбилася. Відповідно $(1 - R)$ буде означати заломлене світло.

Отже, коефіцієнт дзеркального відображення R можна знайти наступним чином:

$$F_p = \frac{\eta_2 \cos\theta_1 - \eta_1 \cos\theta_2}{\eta_2 \cos\theta_1 + \eta_1 \cos\theta_2},$$

$$F_s = \frac{\eta_1 \cos\theta_1 - \eta_2 \cos\theta_2}{\eta_1 \cos\theta_1 + \eta_2 \cos\theta_2},$$

$$R = \frac{1}{2}(F_p^2 + F_s^2),$$

де F_p - коефіцієнт відбиття р-поляризованого світла;

F_s – коефіцієнт відбиття s-поляризованого світла;

η_1 – коефіцієнт заломлення середовища, з якого світло падає;

η_2 – коефіцієнт заломлення середовища, у яке світло проходить;

θ_1 – кут падіння;

θ_2 – кут заломлення.

У даному випадку кут падіння пов'язан з кутом заломлення законом Снеліуса.

В комп'ютерній графіці для апроксимації вкладу фактора Френеля в дзеркальне відображення використовується апроксимація Шліка. Згідно з моделлю Шліка, коефіцієнт дзеркального відображення R може бути наближений наступним чином:

$$R = R_0 + (1 - R_0)(1 - \cos\theta)^5,$$

де θ - кут падіння, а R_0 обчислюється як:

$$R_0 = \left(\frac{\eta_2 - \eta_1}{\eta_2 + \eta_1} \right)^2.$$

Хоча апроксимація Шліка і вирішує проблему одночасного відбиття та заломлення світла, вона породжує іншу. Так як на один піксель кидається один промінь, то навіть при статичному зображенні колір пікселя на кожному наступному кадрі може змінюватися в залежності від результату апроксимації, що може виглядати як шум накладений поверх зображення.

Для вирішення цієї проблеми збільшимо кількість променів, які кидаються на один піксель. Після знаходження кольору для кожного променя необхідно знайти середнє арифметичне значення, яке і буде результуючим кольором.

Також використаємо акумулятивний рендеринг у дві текстури почергово з використанням попереднього зображення. Таким чином кожен кадр буде вміщувати в собі попередній і, при статичному зображенні, поступово буде ставати чіткішим.

Результати і обговорення

Проведено ряд тестувань, які показали можливість використання запропонованої імплементації алгоритму.

Алгоритм забезпечує рендеринг у реальному часі з достатньою для користувача кількістю кадрів за секунду без додаткових апаратних розширень.

Визначено оптимальне роздільної здатності екрану у 1280 на 720 пікселів через свою збалансованість між навантаженням на GPU та якістю результуючого зображення. Для запропонованого методу покращення зображення обрано оптимальне значення кількості променів у 6 одиниць

Висновки

В даному дослідженні розроблено апаратно-прискорений алгоритм рендерингу на основі трасування променів. Запропонований метод працює без потреби у додаткових апаратних розширеннях, що є його основною перевагою. Область застосування включає сферу комп'ютерної графіки, а саме інтегрування у графічні рушії для розробки відеоігор або кіноіндустрії.

Література

1. Eric Lengyel. Mathematics for 3D Game Programming and Computer Graphics, Third Edition. Видавництво: Cengage Learning, Inc. 2011.624 с.

2. Greg Humphreys, Wenzel Jakob, Matt Pharr. Physically Based Rendering: From Theory to Implementation, Third Edition. Видавництво: Morgan Kaufmann. 2016. 1266 с.

УДК 004.4

Студент Потурай М.В., к.т.н., ст. викл. Коляда К.В.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

РОЗПІЗНАВАННЯ ТЕХНІКИ В РЕАЛЬНОМУ ЧАСІ НА ВІДЕОМАТЕРІАЛАХ, ЩО ЗНЯТІ БПЛА

Abstract

Mykyta Poturai, student; Kostiantyn Koliada, senior lecturer, PhD

Detection of vehicles from drone footage in real time

This post is a breakdown of methods of real-time vehicles detection on a drone footage. Problem specific to vehicle detection and drone footage are investigated and possible solutions are suggested. Examples of working system are given.

Вступ

Розпізнавання військової техніки з відео потоку безпілотних літальних апаратів є дещо складнішим за класичну задачу розпізнавання, наприклад – розпізнавання автомобілів на автодорозі.

Наприклад, при розпізнаванні автомобілів, камера спостереження зазвичай встановлена статично, та всі автомобілі рухаються в одному напрямку, та мають однакову орієнтацію відносно камери. У той час як у відео з БПЛА камера може динамічно рухатись та спостерігати техніку з будь-якої сторони. Ще однією проблемою є якість відеопотоку.

Ціль роботи

Створення системи розпізнавання техніки на відео з БПЛА в реальному часі, що має виконувати наступні функції:

- Визначення техніки у відеопотоку з врахуванням специфічних для цієї галузі проблем.
- Система має бути достатньо ефективною для роботи на планшетах та ноутбуках середньої потужності

Термінологія

FPS (*frames per second*) – частота кадрів в секунду на екрані монітора чи телевізора.

YOLO (скорочення від *You Only Look Once*) – алгоритм машинного навчання.

SSD (скорочення від Single Shot Detector) – алгоритм машинного навчання.

CNN (скорочено від англ. *Convolutional Neural Network*) – згорткова нейронна мережа. Одна із популярних архітектур нейронних мереж.

Матеріали і методи

Сучасні алгоритми розпізнавання об'єктів за допомогою нейронних мереж поділяються на два основних підходи. Перший підхід – двоетапний. Спочатку за допомогою класичних алгоритмів визначається приблизне розташування об'єкта. Далі на визначених ділянках зображення застосовується згорткова нейронна мережа. Приклади таких алгоритмів – R-CNN та Faster R-CNN. Другий підхід – коли згорткова нейронна мережа застосовується одразу на всьому зображенні. Таки алгоритми називають однопроводовими, або алгоритмами одного перегляду. Приклади таких алгоритмів – SSD та YOLO.

Перевагою першого підходу є трохи більша точність, але вона досягається за рахунок використання більшої кількості машинного часу.

Для тренування нейронних мереж використовується набір даних, зібраний з відкритих джерел та соціальних мереж.

Алгоритм SSD

Мережа SSD складається з двох компонентів: хребет (backbone) і голова (head). Зазвичай хребет це – завчасно натренована мережа класифікації об'єктів. Наприклад ResNet або ImageNet, у якій відсутній повнозв'язний рівень класифікації. Завдяки таким маніпуляціям можна отримати глибоку мережу, що може вичленити з зображень семантичний зміст. Але в той же час зберігається й саме зображення, хоча і з меншим

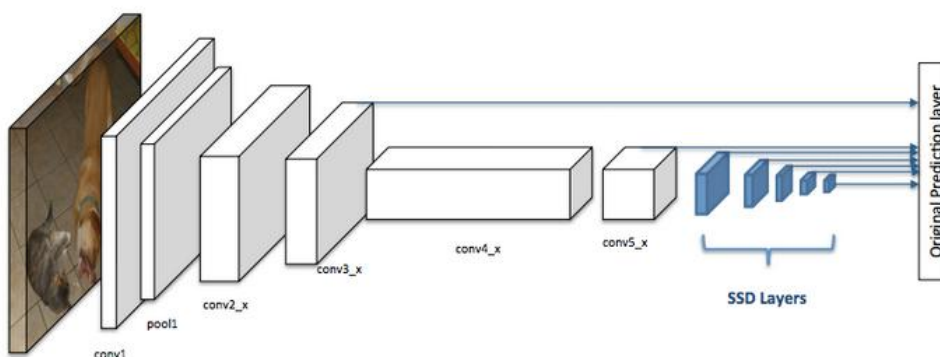


Рис. 1. Архітектура мережі SSD

розширенням.

Наприклад, при використанні ResNet34 в результаті виходить 256 карт особливостей з розширенням 7 на 7. Голова SSD – просто додаткові

згорткові слої, виходи яких інтерпретуються як обмежувальні рамки з класами. Візуально архітектура мережі SSD показана на рис. 1 [2].

Алгоритм YOLO

Алгоритм YOLO працює в 3 етапи. Спочатку зображення розділяється на залишкові блоки. На практиці це виглядає як накладання сіток різного розміру на зображення. Приклад такого розділення показаний на рис. 2.

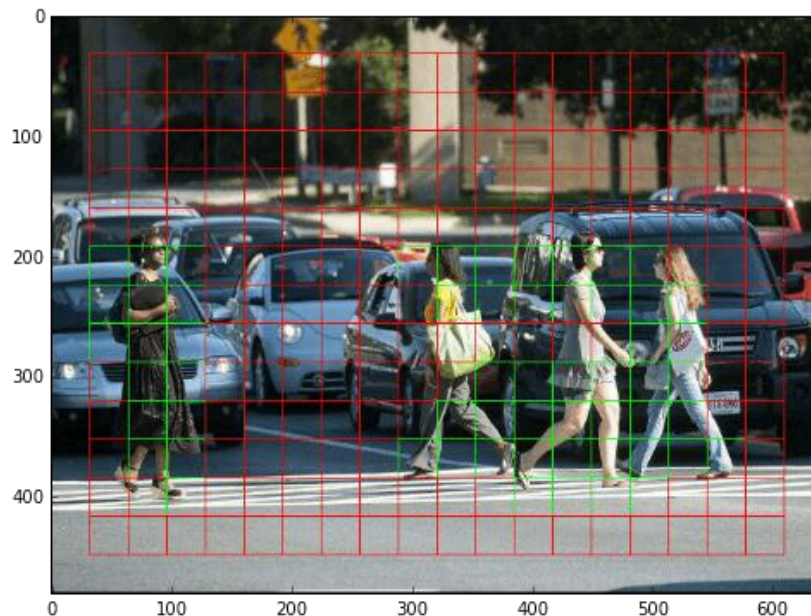


Рис. 2. Розділення зображення на блоки

Кожна клітинка сітки відповідає за розпізнавання частини зображення, яка знаходиться в ній. Зазвичай кожен регіон може розпізнати до 5 об'єктів в середині себе. Після початкового відпрацювання нейронної мережі, відбувається фільтрація. Під час неї відкидаються всі знахідки, які мають нижчу ймовірність бути точними ніж порогове значення [3].

В кінці обробки, клітини, що мають досить точну ймовірність мати в собі шуканий об'єкт групуються у обмежувальні рамки [4].

Результати і обговорення

Визначаємо системні параметри на яких проводились дослідження:

- Процесор: Intel Core i7 7700HQ – 4 потоки.
- Оперативна пам'ять: 16 Гб.
- Відеокарта: NVIDIA GeForce GTX 1050, 4 Гб.
- ОС: Windows 10.
- Мова програмування: Python.

Було проведено порівняльне дослідження декількох алгоритмів, метою якого було обрати найбільш придатний до поставлених задач. Результати дослідження наведені в таблиці 1.

Таблиця 1

Порівняння ефективності різних алгоритмів

Алгоритм	Середня точність	FPS
SSD	72.8	19
YOLO	68.4	45
Fast R-CNN	71.2	7

За результатами дослідження було прийнято рішення використовувати алгоритм YOLO, оскільки він має краще співвідношення швидкості роботи до точності. З використанням даного алгоритму була натренована нейронна мережа та розроблена система для визначення



Рис 3. Результат роботи системи

техніки на потоковому відео в реальному часі. Приклад роботи системи можна побачити на рис. 3.

Висновки

В даній роботі було проведено дослідження найбільш популярних алгоритмів машинного зору та розроблено систему розпізнавання техніки в реальному часі. Були розглянуті такі алгоритми як YOLO, SSD, та Fast R-CNN. Найбільш придатним до задачі алгоритмом виявився YOLO, на його основі було розроблено систему визначення техніки у реальному часі. Розроблена система є досить точною та продуктивною та може бути використана для покращення візуального сприйняття інформації у стресових ситуаціях а також ситуаціях коли відеосигнал є нестабільним через роботу засобів радіоелектронної боротьби. Також вона дозволяє швидше сприймати візуальну інформацію, що прискорює прийняття критичних рішень.

Література

1. Reinhard Klette. Concise Computer Vision. Springer. ISBN 978-1-4471-6320-6
2. Liu, Wei. "SSD: Single shot multibox detector". Computer Vision // ECCV 2016. European Conference on Computer Vision. Lecture Notes in Computer Science. Vol. 9905. pp. 21–37.
3. Redmon, Joseph. "You only look once: Unified, real-time object detection" // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
4. Wang, Chien-Yao. "Scaled-YOLOv4: Scaling Cross Stage Partial Network". // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

Д.т.н., професор Романкевич В.О., студентка Дорош К.А.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

ЗАХИСТ ІНФОРМАЦІЇ З ВИКОРИСТАННЯМ РОЗПОДІЛЕНИХ СИСТЕМ

Abstract

Vitaliy I. Romankevich, prof., PhD; Karyna Dorosh, student
Information security using distributed systems

This paper describes the usage of distributed systems for information security. The paper presents the idea how to store information in files with help of Blockchain.

Вступ

В сучасному швидкоплинному світі, особливо з появою Інтернету, усі взаємовідносини між людьми вирішуються з використанням новітніх технологій. Навіть раніше покупки, замовлення товарів відбувалися онлайн, а зараз, в час, коли соціальна дистанція та самоізоляція є одним з важливих факторів підтримання стабільної ситуації в країні, це просто невідкладна частина життя кожного.

В еру стрімкого розвитку онлайн-сервісів новітні технології потребують новітніх рішень, адже зростає і кількість зловмисників, які хочуть заробити на них – хакерів. Одним з рішень забезпечення більшої безпеки – децентралізація системи. В такому випадку, зловмисники не можуть зламати «центральний» вузол системи та заволодіти всією інформацією.

На даному етапі розвитку інформаційних технологія блокчейн поступово стає дуже важливим інструментом забезпечення безпеки, оскільки після створення блоків в ланцюгу, вони не підлягають змінні. А отже разом з децентралізацією системи та використанням криптографії блокчейн є технологією даними якої майже неможливо маніпулювати та фальсифікувати.

Постановка задачі

Суть задачі полягає у дослідженні існуючих засобів інформаційної безпеки та технології Blockchain, як засобу для забезпечення захисту

інформації, створенні програмного комплексу з підвищеним рівнем захисту персональної інформації та можливістю керувати доступом до даної інформації.

Термінологія

Розподілена система керування (Distributed Control System, DCS) — автоматизована система керування технологічним процесом, що характеризується побудовою розподіленої системи введення-виведення та децентралізацією обробки даних..

Blockchain – це peer-to-peer технологія, яка зберігає всю інформацію про всі транзакції, проведених учасниками системи.

Розумні контракти (Smart contract)— це просто програми, що зберігаються в блокчейні, які запускаються, коли виконуються заздалегідь визначені умови, які, як правило, використовуються для автоматизації виконання угоди, щоб усі учасники могли бути миттєво впевнені в результаті, без залучення будь-якого посередника або втрати часу.

Існуючі методи забезпечення інформаційної безпеки та переваги технології Blockchain

Існують різні методи забезпечення інформаційної безпеки:

- Програмні засоби, які включаються в ПЗ системи. Такі засоби виконують логічні функції захисту.
- Апаратні засоби – це пристрої (механічні, електронні, електронно-механічні), які вирішують задачі захисту інформації на апаратному рівні, вони можуть бути вмонтовані в системи обробки та передачі даних для внутрішнього захисту.
- Фізичні засоби – засоби, які використовуються для зовнішнього захисту техніки, на якій знаходяться інформаційні системи. Їх реалізація виконується на основі створення фізичних перешкод від несанкціонованого доступу конфіденційної інформації.
- Законодавчі засоби – це закони та інші нормативно-правові акти, за рахунок яких врегульовується правила використання конфіденційної інформації та встановлюється відповідальність за їх порушення.

Загальні методи забезпечення інформаційної безпеки активно використовуються на будь-якій стадії управління загрозами. До таких стадій належать: прийняття рішення по визначенню області та контексту інформаційної загрози і складу учасників процесу протидії; ухвалення загальної стратегії і схеми дій в політичній, економічній, соціальній та інших сферах життєдіяльності; забезпечення адекватного сприйняття загрози та небезпеки у нижчих організаційних ланках системи управління; виділення необхідних політичних, економічних, соціальних,

адміністративних і організаційних ресурсів, достатніх для реалізації програми відбиття інформаційної загрози і збереження сталого розвитку інформаційних ресурсів системи управління: трансформації результатів оцінки ризиків у відповідну політику безпеки, включаючи національну [1].

Головними перевагами технології Blockchain є прозорість та відкритість, децентралізованість та розподіленість, незмінність, саме за допомогою них забезпечується безпека інформаційних систем.

У блокчейні децентралізація має на увазі передачу контролю та прийняття рішень від централізованого суб'єкта до розподіленої мережі. Децентралізовані мережі прагнуть знизити рівень довіри, який учасники повинні виявляти одна до одної, та стримувати їх здатність здійснювати владу або контроль один над одним таким чином, що погіршує функціональність мережі. А таким чином для зловмисників стає майже неможливим втручання або незаконна зміна інформації, оскільки для цього необхідно будет змінити інформації в кожному вузлі мережі.

Принцип роботи блокчейн – прозорість та незмінність здійснених операцій в системі. Під прозорістю мається на увазі відкритість, комунікативність та підзвітність. Завдяки ній учасники мережі можуть побачити будь-які зміни, які відбуваються в мережі. Блокчейн зберігає інформацію про всі транзакції, що мали місце з початку запуску технології. Дані публічної системи блокчейн зберігаються в доступі для всіх, тому кожен може легко перевірити інформацію.

Незмінність даних в блокчейн означає, що дані не можуть бути змінені або вилучені ніким після створення. Завдяки використанню хеш-функцій, які використовуються для шифрування блоків при будь-якій зміні інформації в будь-якому з попередніх блоків, зміниться хеш для всього ланцюга і це буде видно усім учасникам системи. Для атаки необхідно перерахувати хеші всіх блоків та переконати більшу частину вузлів, що даний ланцюг є істинним. Враховуючи складність доступу до кожного блоку, здійснити кібератаку стає майже неможливим.

Вибір платформи для створення децентралізованих онлайн-сервісів

Ethereum – це глобальна платформа для створення децентралізованих онлайн-сервісів на базі блокчейну з відкритим кодом [2]. Дана платформа децентралізована та функціонує як єдина програмне середовище, як віртуальна операційна система, робота якої підтримується серверами та вузлами – комп'ютерами учасників платформи.

До появи Ethereum розробка блокчейн-додатків була долею обмеженого числа ентузіастів криптографії та програмування.

Використовуючи смарт-контракти на платформі Ethereum можна отримати наступні переваги [3].

- Автономність та економія. Смарт-контракти не потребують брокерів або інших посередників для підтвердження угоди; таким чином, вони виключають ризик маніпуляцій з боку третіх осіб. Більше того, відсутність посередника в смарт-контрактах призводить до економії коштів.
- Резервне копіювання. Всі дані, що зберігаються на блокчейні, дублюються кілька разів; таким чином, оригінали можуть бути відновлені у разі втрати даних.
- Безпека. Смарт-контракти шифруються, а криптографія захищає всі документи від злому.
- Швидкість. Розумні контракти автоматизують завдання за допомогою комп'ютерних протоколів, заощаджуючи години різних бізнес-процесів.
- Точність. Використання смарт-контрактів призводить до усунення помилок, які виникають через ручне заповнення численних форм.

Хоча використання розумних контрактів має безліч переваг, як і будь-яка технологія, яка ще не досягла повної зрілості, є фактори, які можна розглядати як недоліки. По-перше, на даний момент не існує великого регулювання щодо того, як працюють смарт-контракти. З часом прийде більше управління, але тим, хто впроваджує нові проекти, потрібно бути обережними, інвестуючи в цю технологію. Також все ще існує велика залежність програмістів від створення розумних контрактів та виправлення будь-яких помилок. Там, де технологія блокчейну є незмінною, може виникнути недостатня гнучкість у зміні контрактів, коли це потрібно. Наприклад, можуть існувати постійні додатки, які потрібно застосовувати до умов блокчейну.

Висновки

У роботі було розглянуто існуючі методи забезпечення інформаційної безпеки. Також було запропоновано технологію Blockchain як засіб для забезпечення додаткової інформаційної безпеки, проведено дослідження з його переваг, такі як: прозорість та незламність даних, що є основою інформаційної безпеки. Було досліджено тему смарт-контрактів, як засіб для створення децентралізованих онлайн-сервісів, вивчено їх переваги та недоліки.

Література

1. Методи забезпечення інформаційної безпеки – [Електронний ресурс]. – Режим доступу: <http://politics.ellib.org.ua/pages-8288.html>.
2. Ethereum – глобальна платформа с открытым кодом для децентрализованных приложений – [Електронний ресурс]. – Режим доступу: <https://ethereum.org/ru>.
3. Smart Contrats: The Blockchain Technology That Will Replace Lawyers – [Електронний ресурс]. – Режим доступу: <https://blockgeeks.com/guides/smart-contracts/>

УДК 004.05

д.т.н., професор Романкевич О.М., студент Мережко І.П.

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

МОДИФІКАЦІЯ СПОСОБУ ФОРМУВАННЯ GL- МОДЕЛЕЙ ДЛЯ СКЛАДНИХ СИСТЕМ

Abstract

Romankevich O.M, prof., PhD; Ivan Merezhko, student

About one way to modify GL-models

This work concerns the problem of modifying the construction of the GL-model. The construction of basic and non-basic GL-models was investigated. Modifications to the construction of GL-models for models with a certain feature are proposed. The correctness of the incentive was checked on an example.

Вступ

Існує багато методів розрахунку надійності відмовостійких багатопроцесорних систем (ВБС), які використовуються у керуванні доволі важливих об'єктів (літаки, ракети, кораблі, машини) [1]. Вони всі дуже складні, особливо у випадку, коли досліджувана система веде себе по-різному при однаковій кількості відмов. Ще більшу складність мають системи, де окрема частина процесорів може брати на себе роботу інших під час їх відмови. Сучасні ВБС можуть складатися з багатьох систем та підсистем, які також містять велику кількість елементів. В даній роботі досліджується алгоритм будівництва GL-моделей [2] для складних систем, в яких певні процесори можуть замінити роботу інших, але не навпаки.

Постановка задачі

Задача полягає в дослідженні та оптимізації алгоритму будівництва GL-моделей для складних систем, в яких певна частина процесорів може брати на себе роботу інших, які вийшли з ладу.

Результати дослідження та оптимізації

GL-моделі можна використовувати для підрахунку надійності будь-якої системи, у тому числі відмовостійкої. Розрахунок проводиться статистичним методом на базі експериментів з моделями. GL-моделі

представляють собою неорієнтовані графи, і кожному ребру відповідає булева функція, де змінними є стани процесорів з можливими значеннями 1 (процесор працює) та 0 (процесор вийшов з ладу). Зв'язність графа відповідає працездатності системи. В [3] була запропонована МВР модель, яка корисна тим, що втрачає одне ребро при появі допустимого числа відмов і два при виявленні 1-го зайвого. Цей метод дозволяє швидко аналізувати базові GL-моделі виду $K(m, n)$, де n – кількість процесорів, а m – максимально допустима кількість відмов. Якщо ж система складається з декількох підсистем, моделі будуються для кожної окремо, а потім поєднуються в одну. Для небазових систем існують методи модифікації GL-моделей, які направлені на збереження адекватності функціонування моделі поведінці системи у потоці відмов.

Мінімізація побудови та дослідження GL-моделей має практичний сенс, бо чим швидше можна буде отримати результат одного дослідження, тим більше може бути елементів статистичного дослідження цієї моделі, що збільшує точність підрахунку ймовірності безвідмовної роботи системи за даний час.

При побудові складних ВБС доволі часто можна зустріти випадки, коли при відмові однієї кількості процесорів система веде себе по-різному, або коли тільки деякі процесори можуть брати на себе задачі інших компонентів системи. Саме цей випадок розглядається у доповіді.

Для ілюстрації розглянемо самий простий варіант системи, який підпадає під вимоги:

- системи складається з двох підсистем А та В;
- кожна підсистема є базовою (виходить з ладу при появі певної кількості помилок), вигляду $K(1, n)$;
- процесори підсистеми В можуть брати на себе функції процесорів підсистеми А, але не навпаки.

Можна виділити наступні умови виходу з ладу усієї системи загалом:

- відмова 2-х чи більше процесорів підсистеми В;
- відмова 2-х процесорів у підсистемі А та 2-х процесорів у підсистемі А;
- відмова більше 2-х процесорів у підсистемі А;
- відмова 2-х процесорів у підсистемі А та одного процесора у підсистемі В.

Всі інші комбінації не призводять до відмови працездатності всієї системи, тобто модель втрачає не більше одного ребра – граф залишається зв'язним.

Дослідивши всі умови, можна зробити висновок – дану систему можна представити наступним чином: для В так само як МВР-модель 1-

відмовостійкої ВБС, для підсистеми А, як для МВР-моделі 2-відмовостійкої системи.

Можливі такі випадки, коли кожна підсистема має свою унікальну, максимально допустиму кількість відмов. Для такого випадку можна використовувати наслідки з твердження:

- якщо система А має відмовостійкість m_a , то під час побудови моделі для всієї системи кожна така підсистема формує реберні функції як для $(m_a + 1)$ -відмовостійкої МВР моделі;
- для системи В- як для 1-відмовостійкої МВР моделі;

Перевіримо алгоритм побудови на простому прикладі, коли система складається з двох підсистем А та В. Підсистема А являється К(3, 7) системою, а В - К(1, 5). Також, якщо вийшло з ладу 4 процесора у системи А, але всі процесори підсистеми В працюють, - система загалом працює. Розроблено алгоритм та програма будівництва відповідної GL-моделі.

Функції GL-моделі будуть мати вигляд:

$$f_1 = a_1 \vee a_2 \vee a_3 a_4$$

$$f_2 = a_1 a_2 \vee a_3 \vee a_4$$

$$f_3 = (a_1 \vee a_2)(a_3 \vee a_4)(a_1 a_2 \vee a_3 a_4) \vee a_5 a_6 a_7 a_8$$

$$f_4 = (a_5 \vee a_6)(a_5 a_6 \vee a_7 a_8)(a_7 \vee a_8) \vee a_1 a_2 a_3 a_4$$

$$f_5 = a_5 \vee a_6 \vee a_7 a_8$$

$$f_6 = a_5 a_6 \vee a_7 \vee a_8$$

$$f_7 = b_1 \quad f_8 = b_2 \quad f_9 = b_3 \quad f_{10} = b_4 \quad f_{11} = b_5$$

Висновки

Було досліджено будівництва GL-моделей для складних систем, в якій існує підсистема з універсальних процесорів, що мають можливість брати на себе функції процесорів інших підсистем під час їх відмови, але не навпаки. Було запропоновано модифікацію алгоритму побудови GL-моделі для таких систем, що дозволяє вирішати задачу проектування в рамках однієї циклічної моделі замість розбиття на підмоделі.

Література

1. Avizienis, J. Laprie, B. Randell, Dependability and its threats: a taxonomy, Building the Information Society (2004) 91–120.
2. Romankevich, A., Feseniuk, A., Maidaniuk, I., Romankevich, V. Fault-tolerant multiprocessor systems reliability estimation using statistical

experiments with GL-models // *Advances in Intelligent Systems and Computing*.– 2019.– Vol. 754, P. 186-193

3. Романкевич А.М., Романкевич В.А., Майданюк И.В. Граничные оценки числа рёбер GL-моделей поведения отказоустойчивых многопроцессорных систем в потоке отказов // *Электронное моделирование*.–№1, т.30.– 2008.– С.59-70

Д.т.н., професор Романкевич О.М., студент Горба Д.О.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

АЛГОРИТМ ФОРМУВАННЯ ГАМІЛЬТОНОВИХ ЦИКЛІВ ДЛЯ ГЕНЕРАТОРІВ ПСЕВДОВИПАДКОВИХ ЧИСЕЛ З РІЗНОЮ ВАГОЮ

Abstract

Oleksii M. Romankevich, prof., D.Sc.; Dmytro Horba, student
Algorithm for generating Hamiltonian cycles for generators of pseudo-random numbers with different weights

The paper analyzes the algorithm for finding Hamiltonian cycles for the generator of all pseudo-random binary vectors with weights $p_1 \dots p_k$ based on the standard scheme. The article considers the method of forming vectors, their distribution into groups and the subsequent formation of Hamiltonian cycles, taking vectors as the vertices of graphs as a basis

Вступ

Випадкові числа використовуються давно і досить широко. Нагадаємо деякі області їх застосування:

- Соціологічні та наукові дослідження. Підготовка випадкових вибірок при зборі даних, опитуванні думок або в дослідженні фізичних явищ з випадковим вибором результатів експериментів.

- Моделювання. У комп'ютерному моделюванні фізичних явищ. Крім того, математичне моделювання використовує випадкові числа як один з інструментів чисельного аналізу.

- Криптографія та інформаційна безпека. Випадкові числа можуть використовуватися в тестуванні коректності або ефективності алгоритмів і програм. Багато алгоритмів використовують генерацію псевдовипадкових чисел для вирішення прикладних завдань (наприклад, криптографічні алгоритми шифрування, генерація унікальних ідентифікаторів та ін.).

Теорія побудови генераторів псевдовипадкових чисел (ГПСЧ) глибоко і добре вивчена. Однак у тій же інженерній практиці іноді виникає необхідність у генерації якихось спеціальних послідовностей певної

підмножини векторів довжини n . Прикладом може бути область тестування цифрової апаратури.

Постановка задачі

При будь-яких величинах n , а також $p_1, p_2 \dots p_k$, де $1 < k \leq n$, $p_{i+1} = p_i + 1$, $i = 1 \dots k$ та $p_1 < n$ сформувані вектори довжини n , з вагою відповідною до можливих значень p і розподілити їх на групи згідно їхньої ваги послідовно, після чого знайти гамільтоновий цикл для графа, вершинами якого будуть сформовані вектори.

Термінологія

Криптографія – наука про математичні методи забезпечення конфіденційності і автентичності.

Генератор псевдовипадкових чисел – алгоритм, що генерує послідовність чисел, елементи якої майже незалежні один від одного і підкоряються заданому розподілу.

Опис алгоритму

Для набору параметрів n та $p_1 \dots p_k$ сформуємо вектори та розподілимо їх на групи в циклічній послідовності згідно їхньої ваги. На рисунку 1 (Даний рисунок, можна вважати графом, якщо трактувати кожен його вектор в якості вершини графа) описано приклад для генератора з параметрами $n=5$, $p=1,2,3$ та групами А – вектори з вагою 1; С1, С2 – з вагою 2; В1, В2 – з вагою 3.

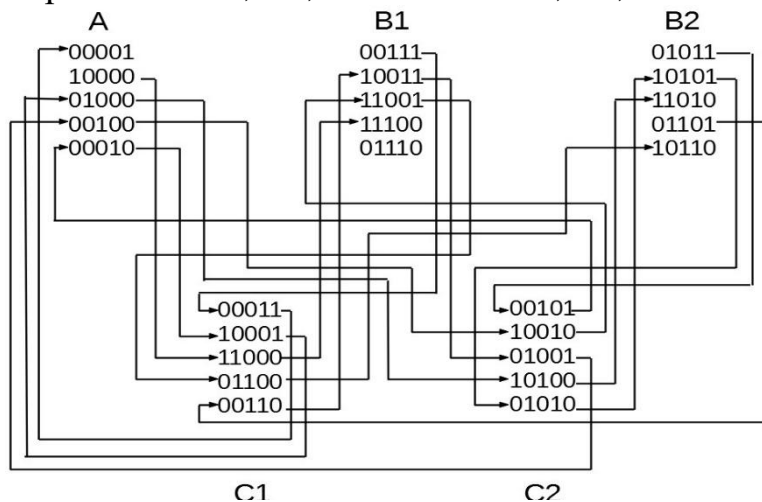


Рис.1. Усі вектори для набору $n=5$, $p=1,2,3$ та переходи між ними

Стрілками на малюнку позначені можливі переходи з блоку в блок при підвищенні або зниженні ваги вектора. Вектори позначаються відповідно до

розташування: наприклад, вектор 11010, розташований у блоці В2 третім зверху, позначається як В2₃. Стрілки, що позначають переходи з вектора до вектора по простому зсуву, не позначені для зручності.

Задача пошуку гамільтонового шляху (Гш) або гамільтонового циклу (Гц) у загальному випадку досить складна, проте аналізований граф має низку властивостей, що полегшують її розв'язання. Серед них на основі аналізу векторів та переходів між ними (узагальнюючи на довільні значення n та p) можна відзначити такі властивості:

1. Якщо має місце перехід із вектора a_i блоку А у вектор b_j блоку В, то має місце і перехід із вектора b_{j-1} у вектор a_{i+1} тих же блоків (складення та віднімання за модулем n). Наприклад, на рис. 1 є перехід $A_2 \rightarrow C1_3$. А це означає, що перехід $C1_2 \rightarrow A_3$ також має місце.

2. Кількість вхідних та вихідних стрілок у будь-якому блоці збігається. Цю властивість легко довести, використовуючи попередню властивість.

3. Властивість 2 має місце для будь-якої групи векторів.

4. Можливий послідовний перехід з будь-якого вектора в будь-яку іншу множину. Будь-який такий процес знаходить свій відбиток на графі G послідовністю переходів.

5. Будь-який вектор (вершина графа) може мати (включаючи не зазначені на малюнку стрілки по переходах з вектора у вектор за простим зсувом) одну або дві вхідні стрілки і одну або дві вихідні. Третьою бути не може.

Властивість 4 дозволяє стверджувати, що граф G є зв'язним.

На базі описаних особливостей, було розроблено загальний алгоритм побудови гамільтонових циклів у графах, що розглядаються, за допомогою якого, можна побудувати гц починаючи з будь-якої вершини графа, де вершинами є вектори на рис. 1. Якщо починати шлях з вектора $C2_5$ у наведеному на рис. 1 прикладі, то Гц вийде наступним.

$C2_5 \text{ } B2_{234} \text{ } C1_5 \text{ } B1_{23451} \text{ } C1_1 \text{ } A_{12345} \text{ } C1_{234} \text{ } B2_{51} \text{ } C2_{12345}$.

Висновки

В роботі розглядається можливість пошуку різних гамільтонових циклів у графі, в якості вершин якого є двійкові вектори сформовані згідно заданих параметрів. Алгоритм досить простий і може бути застосований при формуванні функцій нелінійного зворотнього зв'язку у ГПСЧ за узагальненою класичною схемою. Існують різні області застосування гамільтонових циклів, наприклад при побудові нестандартних схем

генераторів псевдовипадкових чисел, у криптографії гц використовують у системі протоколів з нульовим розголошенням.

Література

1. *Gill, A., Linear sequential circuits / Arthur Gill.* 1966, New York : McGraw-Hill.
2. *Gill, A., Introduction to the theory of finite-state machines / Arthur Gill.* 2002, New York : McGraw-Hill.
3. *A. Fúster-Sabater, P. Caballero-Gil.* Linear solutions for cryptographic nonlinear sequence generators // *Physics Letters A* Vol. 369, Is. 5-6, 1 Oct. 2007, pp. 432-437

УДК 004.627

К.т.н., доц. Тарасенко-Клятченко О.В., студентка Білоха А.К.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

ОПТИМІЗАЦІЯ АЛГОРИТМУ СТИСНЕННЯ ДАНИХ ГАФФМАНА

Abstract

Oksana I. Tarasenko-Klyatchenko, assoc. prof., PhD; Anna Bilokha, student
Optimization of the Huffman data compression algorithm

This paper concerns the task of comparing the standard Huffman data lossless compression method and its optimization with the use of idle bits. Also, the work of the standard algorithm is explained, as well as the basic principles that are needed to understand the work of individual algorithms. The ways for further research are offered as well.

Вступ

Людство живе в епоху інформації, тому з перших днів розвитку комп'ютерів почали використовуватися алгоритми зменшення розмірів файлів через обмежений розмір тогочасних жорстких дисків. На сьогодні, збільшення розмірів зображення виконується за рахунок більшої роздільної здатності, при цьому якість відео та аудіо покращується, що також впливає на займаний ними обсяг пам'яті та на швидкість передачі даних, що в свою чергу може зумовлювати великі інвестиції в дороговартісне обладнання для пропускної спроможності мереж. Інженери намагаються вигадати швидші методи передачі даних, а також можливості стиснення інформації, яку потім можна відновити. Стиснення даних також використовується для зберігання даних при резервному копіюванні, видаленні надлишковості.

Завдання стиснення складається з двох компонентів: алгоритм кодування, який приймає повідомлення та генерує стиснуте представлення інформації (з меншою кількістю бітів), а також алгоритм декодування, який реконструює оригінальне повідомлення або деяке його наближення зі стиснутого представлення [1].

Прикладом може бути стиснення зображень, що мають в собі послідовність повторюваних кольорів, як наприклад: синій-червоний-синій, що зберігається на всьому просторі інформації, може бути переписаний на один біт.

Оскільки для людини простіше розуміти інформацію з максимальною надлишковістю, а для засобів комп'ютерної техніки це відіграє негативну роль, виникає проблема позбавлення від зайвої інформації або стиснення даних.

Коли інформація передається через Інтернет, файли великого розміру (окремо або разом з іншими - як частина архівного файлу), можуть передаватися в ZIP, GZIP або іншому стисненому форматі.

Постановка задачі

Задача полягає в модифікації та аналізі алгоритмів стиснення даних з урахуванням особливостей мовного конструкту людини та способах представлення даних на інформаційних носіях. Як результат даної роботи дерево кодування Гаффмана буде займати менше місця, що зумовить менший час побудови даного дерева та обробки інформації в ньому для подальшого кодування та декодування інформації.

Термінологія

Кодування символів — це процес присвоєння символу послідовності довжиною в один або більше байт, що допомагає передавати та перетворювати їх за допомогою цифрових комп'ютерів.

Стиснення даних без втрат – метод стиснення даних при якому, закодована інформація може повністю декодуватися зі стиснутих даних.

Надлишковість інформації – величина, що показує наскільки можна стиснути зробивши повідомлення коротшим, якщо закодовано ту саму інформацію.

Алгоритм Гаффмана (Huffman coding) – це жадібний алгоритм кодування вхідних даних з мінімальною надмірністю.

H-дерево (Huffman tree) – повне двійкове дерево, в якому кожний leaf(листок) відповідає букві заданого алфавіту з вхідного тексту.

Інформаційна ентропія – випадковість або невпорядкованість системи, міра невизначеності.

Групове кодування RLE(Run-length encoding) – алгоритм стиснення, при якому серія однакових символів замінюється рядком, який містить повторювальний символ та кількість повторів.

Опис способу

Техніка використання алгоритму Гаффмана без втрат дозволяє відновлювати дані до їх автентичних станів, не втрачаючи при цьому інформації. У залежності від вхідних повідомлень: відео, аудіо чи текст, принцип роботи алгоритму не змінюється, лише змінюється принцип

обробки даної інформації. На рис. 1 ми можемо побачити процес побудови H-дерева.

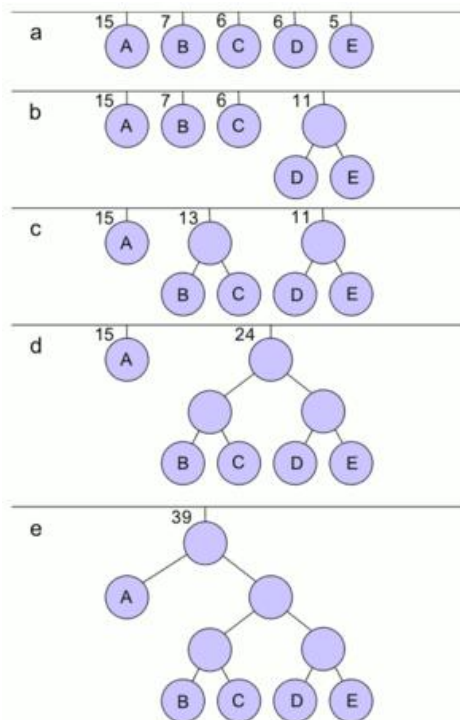


Рис. 1. Приклад побудови дерева з вхідної таблиці частоти

Метод Гафмана дещо схожий на метод Шеннона-Фано. Різниця між двома методами полягає в тому, що Шеннон-Фано будує свої коди вершиною донизу (від крайнього лівого до крайнього правого біта), тоді як Гафман будує кодове дерево знизу вгору (коди справа наліво) [2].

Даний метод починається з створення таблиці спадання ймовірностей зі списку вхідної послідовності символів повідомлення. Наступним кроком будується дерево з символом на кожному листку знизу вгору. Даний алгоритм проводиться покроково, де на кожному кроці вибираються два символи з найменшими ймовірності, що згодом додаються до верхньої частини часткового дерева і потім видаляються зі списку. Коли список скорочується тільки до одного символу, дерево завершено. Потім дерево проходить для визначення кодів символів.

У нашій оптимізації ми використали особливості мов та таблиць ASCII, що можна помістити лише в 7 бітів, а останній старший біт не використовується. При поверненні у вершину ми будемо заповнювати цей біт для того аби визначити положення дочірнього вузла. Це дозволить нам зберегти пам'ять, що у результаті пришвидшить наші результати.

Оцінка стиснення

При імплементації таких алгоритмів та методів в усуненні надлишковості даних використовується оцінка ефективності, або коефіцієнт стиснення, яке визначається як співвідношення обсягу вихідних нестиснутих даних до обсягу стислих.

$$k = \frac{S_0}{S_c},$$

де k - коефіцієнт стиснення,

S_0 - обсяг вихідних даних,

S_c - об'єм стислих.

З даної формули випливає, що чим більше k , чим більше повідомлення було стиснене. Залежно від вхідного тексту цей коефіцієнт буде різним. Якщо взяти англійську мову і присвоїти рівні ймовірності для всіх символів, то ентропія буде рівна 7, у випадку застосування вихідного тексту Гаффмана ентропія середня: 4.7 бітів на символ[3].

Висновки

Представлений метод враховує особливості мови та представлення її на інформаційних носіях. Ефективність цього способу забезпечується використанням незадіяного біту, що зменшує подальше використання пам'яті з пришвидшенням обходу дерева Гаффмана. Оскільки при декодуванні ми повертаємо не лише вихідний текст, а й саме дерево, що також займає місце, нам потрібно використовувати простір ефективніше.

Подальші дослідження можуть бути розширені на використанні інших вхідних даних, уніфікація для мов UNICODE, ефективніше об'єднання піддерев у одне дерево Гаффмана, а також спроби оптимізації їх побудов.

Література

1. *John R. Pierce*. An introduction to information theory. 78-107. 1961
2. *David Salomon*. Data Compression: The complete Reference. 68-84. 2006.
3. *Guy E. Blelloch*. Introduction to Data Compression. 5-9. 2013.

УДК 004.45

К.т.н., доцент Тарасенко-Клятченко О.В., студент Качанов В.І.

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

ПРОГРАМНИЙ МОДУЛЬ НАДСИЛАННЯ КОРОТКИХ ПОВІДОМЛЕНЬ СТІЛЬНИКОВОЮ МЕРЕЖЕЮ

Abstract

Tarasenko-Klyatchenko O.V, assoc. prof., PhD; Vitalii Kachanov, student
Modified iterative algorithm for plane Delaunay triangulation

This paper is dedicated to the creation of a software module for sending short messages to a cellular user. The choice of topics is due to the lack of powerful solutions that could be used during large marketing campaigns and reports of huge data objects.

Вступ

Вибір теми обумовлений відсутністю потужних рішень, які б могли використовуватись під час великих маркетингових компаній та відправки величезних об'ємів даних.

Виконано порівняльний аналіз відомих програмних застосунків для надсилання коротких повідомлень користувачам стільникового зв'язку. Завдяки цьому специфіковано функціональні вимоги до програмного модулю надсилання коротких повідомлень користувачам стільникового зв'язку та реалізовано програмний модуль надсилання коротких повідомлень користувачам стільникового зв'язку.

Постановка задачі

Задача полягає у порівнянні вже існуючих програмних застосунків, визначенні функціональних вимог та аналізі протоколу передачі коротких повідомлень

Термінологія

EMS – стандарт передачі повідомлень, який являє собою розширенням стандарту SMS для стільникових мереж стандартів GSM, TDMA.

GSM – міжнародний стандарт для мобільного цифрового стільникового зв'язку з розділенням каналу за принципом TDMA та високим рівнем безпеки за рахунок шифрування з відкритим ключем.

Опис програмного модулю та протоколів передачі коротких повідомлень

Служба коротких повідомлень, (англ. *SMS, Short Message Service* [1]) – сервіс передачі і прийому повідомлень в текстовому форматі по стільниковій мережі. Даною послугою можуть користуватись більшість пристроїв комунікації, таких як мобільний телефон, пейджер та інші. Для реалізації сервіс повинен підтримуватись оператором стільникового зв'язку, пристроєм для комунікації або програмним забезпеченням пристрою, який ви використовуєте. У випадку мобільного зв'язку для користування послугою потрібна *SIM*-карта. Дану технологію підтримують такі мобільні мережі як *GSM, NMT, D-AMPS, CDMA, UMTS* [2].

Максимальна стандартна довжина одного смс-повідомлення становить до 160 знаків латиницею або 70 кирилицею. Це зумовлено тим що максимальний розмір повідомлення за стандартом *GSM* – 140 байт або 1120 біт. Таким чином, при використанні 7-бітного кодування (латинський алфавіт та цифри) можна відправляти повідомлення довжиною 160 символів. Для підтримки інших національних алфавітів використовують 16-бітне кодування *UCS-2*, тобто максимальна довжина 70 символів [2]. Довші повідомлення розбиваються на кілька повідомлень. Оплата здійснюється за кожне повідомлення окремо згідно з умовами встановленого тарифу.

Особливості технології :

- *SMS*, як правило, доставляються протягом не більше 10 секунд. Відправник може отримувати повідомлення про доставку одержувачу.
- Можна відправити повідомлення на вимкнений або на телефон, що знаходиться поза зоною дії мережі. Як тільки адресат з'явиться в мережі, він отримає повідомлення. Якщо відправник отримує

Також було проведено порівняння різних програмних застосунків надсилання коротких повідомлень стільниковою мережею [3], встановлено переваги та недоліки. На основі отриманих результатів сформовано функціональні вимоги до програмного модулю надсилання коротких повідомлень користувачам стільникового зв'язку, а саме

Функціональні вимоги до серверної частини програмного модулю

- Можливість відправляти повідомлення
- Можливість створювати списки отримувачів повідомлень
- Можливість створювати короткі посилання
- Можливість отримувати звіти про отримання повідомлень користувачем
- Безперервна робота програмного забезпечення
- Можливість налаштовувати ім'я відправника
- Можливість відправляти відкладені повідомлення
- Можливість масштабування системи

Висновки

Було розглянуто програмні застосунки надсилання коротких повідомлень користувачам стільникового зв'язку. Проведено порівняння їх можливостей, встановлено переваги та недоліки. На основі отриманих результатів сформовано функціональні вимоги до програмного модулю надсилання коротких повідомлень користувачам стільникового зв'язку. Також було проаналізовано протокол передачі коротких повідомлень.

Література

1. SMS [Електронний ресурс] – Режим доступу до ресурсу: <https://bit.ly/2SDPPWW>
2. GSM [Електронний ресурс] – Режим доступу до ресурсу: <https://bit.ly/2SJBNmw>
3. The 5 best SMS apps in 2021 [Електронний ресурс] – Режим доступу до ресурсу: <https://bit.ly/34z3VM0>

УДК 004.8

Д.т.н., професор Терейковський І.А., студент Бойко Д.Ю.

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

МЕТОД РОЗРОБКИ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ АНАЛІЗУ ГОЛОСОВОГО СИГНАЛУ

Abstract

Ihor Tereikovskiy, prof., D.Sc.; Dmytro Boiko, student

Method of developing a neural network system for voice signal analysis

This work deals with the problem of methods of developing neural network systems for voice signal analysis. The basic method of developing a neural network system for voice signal analysis has been studied. A modified method of developing a neural network system for voice signal analysis based on independent component analysis is proposed. A comparative analysis of the effectiveness of both classical and modified methods was carried out.

Вступ

У сучасному світі одним з найбільш пріоритетних напрямків штучного інтелекту є соціальна робототехніка, в межах якої забезпечується взаємодія людини з автономними роботами-помічниками. Голосове спілкування є одним з найважливіших природних способів діалогу людини з роботом.

Щодня ринок пристроїв з технологіями розпізнавання мови все більше зростає: їх використовують не лише в якості голосових помічників у пристроях, але й для управління безпілотними автомобілями, і навіть в автоматизації виробництв.

Ця нейронна мережа дозволяє здійснювати навчання системи на аудіоданих та розпізнавання мови. Крім того, передбачено можливість використання системи через API у вигляді чат бота.

Постановка задачі

Задача полягає у визначенні недоліків відомих методів розробки нейромережових систем аналізу голосового сигналу та обґрунтуванні підходів їх виправлення у модифікованому методі.

Термінологія

Нейромережеві системи – це обчислювальні системи, що базуються на використанні штучних нейронних мереж.

Рекурентна нейронна мережа – це тип штучної нейронної мережі, який використовує послідовні дані або дані про послідовності. Для побудови рекурентних нейронних мереж використовуються алгоритми глибокого навчання, що зазвичай адаптовані до аналізу динамічних рядів даних.

Атака супротивника – це процес експлуатації, при якому в природні вхідні дані вносяться незначні зміни, що призводить до неправильної класифікації вхідних даних нейронними моделями.

Опис методу

Одною з основних задач в розпізнаванні слів з аудіозапису є отримання та попередня обробка файлу, адже від нього напряму залежатиме успіх та процент точності в розпізнаванні мовлення.

Атака супротивника створює безліч проблем для класифікаторів мови з обмеженим словниковим запасом. Для виявлення цих несприятливих прикладів використовуються комбінації попередньої обробки звукового файлу. Один з конкретних комбінованих методів захисту, що включає стиснення, кодування мови, фільтрацію і панорамування аудіо, виявився досить ефективним проти атаки на модель мовних команд, виявляючи несприятливі приклади аудіо з точністю 93,5% і ймовірністю 91,2%.

Алгоритми стиснення багатоканального звуку з низькою швидкістю передачі даних використовують методи трансформаційного кодування для видалення статистичної надмірності в кожному каналі багатоканального аудіофайлу. Стиснення звуку можливе за допомогою різних методів кодування, проте аудіофайл повинен бути стиснутий без зниження якості. Сегментація аудіо є основою для аналізу мультимедійного контенту.

Шум в аудіопотоках сегментується як звук навколишнього середовища. Також використовується гібридний підхід до класифікації, який об'єднує машини опорних векторів зі штучними нейронними мережами. Аудіопотік класифікується на тишу і сегменти чистої мови на основі класифікатора, заснованого на правилах.

Аудіопотік береться в якості вхідного сигналу, потім він знижується до 8000 Гц, до нього застосовується крок попередньої класифікації, витягуються ознаки нульової швидкості перетину, короткочасної енергії, спектрального потоку, коефіцієнтів Мел- частотної цесптральності і аналізу періодичності, і використовується гібридний класифікатор. На етапі попередньої обробки для сегментації аудіозаписів всі вхідні сигнали

знижуються до частоти дискретизації 8000 Гц. Потім звукові записи сегментуються на 1-секундні кадри. Цей 1-секундний кадр береться в якості основної класифікаційної одиниці. Для виділення ознак використовуються непересічні кадри. Ознаки вказують на характерну інформацію, присутню в кожному 1-секундному звуковому ефекті.

Висновки

В результаті проведеного аналізу відомих методів розробки нейромережових систем розпізнавання голосового сигналу на базі глибокого навчання визначено ряд суттєвих недоліків. Також в результаті проведених досліджень обґрунтовано висновок, що для взаємодії з нейронною мережею слід реалізувати REST-арі, що дає змогу працювати з різними типами даних та у різних режимах роботи.

За рахунок цього нейромережева система отримує можливість проводити навчання на специфічних навчальних базах даних, що значно підвищує універсальність її застосування. Також запропоновано ряд рішень, спрямованих на підвищення швидкості та точності розпізнавання. Вказані рішення пропонується покласти в основу модифікованого методу розпізнавання голосового сигналу.

Література

1. A hybrid algorithm for artificial neural network training [Електронний ресурс] – Режим доступу до ресурсу: <https://doi.org/10.1016/j.engappai.2012.01.023>
2. Speech Coding and Audio Preprocessing for Mitigating and Detecting Audio Adversarial Examples on Automatic Speech Recognition [Електронний ресурс] – Режим доступу до ресурсу: https://www.researchgate.net/profile/Basemah-Alshemali/publication/336083538_Speech_Coding_and_Audio_Preprocessing_for_Mitigating_and_Detecting_Audio_Adversarial_Examples_on_Automatic_Speech_Recognition/links/5d8d608c299bf10cff12bd27/Speech-Coding-and-Audio-Preprocessing-for-Mitigating-and-Detecting-Audio-Adversarial-Examples-on-Automatic-Speech-Recognition.pdf
3. Глибокі нейромережі та елементи їх реалізації FPGA [Електронний ресурс] – Режим доступу до ресурсу: <http://dSPACE.nbuu.gov.ua/xmlui/bitstream/handle/123456789/122853/16-Samolyuk.pdf?sequence=1>
4. Нейронні мережі [Електронний ресурс] – Режим доступу до ресурсу: <https://evergreens.com.ua/ua/development-services/neural-network.html>

УДК 004.9

Д.т.н., професор Терейковський І. А., аспірант Дідус А.В.

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

ПРОБЛЕМАТИКА РОЗПІЗНАВАННЯ ГОЛОСОВИХ СИГНАЛІВ В МАЛОРЕСУРСНИХ КОМП'ЮТЕРНИХ СИСТЕМАХ

Abstract

Ihor Tereikovskiy, professor, D.Sc.; Andrii Didus, PhD student

Justification of the relevance of research in the field of keyword recognition in the voice signal in low-resource computer systems

This paper concerns statements about the actuality of keyword spotting in voice signals in computer systems with low resources, main tasks, problems in this field.

Вступ

В сучасному світі тематика досліджень в області голосових сигналів вже багато років не втрачає актуальності, адже сьогодні, окрім наявності великої кількості текстової інформації, також можна знайти велику кількість даних, які можуть відображатись у голосових сигналах. Також все більшому поширенню набувають і технології аналізу голосових сигналів, поширення голосових помічників, як Google Assistant, Apple Siri на таких електронних пристроях, як годинники, колонки є гарним доказом цьому. Проте відмінною рисою голосових асистентів є використання різних методів аналізу голосових сигналів [1-3].

Постановка задачі

Задача полягає у обґрунтуванні актуальності тематики застосування методів динамічного програмування для розпізнавання ключових слів в голосовому сигналі в малоресурсних комп'ютерних системах.

Термінологія

Задача виявлення ключових слів – це задача ідентифікації ключових слів у голосових сигналах [1].

Мовний звук – найменша неподільна одиниця мовлення, яка утворюється апаратом мовлення.

Обґрунтування

Як показують результати досліджень [2], до одних із найбільш актуальних задач в області розпізнавання голосових сигналів відносяться:

- розпізнавання ключових слів;
- розпізнавання семантики мовлення;
- верифікація за допомогою голосу.

В даній роботі акцент ставиться на задачі розробки засобів розпізнавання ключових слів. Задача розпізнавання ключових слів полягає у ідентифікації ключових слів у аудіопотоці для їх подальшого використання в системах розпізнавання мовлення.

В багатьох випадках виникає необхідність адаптації таких засобів до умов використання у малоресурсних комп'ютерних системах, що дозволяє використовувати її, наприклад, для формування вхідного сигналу голосових помічників навіть на пристроях з обмеженими або відносно невеликими обчислювальними можливостями [1]. Наприклад, на Рис. 1 зображено схему використання модулю розпізнавання ключових слів як однієї зі складових малоресурсної комп'ютерної системи. Зазначимо, що в даному випадку розпізнане ключове слово використовується для активації голосового помічника.

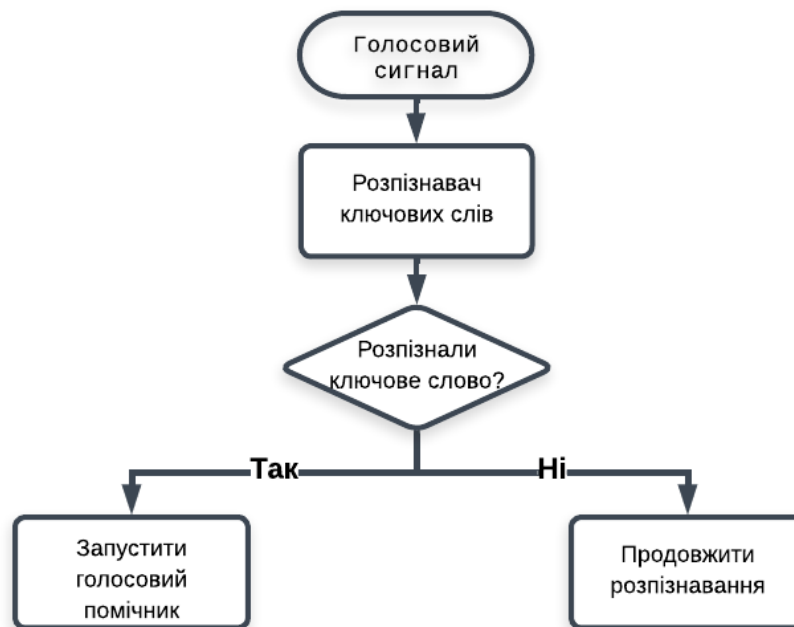


Рис. 1. Типова схема використання модулю розпізнавання ключових слів в голосовому сигналі

До найбільш популярних та актуальних підходів до вирішення задачі розпізнавання ключових слів у голосових сигналах належать підходи на основі:

- методів динамічного програмування;
- прихованої моделі Маркова;
- глибинних нейронних мереж.

В розпізнаванні ключових слів у голосових сигналах є певні особливості, які важливо враховувати, адже вони напряду впливають на ефективність аналізу вхідних даних. Однією із таких особливостей є попередня обробка і фільтрація вхідного голосового сигналу. Ще однією задачею, яка потребує специфічного підходу до вирішення, являється врахування різноманітних акцентів серед носіїв однієї мови, адже через це одні і ті ж самі слова можуть звучати зовсім по-різному, що може значно ускладнити процес розпізнавання. Також, використання модулю розпізнавання ключових слів в малоресурсних комп'ютерних системах обмежує доступні обчислювальні ресурси, які можуть бути використані для аналізу. Це значно обмежує потенціал сучасних методів розпізнавання, до яких в першу чергу відносяться нейромережеві методи та методи на основі прихованої моделі Маркова [5].

Висновки

Отже, варто чітко розуміти, що актуальність проблематики розпізнавання голосових сигналів в малоресурсних комп'ютерних системах лише зростає, а обмеженість обчислювальних ресурсів таких систем призводять до необхідності застосування методів динамічного програмування.

Література

1. López-Espejo, Iván & Tan, Zheng-Hua & Hansen, John & Jensen, Jesper., 2021. Deep Spoken Keyword Spotting: An Overview. Доступно: <https://www.researchgate.net/publication/357445314_Deep_Spoken_Keyword_Spotting_An_Overview>; [Дата звернення 06 Листопад 2022].
2. Momeni, L., Afouras, T., Stafylakis, T., Albanie, S., & Zisserman, A., 2020. Seeing wake words: Audio-visual Keyword Spotting, Доступно: <<https://arxiv.org/pdf/2009.01225.pdf>>; [Дата звернення 06 Листопад 2022].
3. Hoy M., 2018. Alexa, Siri, Cortana, and more: An introduction to voiceassistants. Medical Reference Services Quarterly vol. 37, с.81–88.
4. В. М. Михайленко, Л. О. Терейковська, І. А. Терейковський., Б.Б. Ахметов, 2017. Нейромережеві моделі та методи розпізнавання фонем в голосовому сигналі в системі дистанційного навчання, Монографія., К.: ЦП «Компринтр».

5. І.А. Терейковський, Д.А. Бушуєв, Л.О. Терейковська, 2022. Штучні нейронні мережі. Базові положення. Доступно: <<https://ela.kpi.ua/bitstream/123456789/50135/1/ANN.pdf>>; [Дата звернення 06 Листопад 2022].

УДК 004.9

Д.т.н, професор І.А. Терейковський, аспірант О.С. Коровій

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

ОБҐРУНТУВАННЯ АКТУАЛЬНОСТІ ДОСЛІДЖЕНЬ В ОБЛАСТІ РОЗПІЗНАВАННЯ ЕМОЦІЙНОЇ ТОНАЛЬНОСТІ ФРАГМЕНТІВ ТЕКСТУ В УНІВЕРСАЛЬНИХ КОМП'ЮТЕРНИХ СИСТЕМАХ

Abstract

Ihor A. Tereikovskiy prof., D.Sc.; Oleksandr Korovii, student

Justification of the relevance of research in the field of recognition of emotional tone of text fragments in universal computer systems

This paper concerns the task of recognizing the emotional tonality of text fragments has become an integral and relevant part of scientific research. The accuracy of sentiment analysis can not only help organizations determine how they are perceived, but can also help them identify potential pitfalls in their marketing operations and branding content that can be addressed in time.

Вступ

Розпізнавання емоційної тональності фрагментів тексту стало невід'ємною та актуальною частиною наукових досліджень. Точність аналізу настроїв може не тільки допомогти організаціям визначити, як їх сприймають, але також може допомогти їм виявити потенційні підводні камені в їхніх маркетингових операціях і вмісті брендингу, які можна вчасно вирішити. Хоча багато компаній стикаються з проблемами аналізу емоційної тональності, вирішення яких потребує проведення досліджень і використання нейромережових моделей та методів [1].

Постановка задачі

Задача дослідження полягає у обґрунтуванні актуальності проблематики досліджень в області розпізнавання емоційної тональності фрагментів тексту в універсальних комп'ютерних системах.

Об'єкт дослідження – процеси розпізнавання емоційної тональності фрагментів тексту в універсальних комп'ютерних системах.

Предмет дослідження – нейромережові моделі та методи розпізнавання емоційної тональності фрагментів тексту в універсальних комп'ютерних системах.

Термінологія

Машинне навчання (англ. machine learning) – це підгалузь штучного інтелекту в галузі інформатики, яка часто застосовує статистичні прийоми для надання комп'ютерам здатності «навчатися» (тобто, поступово покращувати продуктивність у певній задачі) з даних, без того, щоби бути програмованими явно [1].

Штучні нейронні мережі – це обчислювальні системи, натхнені біологічними нейронними мережами, що складають мозок тварин. Такі системи навчаються задач (поступально покращують свою продуктивність на них), розглядаючи приклади, загалом без спеціального програмування під задачу [1].

Опис проблематики

Виділяються наступні проблеми, які потрібно вирішувати при використанні нейромережових моделей та методів для розпізнавання емоційної тональності фрагментів тексту:

1. Визначення емоційної тональності

Тон може бути важко витлумачити усно[2], і ще важче зрозуміти в письмовому слові. Справа стає ще складнішою, коли хтось намагається проаналізувати величезний обсяг даних, який може містити як суб'єктивні, так і об'єктивні відповіді.

2. Полярність слів

Такі слова, як «любов» і «ненависть», мають високі позитивні і негативні значення за полярністю. Це легко зрозуміти. Але є проміжні сполучення слів на зразок «не так погано», які можуть означати «середнє» значення і, отже, знаходяться в середній полярності. Іноді такі фрази пропускають, що погіршує оцінку настроїв.

3. Сарказм

Люди використовують іронію та сарказм у невимушених розмовах та мемах у соціальних мережах. Акт вираження негативних почуттів за допомогою компліментів із зворотного боку може ускладнити інструментам аналізу настроїв виявлення справжнього контексту того, що насправді означає відповідь. Це часто може призвести до збільшення кількості «позитивних» відгуків, які насправді є негативними.

4. Емодзі

Проблема текстового вмісту соціальних мереж, як-от Twitter або Facebook, полягає в тому, що вони переповнені емодзі. Моделі для обробки природньої мови навчаються з урахуванням мови. Більшість рішень для аналізу емоцій сприймають емодзі як спеціальні символи, які видаляються з даних у процесі аналізу настроїв. Але це означає, що в результаті не отримуємо цілісної інформації з даних.

5. Ідіоми

Моделі машинного навчання не обов'язково розуміють фігуру мови. Наприклад, така ідіома, як «не моя чашка чаю», вразить модель, оскільки вона розуміє речі в буквальному сенсі. Отже, коли в коментарі чи огляді використовується ідіома, речення може бути неправильно витлумачено алгоритмом або навіть проігноровано. Щоб подолати цю проблему, платформу аналізу настроїв потрібно навчити розуміти ідіоми [3].

6. Заперечення

Заперечення, задані такими словами: “як не”, “ніколи”, “не можу”, “не було тощо”, можуть заплутати модель. Наприклад, модель повинна розуміти, що фраза, яка говорить: «Я не можу, не піти на зустріч свого класу», означає, що людина має намір піти на зустріч класу.

7. Порівняльні речення

Порівняльні речення можуть бути складними, оскільки вони не завжди можуть висловлювати думку. Такі речення підлягають додатковому аналізу. Наприклад, коли хтось пише: «телефон Galaxy S20 більший, ніж iPhone 12», у реченні не згадуються жодні негативні чи позитивні емоції, а вказується відносний порядок з точки зору розміру двох телефонів.

8. Аналіз багатомовної емоційної тональності

Аналіз багатомовної емоційної тональності означає, що всі перераховані вище проблеми ускладнюються, коли додається суміш мов. Оскільки кожна мова унікальна, її не можна перекласти на базову мову. Простий приклад: якщо ідіому «як риба у воді» перекласти, скажімо, німецькою мовою, ця ідіома втратить своє значення [4].

Висновки

Дослідження в області розпізнавання емоційної тональності фрагментів тексту в універсальних комп'ютерних системах є актуальними та важливими з практичної точки зору. Проведений аналіз показує, що найбільш перспективні засоби такого розпізнавання доцільно будувати на основі нейромережових моделей та методів. Використання рішень в даній області, дозволяє автоматизувати аналіз фрагментів текстів для визначення емоційної тональності, що дозволить полегшити та пришвидшити роботу при великій кількості текстової інформації.

Література

1. *Oleksandr Korovii, Andrii Petrashenko. Adaptation of Distilling Knowledge Method in Natural Language Processing for Sentiment Analysis. Advances in Computer Science for Engineering and Manufacturing, 2021. – С. 170 - 180.*

2. *Sergey Toliupa, Ihor Tereikovskiy, Liudmyla Tereikovska, Shynar Mussiraliyeva, Kalamkas Bagitova*. Deep Neural Network Model for Recognition of Speaker's Emotion. 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T), 2020. – C. 172-176.
3. *Vysotska, V., Fernandes, V.B., Lytvyn, V., Emmerich, M., Hrendus, M.* Method for Determining Linguometric Coefficient Dynamics of Ukrainian Text Content Authorship. Advances in Intelligent Systems and Computing, 2020. – C. 132–151.
4. *Hu, Z., Tereykovskiy, I., Zorin, Y., Tereykovska, L., & Zhibek, A., 2018*. Optimization of convolutional neural network structure for biometric authentication by face geometry. In International Conference on Computer Science, Engineering and Education Applications (pp. 567-577). Springer, Cham.

УДК 004.65

Д.т.н., професор Терейковський І. А., аспірант Самофалов А. В.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

ПРОБЛЕМАТИКА ФОРМУВАННЯ ВЕБ-ОРІЄНТОВАНИХ БАЗ ДАНИХ ЕМОЦІЙНО ЗАБАРВЛЕНИХ ГОЛОСОВИХ СИГНАЛІВ

Abstract

Ihor Tereikovskiy, professor, D.Sc.; Andrii Samofalov, PhD student

Problems of forming web-oriented databases of emotional voice signals

This paper concerns the problems of forming web-oriented databases of emotional voice signals. The overview of existing databases is provided. The importance of creating new approaches to forming speaker-independent and text-independent emotion recognition systems is shown.

Вступ

Розпізнавання емоцій людей, що нас оточують, є невід'ємною складовою життя кожної людини. Ми вчимося цим навичкам з самого дитинства, зважаючи увагу на найменші зміни вигляду людини, її голосу та ситуації, під час яких ці зміни відбуваються, й події, які їх породжують. Завдяки цим вмінням ми можемо краще реагувати на емоційні зміни наших колег, друзів, тощо, та, відповідно, вживати необхідних заходів у всіх соціальних ситуаціях.

Постановка задачі

Задача полягає в наведенні існуючих проблем та викликів, з якими стикаються розробники баз даних емоційно забарвлених голосових сигналів.

Об'єктом дослідження – процеси розпізнавання людських емоцій в голосових сигналах та формування з них баз даних.

Предметом дослідження є нейромережеві моделі та методи розпізнавання емоцій людини в аудіофрагментах.

Термінологія

Емоція – психологічний стан людини, що виникає в результаті нейрофізіологічних змін, пов'язаних з думками, почуттями, поведінковими реакціями і ступенем задоволення чи страждання.

Голосовий сигнал – складний акустичний сигнал, джерелом якого являється голос людини.

Опис проблематики

У той час, як розпізнавання емоцій не є для нас чимось незвичним, для комп'ютерів це є дуже складною задачею, яка потребує певних вхідних даних та відповідних алгоритмів, які можуть ці дані потім обробити та надати результат, що є якомога правильнішим відносно тих вхідних даних. В якості вхідних даних можуть використовуватися зображення обличчя людини, зразки її голосу, жести, позиція тіла та багато інших. Багато існуючих алгоритмів комп'ютерного розпізнавання емоцій людини працюють лише з певними вхідними даними, наприклад, зображенням обличчя людини [1]. Тоді результатом роботи цих алгоритмів буде зображення обличчя в певний час та тип емоції, що з найбільшою вірогідністю йому відповідає.

Як показує аналіз літературних робіт [2], прийнято виділяти 6 базових емоцій до яких зазвичай відносять щастя, гнів, відраза, страх, сум і здивування. На додачу до перелічених емоцій, як правило, додають емоцію, що співвідноситься з нейтральним станом.

Очевидно, кількість реальних емоцій, що може відчувати людина, є набагато більшою, але з позицій розробки ефективних комп'ютерних засобів розпізнавання доцільно представити емоційний стан людини у вигляді інтегрованої суміші вказаних вище базових емоцій.

Звісно, з продовженням покращення алгоритму, його можливості можна настроїти на розпізнавання додаткових емоцій, поступово розширюючи список початкових емоцій, що були закладені при розробці: наприклад, додатковими емоціями можуть стати впевненість, симпатія, розчарування та значна кількість інших емоцій, що може відчувати людина.

Основним призначенням веб-орієнтованих баз даних емоційно забарвлених голосових сигналів людини є їх участь у навчанні нейронних мереж, завдяки чому потім можна досягти певних результатів у розпізнаванні емоцій у голосі людини: або проводити розпізнавання в реальному часі та відповідним чином реагувати на це, або на вже відокремлених голосових сигналах, як у навчальних прикладах, але при цьому самостійно встановлювати найбільш ймовірну емоцію, що присутня в аудіофрагменті. В якості застосування розпізнавання емоцій в

реальному часі можна навести їх використання в голосових асистентах та елементах розумного дому: наприклад, можна встановити, що якщо людина знаходиться в депресії, то тоді їй потрібно надавати в рекомендаціях більш веселий, або нейтральний, контент, або якісь поради як з цього стану вийти. Також можна відрізнити ситуації, в яких щось трапилось зі здоров'ям людини, та викликати швидку допомогу.

Як показують результати досліджень в області розпізнавання емоцій людини, одним із найбільш перспективних рішень є нейромережеві засоби аналізу голосу людини [3]. Однак, основною перешкодою створення таких систем є відсутність баз даних з достатньою кількістю навчальних прикладів, що використовуються для побудови відповідних нейромережевих моделей. У мінімальному значенні, ці бази мають складатися з аудіо фрагментів запису голосу людини та базової емоції, яка присутня в цьому аудіосигналі. Також до списку значень, що містить даний аудіофрагмент, можна додати не тільки вказання емоції, що міститься у ньому, а також її тривалість, тон голосу, тощо.

Основними перешкодами, з якими стикаються розробники баз даних емоційно забарвлених аудіофрагментів, є те, що для запису голосових сигналів необхідні професійно навчені люди, які можуть відтворити у своєму мовленні весь спектр необхідних емоцій. Додатково до цього, необхідне приміщення без зайвих шуму та аудіоперешкод й спеціалізоване обладнання, що зможе записувати голосові фрагменти у гарній якості, відповідно до мінімальних вимог для цих записів.

Розглянемо одну з існуючих баз даних емоційно забарвлених голосових сигналів людини. Є велика кількість наявних баз, але більшість з них об'єднує те, що кількість наявних аудіофрагментів в цих базах даних складає від декількох сотень до лише декількох тисяч.

В якості прикладу розглянемо ESD (Emotional Speech Database) [4], яка складається з 350 голосових сигналів, які були створені 10 носіями англійської мови та 10 носіями китайської мови. В якості базових емоцій були виділені нейтральний тон, радість, злість, сум та здивування. Ці 350 голосових фрагментів сумарно складають більше 29 годин аудіозапису.

Як можна побачити, при дуже невеликій кількості фрагментів голосових сигналів, їх загальний час є дуже значною величиною. Через це формування баз даних емоційно забарвлених голосових сигналів є витратною операцією не стільки в планах вимог до обладнання і акторів, скільки до часу, враховуючи скільки ще додаткових годин було витрачено на обробку отриманих сигналів та перезапис не дуже вдалих фрагментів.

Не зважаючи на це, основним недоліком існуючих баз даних є їх залежність від підготовлених вхідних даних та акторів, що не дозволяє

створювати текстонезалежні та дикторонезалежні системи розпізнавання емоцій. Літературний аналіз показав відсутність єдиних загальноприйнятих підходів автоматизації створення таких баз даних.

Висновки

В сучасних умовах акцент слід зосередити на створенні веб-орієнтованих баз даних емоційно забарвлених голосових сигналів. Причиною для цього слугує використання Інтернету майже в усіх сферах людської діяльності, тому існує необхідність в базах даних, до яких можна отримати доступ саме через цю мережу.

Цим обумовлюється актуальність досліджень в області створення моделей та методів формування веб-орієнтованих баз даних емоційно забарвлених голосових сигналів. Розробка нових автоматизованих підходів до їх формування є нагальною проблемою, яка допоможе зменшити час на формування учбових екземплярів та надасть їх необхідну кількість.

Література

1. Hu, Z., Tereykovskiy, I., Zorin, Y., Tereykovska, L., & Zhibek, A., 2018. Optimization of convolutional neural network structure for biometric authentication by face geometry. In International Conference on Computer Science, Engineering and Education Applications (pp. 567-577). Springer, Cham.
2. Shiota, Michelle N., 2016. Ekman's theory of basic emotions. In Miller, Harold L. (ed.). The Sage encyclopedia of theory in psychology. Thousand Oaks, CA: Sage Publications. pp. 248–250.
3. Akhmetov, B., Tereykovsky, I., Doszhanova, A., & Tereykovskaya, L., 2018. Determination of input parameters of the neural network model, intended for phoneme recognition of a voice signal in the systems of distance learning. International Journal of Electronics and Telecommunications, 64.
4. Zhou, K., Sisman, B., Liu, R. and Li, H., 2022. Emotional voice conversion: Theory, databases and ESD. Speech Communication, 137, pp.1-18.

УДК 004.4'2

Д.т.н., професор Терейковський І.А., студент Юрченко В.В.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

ПРО СИСТЕМУ АВТОМАТИЗОВАНОГО АНАЛІЗУ БЕЗПЕКИ API

Abstract

Ihor Tereikovskiy, professor, D.Sc.; Valentyn Yurchenko, student
About automated API security analysis system

One of the most common methods in the implementation of unauthorized actions in relation to software products by intruders is the exploitation of software vulnerabilities. For this reason, the use of security scanners at the stage of software development becomes an integral part of security and is integrated into the software development cycle. Thus, the task of identifying vulnerabilities at the development stage is an urgent task, the implementation of which can significantly reduce the number of vulnerabilities in software, and hence the potential risks for both the end user and the business at all.

Вступ

На даний момент автоматизація різноманітних процесів є активно розвивається як при реалізації промислових задач так і в звичайному побуті будь-якої людини. В умовах цифровізації безпека програмного забезпечення стає пріоритетним направленням у всіх аспектах життєдіяльності людини, пов'язаних з використанням програмних продуктів.

Одним з найбільш розповсюджених методів при реалізації несанкціонованих дій по відношенню до програмних продуктів зі сторони злоумисників є експлуатація вразливостей програмного забезпечення. Реалізація даних дій може призвести до того, що потенційний злоумисник може отримати доступ, як до самих даних, так і до реалізації протиправних дій з використанням функцій і алгоритмів програмного забезпечення.

Саме за цієї причини використання сканерів безпеки іще на етапі розробки програмного забезпечення стає невід'ємною частиною забезпечення безпеки і інтегровано в цикл розробки програмного забезпечення.

Таким чином задача по виявленню вразливостей на етапі розробки є актуальною задачею, реалізація якої може суттєво зменшити кількість

вразливих місць в програмному забезпеченні, а отже і потенційні ризики, як для кінцевого користувача, так і для бізнесу в цілому.

Постановка задачі

Задача полягає в обґрунтуванні вимог до розробки автоматизованих засобів для аналізу програмного коду на предмет виявлення вразливостей в області захисту інформації.

Термінологія

API (Application Programming Interface) – інтерфейс, який дозволяє різним програмним компонентам взаємодіяти один з одним [1].

JSON – представляє собою полегшений текстовий формат обміну даними, незалежний від мови і розроблений на основі стандартів мов програмування сценаріїв ECMAScript. JSON визначає невеликий набір правил для представлення структурованих даних [2].

YAML – формат серіалізації даних, розроблений для комфортного читання людиною [3].

Swagger – незалежна від мови специфікація для описання API. Вона дозволяє користувачам краще зрозуміти можливості API без прямого доступу до коду [4].

Результати досліджень

За результатами аналізу був сформований список вимог до реалізації системи:

1. Можливість самостійної авторизації в API.
2. Можливість швидкого і легкого додавання нових механізмів виконання перевірки.
3. Можливість гнучкого налаштування режиму сканування.
4. Можливість контролю фільтрації вхідних параметрів та їх конфігурування.
5. Можливість легкої інтеграції з іншими системами контролю безпеки програмного забезпечення.

В рамках підготовки до рішення поставленої і чітко сформованої задачі був проведений аналіз потреб до системи автоматизованого аналізу безпеки програмного інтерфейсу додатків. За результатами виконаного аналізу був сформований список основних вимог, безпосередньо, до функціоналу системи:

1. Запуск динамічного сканування і аналізу коду на вразливість в режимі реального часу.

2. Статичний аналіз коду і перевірка залежностей.

Статичний аналіз коду може виконуватись наступними, вже існуючими на ринку, сканерами:

1. Checkmarx – статичний аналізатор коду з можливістю проведення повторного сканування коду по розкладу.

2. OWASP Dependency check – сканер, що аналізує код на присутність вразливих залежностей.

3. Npm audit – перевірка залежностей Angular проектів.

Динамічний аналіз коду може виконуватись наступними, вже існуючими на ринку, сканерами:

1. Arachni – сканер безпеки інтернет-додатків.

2. DirSearch – пошук скритих директорій.

Висновки

Показано актуальність задачі підвищення рівня захисту програмного забезпечення. На основі аналізу науково-практичних робіт в області автоматизованого пошуку вразливостей програмного коду та на основі аналізу відомих сканерів безпеки програмного коду визначено перелік вимог до розробки відповідної системи захисту інформації.

Література

1. *Bray T.* 2017. The JavaScript Object Notation (JSON) Data Interchange Format. pp. 28-32.
2. *H. Morris.* 2005. Protection in programming languages. pp.15–21.
3. *Pautasso E.* 2011. REST: From Research to Practice. pp.14-18.
4. *G. Plotkin.* 2004. A structural approach to operational semantics. J. Log. Algebr. Program. pp. 60-61.
5. *E. Brendan.* 2009. ECMA-262: ECMAScript Language Specification. pp. 89-94.

УДК 004.4'6

К.т.н., с.н.с. Тесленко О.К., студент Сенеджук А.Ю.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

ПРО АВТОМАТИЗАЦІЮ ТЕСТУВАННЯ У ПОРІВНЯННІ ІЗ ФУНКЦІОНАЛЬНИМ ТЕСТУВАННЯМ

Abstract

Teslenko O.K., PhD; Senedzhuk Andrii, student

About automatization of testing comparing functional testing

Failures discovered during the testing process are most common are caused by defects and errors present in the test software system. They can also be a consequence of behavior operational or test environment. Such defects can be more frequent all must be analyzed to determine the moment and place of the first appearance of this defect in the system, what types of errors caused it of these defects and when they were first discovered. All this information is used to determine how to improve the testing process itself and how critical the need for such improvements is.

Вступ

Збої, виявлені у процесі тестування, найчастіше породжуються дефектами та помилками, присутніми в тестовій програмній системі. Також вони можуть бути наслідком поведінки операційного чи тестового оточення. Такі дефекти можуть і частіше всього повинні аналізуватися для визначення моменту та місця першої появи даного дефекту в системі, які типи помилок стали причиною цих дефектів і коли їх виявили вперше. Вся ця інформація використовується для визначення того, як же покращувати сам процес тестування і наскільки критична необхідність таких вдосконалень.

Постановка задачі

Задача полягає в тому, аби проаналізувати переваги та недоліки автоматизованого тестування у порівнянні із ручним(функціональним).

Термінологія

Тестування програмного забезпечення – процес аналізу програмного засобу та супутньої документації з метою виявлення дефектів та підвищення якості продукту.

Функціональне тестування - виявлення не відповідності між реальною поведінкою реалізованих функцій і очікуваною поведінкою відповідно до специфікації і вимог[1].

Автоматизоване тестування - частина процесу тестування на етапі контролю якості в процесі програмного забезпечення. Воно використовує програмні засоби для виконання тестів і перевірки результатів виконання, що допомагає скоротити час тестування і спростити його процес[2].

Модель розробки ПЗ (Soft ware Development Model, SDM) - структура, яка систематизує різні види проектної діяльності, їх взаємодія і послідовність у процесі розробки ПЗ. Вибір тієї чи іншої моделі залежить від масштабу та складності проекту, предметної галузі, доступних ресурсів та безлічі інших факторів.

Результати дослідження та оптимізації

Найважливіше правило впровадження покращень: спочатку сформулюється мета і виявляються проблеми у її досягненні. і, звісно, без згуртованості всіх членів команди та без спільної зацікавленості в успіху проекту збір метрик абсолютно марний - зібрані результати не вдасться оцінити, щоб використовувати їх надалі.

Проаналізовано переваги та недоліки автоматизованого тестування порівняно з ручним (функціональним):

1. Швидкість виконання тест-кейсів може в рази і на порядки перевершувати можливості людини;
2. Відсутній вплив людського фактору у процесі виконання тест-кейсів (втоми, неуважності тощо);
3. Засоби автоматизації здатні виконати тест-кейси, принципово непосильні для людини через свою складність;
4. Засоби автоматизації здатні збирати, зберігати, аналізувати, агрегувати та представляти у зручній для сприйняття людиною формі колосальні обсяги даних.
5. Засоби автоматизації здатні виконувати низько рівневі дії з додатком, операційною системою, каналами передачі і т. д.

Однак із впровадженням автоматизованого тестування потрібно врахувати ряд ризиків:

1. Необхідність наявності висококваліфікованих спеціалістів у силу того факту, що автоматизація – це проєкт із своїми вимогами, планами, кодом і т. д.;
2. Розробка і супроводження як самих автоматизованих тест-кейсів, так і всієї необхідної інфраструктури;
3. Автоматизація потребує більш ретельного планування та керування ризиками;
4. Комерційні засоби автоматизації тестування потребують значно більших бізнес-витрат;
5. Засобів автоматизації тестування величезна кількість, що породжує проблему вибору того чи іншого засобу;

Висновки

Оцінка результатів тестування формується на підставі очікуваних результатів роботи системи та збору статистики за заведеним помилок. На продуктове середовище не повинні потрапити блокуючі та критичні помилки в системі, інакше компанії ризикують втратити клієнтів та прибуток. Доцільність застосування автоматизованого тестування оцінюється заздалегідь, тому що в деяких випадках ефект від функціонального (ручного) тестування може бути вищим.

Література

1. Svyatoslav Kulikov. Software Testing – Base Course – 3rd edition – typographic version. // Liderbooks.com.ua
2. Mark Fluster - Software Test Automation: Effective Use of Test Execution Tools // Mark Fluster /dp/0201331403?_

УДК 004.056

д.т.н., професор Яценко В.О., студент Журба А.В.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

ЗАХИСТ ДАНИХ У ХМАРНИХ ТЕХНОЛОГІЯХ

Abstract

*Yatsenko V.O, prof., PhD; Anton Zhurba, student
Data protection in cloud technology*

Cloud technologies open wide opportunities for the average user to work with large amounts of data. But this data in most cases is unprotected. This paper is devoted to the problem of data protection in cloud technologies. The main provisions about data in cloud technologies and the risks that may arise are described. Proposed methods for data protection.

Вступ

Хмарні технології відкривають перед пересічним користувачем широкі можливості для роботи з великими обсягами даних, а також важливими ресурсами, що можуть знаходитись в хмарному сховищі. Характерною особливістю хмарних технологій є типи доступу до них, хмари можуть бути приватні, публічні або змішаного типу. Приватними хмарами користувач користується одноосібно, вони розгортаються локально, натомість публічними можуть одночасно користуватися багато людей. Гібридні хмари займають проміжне становище між приватними й публічними. Найбільш привабливими для користувачів хмарних ресурсів є публічні хмари, бо вони надають широкі можливості, але й вони є найнебезпечнішими, адже зберігають велику кількість інформації від різних користувачів, яка передається, зберігається та копіюється, тож треба приділити особливу увагу захисту інформації що зберігається на публічних хмарах.

Постановка задачі

Задача полягає в аналізі різноманітних методів захисту даних у хмарних технологіях.

Результати дослідження та оптимізації

Розв'язання проблеми захисту даних потребує значних витрат.

Основні проблеми з якими стикаються користувачі – це викрадення, спотворення чи пошкодження даних.

В хмарних технологіях неможливо досягти ідеальної системи. Основні критерії щодо даних можливо сформувані за допомогою теореми CAP[1]:

- Цілісність – дані не можуть бути підроблені чи знищені під час передачі, обробки або зберігання
- Доступність – дані доступні з будь якого місця, і є еквівалентними
- Узгодженість – жодна зі сторін не може заперечити зміни щодо відбулися над даними

Також варто виділити аспект конфіденційності, дані не можуть бути передані третій стороні без згоди інших сторін.

Потенційні ризики:

- Втручання в мережу провайдера
- Атака сторонніми каналами[2]
- Розголошення даних провайдером

Для зниження ризиків можливо привести такі методи:

- Шифрування – по суті приховування даних. Отримати початкові дані може лише людина в якій є ключ. Можна виділити два сімейства шифрування. Асиметричне – ключ шифрування відрізняється від ключа дешифрування, перший публічно транслюється, а другий повинен залишатися в таємниці. Симетричне – відправник і одержувач мають однаковий секретний ключ, який використовується для шифрування і дешифрування, недоліком є збільшення пар ключів.
- Система безпеки в хмарних обчисленнях – NIDS[3]. Багатопотокова система складається з трьох модулів, а саме: модуль збору даних, модуль аналізу та модуль звітності. Перший з них відповідає за перехоплення пакетів даних та відправляє їх до секції аналітики. Другий модуль ефективно аналізує їх на відповідність заздалегідь визначеному набору правил і виділяє пошкоджені та підозрілі пакети з трафіком для подальшої генерації звіту. Третій модуль – модуль звітності інтерпретує тривоги, складає звіт про аналіз, чи негайно повідомляє про підозрілий трафік.
- Токенізація даних – заміна конфіденційних даних іншими значення (токенами). Токен є посиланням на зашифрований текст – тому такі повідомлення неможливо зламати в порівнянні з шифруванням. Токени потребують значно менше обчислювальних ресурсів для обробки. При токенізації конкретні дані залишаються повністю або частково видимими для обробки та аналітики, в той час як

конфіденційна інформація залишається прихованою. Це дозволяє обробляти токенизовані дані швидше і зменшує навантаження на системні ресурси. Це може бути ключовою перевагою в системах, які покладаються на високу продуктивність. У порівнянні з шифруванням, технології токенизації скорочують час, витрати і адміністративні зусилля, забезпечуючи при цьому командну роботу і комунікацію. Але цей метод не підходить для великих наборів даних (на великих наборах даних використовується процес шифрування).

- Анонімізація даних вимагає, щоб інформація, яка може безпосередньо ідентифікувати особу, була видалена або узагальнена. Можливо виділити два ефективні методи:

K-anonymity[4] – це метод, суть якого полягає в тому, що кожен запис стає доступним у базі даних, і він не відрізняється від принаймні k записів. Вважається, що k осіб належать до одного класу еквівалентності. Існує два кроки для досягнення k -анонімності: визначити набори атрибутів, які можуть бути використані для об'єднання анонімних даних з даними з іншого джерела чи узагальнити ці атрибути таким чином, щоб існувало щонайменше k різних записів, які мають однакове значення.

Differential privacy[5] – це метод який вносить шум у дані. Представляє реальні дані з ймовірністю a , і фіктивні з ймовірністю b . Потім об'єднує їх в одну базу даних. Буде важко спостерігати за цим набором даних, щоб відрізнити реальну інформацію від фальшивої. Також цей метод використовує підмножини, а не всю базу даних.

Ці алгоритми були обрані з огляду на наступні факти:

- Алгоритми широко згадуються в літературі
- Алгоритми використовують різні стратегії анонімізації, що дозволяє провести більш повну оцінку
- Публічна реалізація цих алгоритмів є доступною
- Імплементация алгоритмів дозволяє їх використовувати дуже ефективно при використанні хмарних технологій

Висновки

У роботі висвітлено важливість захисту публічних та приватних даних, що зберігаються за допомогою хмарних технологій. Розглянуто основні аспекти захисту хмарних даних, за допомогою теореми CAP було сформульовано основні критерії для даних. Оцінено ризики, які можуть виникнути при розробці системи захисту хмарних даних, запропоновані

методи для зниження цих ризиків. Підготовлена модель для оцінки параметрів захищеності даних.

Література

1. Seth Gilbert and Nancy Lynch, "Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services", ACM SIGACT News, Volume 33 Issue 2 (2002), p. 51–59
2. Толмачов, Є. Ю. Криптографічні атаки на RSA на основі інформації з побічного каналу – Київ, 2022. – 62 с.
3. Eric Conrad, Joshua Feldman, in Eleventh Hour CISSP (Third Edition), 2017
4. Nguyen, An "Understanding Differential Privacy". Towards Data Science Sweeney, Latanya "k-anonymity: a model for protecting privacy" (PDF). International Journal on Uncertainty, Fuzziness and Knowledge-based Systems. p. 557–570

Maria Y. Antusheva, student, Shkurat S. Oksana, PhD

**National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”**

THE MODIFIED NEURAL NETWORK LEARNING ALGORITHM IN THE MEDICAL IMAGE CLASSIFICATION SYSTEM

Abstract

**магістрант Антушева М. Я., к.т.н., Шкурат О. С.
*Модифікований алгоритм навчання нейронної мережі в системі
класифікації медичних зображень***

Стаття присвячена модифікованому алгоритму навчання нейронної мережі для діагностичної системи виявлення раку шкіри. Запропонований алгоритм модифікації оснований на сумісному використанні алгоритмів навчання Adam та SGD. Запропонований алгоритм було реалізовано для моделей ResNet-32, DenseNet, PyramidNet, SENet. Застосування запропонованого алгоритму призводить до зменшення кількості помилок від 4.0-7.5% при збереженні базової швидкості сходимості.

Introduction

Medical and biological data visualization is an informative method of conducting diagnostic studies. Therefore, hardware and software tools that visualize and research the work of the patient's organs in real time are used by medical professionals [1]. Further analysis of obtained graphic data is used to increase the efficiency of medical care [2]. The analysis of these volumes of medical images is referred to in the field of big data, therefore the development and implementation of artificial intelligence technologies in modern medicine is anthe urgent task.

There are many different types of medical research closely related to imaging tasks, such as angiography or medical ultrasound, which allow to visualize the inside of blood vessels or organs of the body accordingly. There are software solutions are needed the refine [3] or the cost of diagnostic software is not available for a lot of medical institutions in Ukraine.

The skin is one of the largest human organs and the main protective barrier between the external environment and internal organs. The skin provides protection against infections, mechanical or physical damage and participates in the regulation of body temperature. Pathological lesions of the skin of the population are increasing every year according to the statistical data of Ministry

of Health of Ukraine. Skin cancer was not even among the top ten cancer diagnoses in 2010, but it was already in the fifth place in 2020 [4]. The study of the size, color, shape, structure, and deformation of pathological neoplasms based on the obtained magnified image of the researched area with a lesion is the most important stage of diagnosis, which provides a conclusion about the condition of the skin [5]. The development and implement of the technologies of medical image processing and recognition for the automated determination of parameters and data classification will increase the efficiency of these processes.

Research analysis

The effectiveness of image recognition methods achieves the 1% errors and the 5% incorrect interpretations due to the use of open knowledge bases reaches [6, 7, 8]. Thus, the development and improvement of image recognition methods for medical diagnostic systems are remained the urgent task which requires the constant improvement in accordance with the current state of medical hardware and software technologies.

In the article [6], the authors describe algorithmic solutions that determine the segment contour of the region of interest in the presence of a priori information about the image structure and the level of interference. The article [2] describes the image processing method based on multifractal analysis. In articles [1], [3] and [9] are considered the methodological foundations of creating automated computer systems for the analysis of biomedical images based on computer analysis methods, including quality improvement, digitization, segmentation, identification etc.

The technology of neural networks application in particular multilayer perceptron is effective for small images usually. For images with the size 32×32 pixels the MNIST has 0.18% errors, the CIFAR-10 has 1.23% errors [10]. However, this approach cannot be used for medical images due to the huge number of required parameters. For example, an image with the size 100×100 has 10 000 pixels, which can force 100 million connections to be built for the first layer alone, while a medical image with the size 512×512 has 262 144 pixels and more than 68 billion connections in a worst-case scenario.

Convolutional neural network (CNN) are more effective for recognizing large medical images than deep neural network (DNN), because CNN is based on the partial connection of layers. CNN has much fewer parameters than DNN, which makes this approach much faster and requires less data for training [10, 11].

Analysis of popular CNN architectures has been performed using the following models: ImageNet, LeNet, AlexNet, VGGNet, GoogLeNet, ResNet. The 3.57% error rate is the best result, it is obtained by using Microsoft's ResNet. Unlike the VGGNet, the ResNet architecture has different depth variations, such as 34, 50, 101 and 152 layers. ResNet has whopping 152 layers compared to AlexNet's 8 layers, VGGNet's 19 layers and GoogLeNet's 22 layers. ResNet architecture is the stack of residual blocks. The basic idea is to

skip layers by adding connections to the neural network. Each residual block has 3x3 convolution layers. After the last conversion layer, the GAP layer is added [12].

The modified neural network learning algorithm

The developed medical image classification system is based on the separation of identification and learning tasks. One of the main tasks is the learning neural network based on available images. Keras, OpenCV and TensorFlow libraries are used for the learning neural network. The scheme of learning algorithm is presented in fig. 1.

The gradient descent algorithm for error backpropagation was improved when learning the model. A comparison of Nesterov Accelerated Gradient, Adagrad, Adam, Adamax, RMSProp and Adadelta algorithms in terms of the speed of finding a global extremum and avoiding local extrema [13] made it possible to stop using the Adam algorithm. But the study [14] proved the possibility of improving the result by switching from the Adam algorithm to the SGD algorithm. Adam algorithm approaches the solution faster at the initial stage, but as the epoch value increased, Adam algorithm avoids the local extrema more slowly. Based on this method, it was decided to abandon the standard functions and build a branching at the damping point of the approach to the extremum (point). The scheme of the algorithm is presented in fig. 1. In this scheme, the following symbols are introduced to simplify the display:

– initialization of the main input parameters: $k = 0, m_k = 0, \alpha_k = 0, \lambda_k = 0, \varepsilon = 0.001$;

– SGD algorithm parameters:

– VW_k: $v_k = \beta_1 v_{k-1} + g_k, \omega_k = \omega_{k-1} - (1 - \beta_1) \Lambda v_k$;

– MAP_k: $m_k = \beta_1 m_{k-1} + (1 - \beta_1) g_k, \alpha_k = \beta_2 \alpha_{k-1} + (1 - \beta_2) g_k^2$,

$$p_k = \frac{-\alpha_k \sqrt{1 - \beta_2^k}}{(1 - \beta_2^k)} \frac{\sqrt{m_k}}{\sqrt{\alpha_k} + \varepsilon};$$

– PG_k: $p_k^T g_k$;

– GL_k: $\gamma_k = \frac{p_k^T p_k}{-p_k^T g_k}; \lambda_k = \beta_2 \lambda_{k-1} + (1 - \beta_2) \gamma_k$;

– VL_k: $v_k = 0; \Lambda = \frac{\lambda_k}{1 - \beta_2^k}$;

– switching point between algorithms:

$$point = \left| \frac{\lambda_k}{1 - \beta_2^k} - \gamma_k \right|.$$

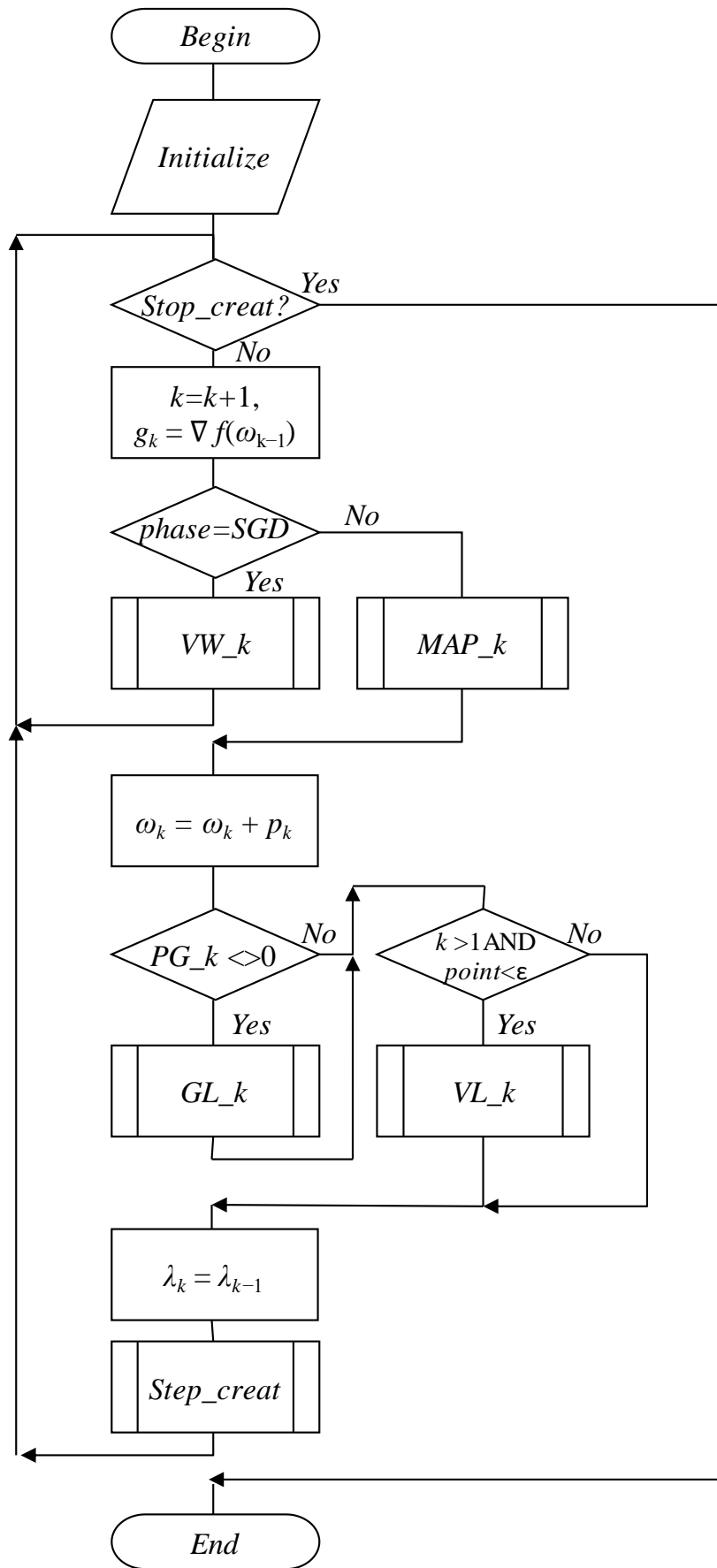


Fig. 1. Scheme of the switching algorithm between Adam and SGD

The results of applying the modified neural network learning algorithm are shown in fig. 2.

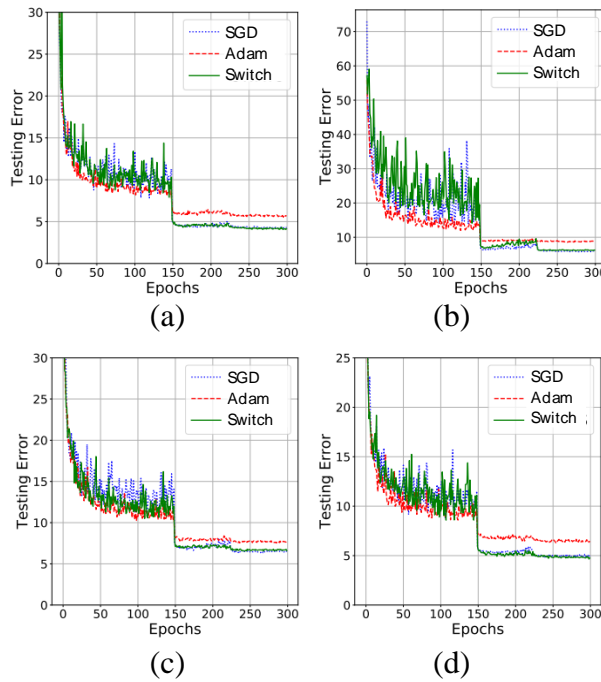


Fig. 2. Comparing the performance of the proposed algorithm on CIFAR-10 training samples for models: a) ResNet-32, b) DenseNet, c) PyramidNet, d) SENet

Applying the modified neural network learning algorithm allows to reduce the number of errors on 4.0-7.5% compared Adam algorithm. To define the effectiveness of proposed algorithms the CIFAR-10 training samples for ResNet-32, DenseNet, PyramidNet, SENet models are used.

Conclusion

Developing combination of pattern recognition methods, which is the effective to solve the problem of medical diagnostics is the one of difficult task. The proposed algorithm of learning neural network is based on switching between Adam and SGD algorithms for various input parameters. The proposed algorithm is used on test data, in particular CIFAR-10 training samples for ResNet-32, DenseNet, PyramidNet, SENet models. Applying the modified neural network learning algorithm allows to reduce the number of errors on 4.0-7.5% compared Adam algorithm while maintaining the basic rate of descent.

The proposed algorithm allows expanding the capabilities of image classification technologies and can be used to develop medical decision support systems for automated classifying diseases.

References

1. Luo J., Wu M., Gopukumar D., Zhao Y. Big data application in biomedical research and health care: a literature review. *Biomedical Informatics Insights*. 2016;8 doi: 10.4137/BII.S31559.BII.S31559
2. Rea S., Pathak J., Savova G., et al. Building a robust, scalable and standards-driven infrastructure for secondary use of EHR data: the SHARPn project. *Journal of Biomedical Informatics*. 2012;45(4):763–771. doi: 10.1016/j.jbi.2012.01.009.
3. Najarian K., Ward KR, Shirani S. Biomedical signal and image processing for clinical decision support systems. *Computational and Mathematical Methods in Medicine*. 2013; 2013:2. doi: 10.1155/2015/974592.974592
4. World Health Organization. Cancer. URL: <https://www.who.int/news-room/fact-sheets/detail/cancer>.
5. Moja L., Kwag KH, Lytras T., et al. Effectiveness of computerized decision support systems linked to electronic health records: a systematic review and meta-analysis. *American Journal of Public Health*. 2014;104(12):e12–e22. doi: 10.2105/AJPH.2014.302164.
6. Abadi M., Agarwal A., Barham P., Brevdo E., Chen Z., Citro C., Ghemawat S. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.
7. Alom MZ, Taha TM, Yakopcic C., Westberg S., Sidike P., Nasrin MS, Asari VK 2018. The history began from alexnet: A comprehensive survey on deep learning approaches. arXiv preprint arXiv:1803.01164.
8. Anisimov D., Khanova T. Towards lightweight convolutional neural networks for object detection. 2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS); IEEE; 2017. pp. 1–8. August.
9. Thimbleby H. Technology and the future of healthcare. *Journal of Public Health Research*. 2013;2(3):p. 28. doi: 10.4081/jphr.2013.e28.
10. He K., Zhang X., Ren S., Sun J. Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016:770–778.
11. Nguyen H. Fast object detection framework based on mobilenetv2 architecture and enhanced feature pyramid. *Journal of Theoretical and Applied Information Technology*. 2020; 98 (05).
12. Ojala T., Pietikainen M., Maenpaa T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2002;24(7):971–987.
13. Saad Hikmat Haji, & Adnan Mohsin Abdulazeez. (2021). Comparison of optimization techniques based on gradient descent algorithm: a review.

PalArch's Journal of Archeology of Egypt / Egyptology, 18(4), 2715-2743.

14. Keskar, NS; Socher, R. (2017) Improving generalization performance by switching from Adam to SGD. arXiv 2017, arXiv:1712.07628.

Магістрант Вернік М. О., к.т.н., доцент Олещенко Л. М.

Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»

ПРОГРАМНИЙ МЕТОД СТВОРЕННЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ З ВИКОРИСТАННЯМ МЕТАЕВРИСТИЧНОЇ ОПТИМІЗАЦІЇ

Abstract

Mykhailo Vernik, master student; Lubov M.Oleshenko, assoc. prof., PhD
Software method of creating recommendation system using metaheuristic optimization

This paper deals with creating a recommendation system using metaheuristic optimization in the recreation sphere. The modular structure of the software, the modules of which implement the presented algorithms, can facilitate the implementation of the software as a whole.

Вступ

Більшість сервісів мережі Інтернет побудовані з використанням надання рекомендацій користувачу – рекомендацій товарів (e-commerce), фільмів (Netflix), мобільних додатків (App Store, Play Market), програмного коду (GithubCopilot) тощо. Мета кожного із рішень визначається потребами конкретного бізнесу та користувачів, а також ступенем «задоволення» – збільшення / зменшення прибутку компаній та часу перебування на сайті клієнтів відповідних сервісів. Будь-яка система рекомендацій розроблена виключно для покращення таких показників, як зменшення витрат часу користувачів завдяки спрощенню вибору з великої кількості продукції, збільшення прибутку компанії завдяки збільшенню обсягу збуту товарів і послуг. Існуючі рішення можна поділити на 2 категорії: класичні та персоналізовані моделі.

Термінологія

NCF – нейро-колаборативний фільтр;

MLP – мультишаровий пересптрон;

MF – матриця факторизацій;

GWO – алгоритм оптимізації «Сірого Вовка»;

WOA – алгоритм оптимізації «Китів»;

BP – метод оберненої похибки (Back Propagation);
МОА – метаевристичні алгоритми оптимізації.

Постановка задачі

У даному дослідженні вирішується задача розроблення програмного забезпечення для створення рекомендаційних систем на основі метаевристичних методів оптимізації.

Алгоритм створення рекомендаційної моделі на основі даних користувачів

Основна ідея полягає у тому, що кожний користувач системи “Entertainment Planner” («Планувальник Розваг») залишає наступні дані:

1. Вибрана подія.
2. Кількість запрошених людей.
3. Час, витрачений на вибір події.
4. Параметри вибору користувача.
5. Психологічний портрет користувача.

Дані користувачів та подій представлені як окремі вектори даних, де кількість активних користувачів системи складає 110, а кількість подій – 5370. Користувачами було додано більше 2000 подій загалом. Також більше 10,000 даних про дії користувачів – такі, як час перебування на окремій сторінці, запрошені люди, параметри вибору, психологічний портрет. Таким чином, на виході системи більше 12,000 рядків даних про користувачів і їх взаємодію в системі. Даний датасет можна представити у вигляді 2-х датасетів для навчання та тестування, де 10,000 рядків – навчання, а 2,000 рядків – тестування. Основою рекомендацій є дані користувача, в якості рекомендацій може виступати будь-яка характеристика (наприклад, побудована нейронна мережа психологічного портрету користувачів для створення рекомендованого списку запрошень), в нашому випадку – список рекомендованих подій. В даній системі рекомендацій використано модель нейро-колаборативного фільтру [6], що є базовим методом в даній розробці, головна ідея якого полягає в поєднанні результатів загальної матриці факторизацій векторів властивостей та багатозарового персептронну, що робить метод гнучким до модифікацій рекомендацій, таким чином, пропонується створення модифікації навчання та оптимізації нейронної мережі за допомогою методів метаевристичної оптимізації (GWO, PSO, WOA, BA, GA)[1-4].

Метаевристичні методи оптимізації взаємодіють з нейронною мережею наступним чином:

1. Нейронна мережа розкладається в одновимірний вектор важелів нейронних зв'язків та важелів відхилень.

2. Вектор розкладеної нейронної мережі будемо називати генерацією, або поколінням.
3. Кожен із 5-и алгоритмів буде генерувати нове покоління.
4. Серед 5 поколінь вибирається найкраще покоління за результатами середньоквадратичної похибки фінального результату.
5. Відбувається оновлення важелів нейронних зв'язків та важелів відхилень.

На рис. 1 зображено ядро рекомендаційної системи та його взаємодія із метаевристичними методами оптимізації, що у даному випадку створює нове покоління для перенавчання / корегування вагів нейронної мережі.

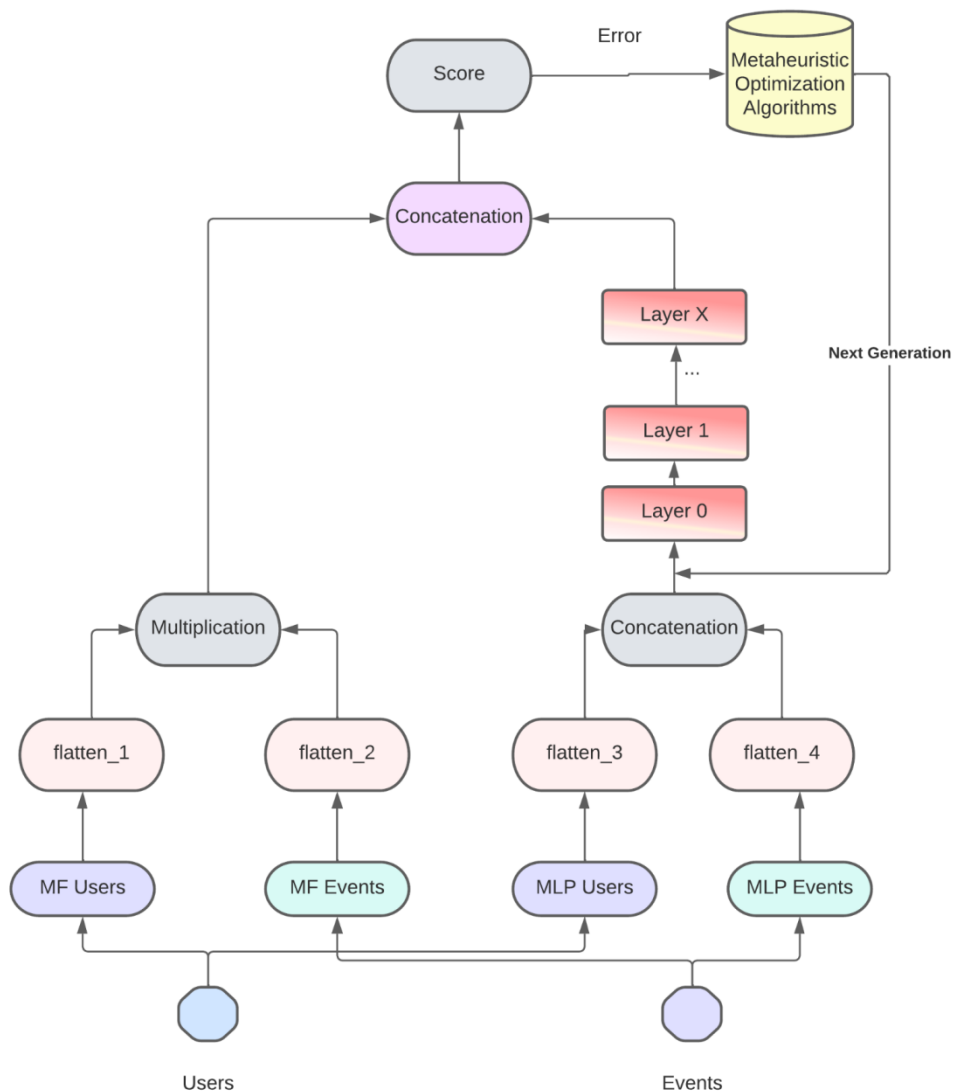


Рис. 1. Модель рекомендаційної системи.

Система на вході отримує 2 вектори – «Users» та «Events», де вектор користувачів – дані психологічних портретів, а вектор подій – дані

параметрів опису подій, відповідно, кожен із яких створює по 2 вектори:

1. MF Users (вектор матриці факторизації користувачів);
2. MF Events (вектор матриці факторизації подій);
3. MLP Users (вектор багат шарового перцептрон користувачів);
4. MLP Events (вектор багат шарового перцептрон подій).

Після обробки та розкладання векторів матрицю факторизації (MF) сформулюють вектори «MF Users» та «MF Events», у той же час в багат шаровий перцептрон (MLP) використаються об'єднані вектори «MLP Users» та «MLP Events». На етапі отримання результатів як з матриці факторизації, так і з MLP, вони об'єднуються у вихідний результат, який буде відправлено в множину методів метаевристичної оптимізації, з яких буде вибрано найкраще покоління для отримання генерації нового покоління (оновлення вагів нейронної мережі), а також знаходження помилки відносно заздалегіть відомого результату вибору.

Загальний алгоритм реалізації системи виглядає наступним чином:

1. Побудова “дерева структури подій”.
2. Введення даних користувачів системи.
3. Застосування метаевристичних алгоритмів оптимізації.
4. Персоналізована рекомендаційна система з підбором подій та людей.

На рис. 2 зображено приклад скороченої структури подій в рекреаційній сфері.

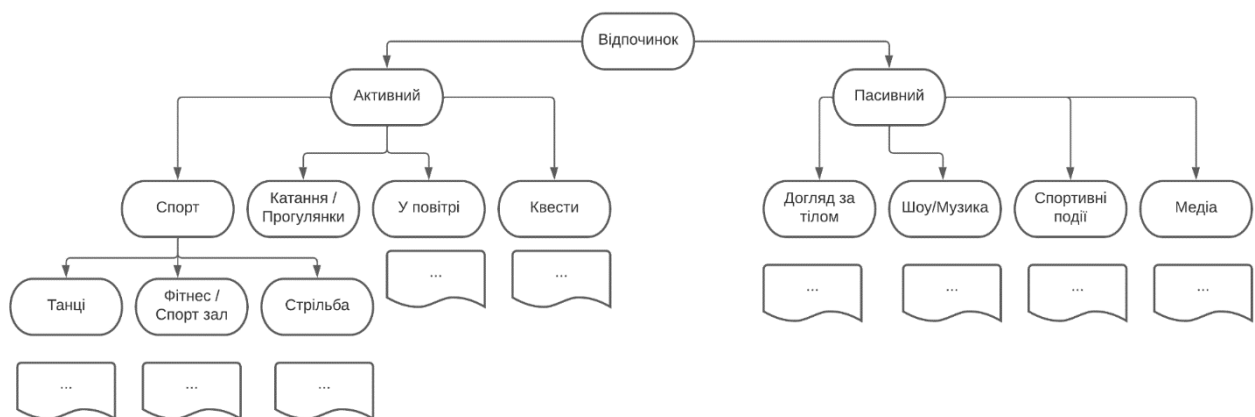


Рис. 2. Дерево структури подій.

Дані користувачів були описані вище, на етапі збору даних для системи. Множина метаевристичних алгоритмів оптимізації включає в себе 5 алгоритмів оптимізації: GWO, WOA, PSO, BA, GA.

Алгоритм Whale Optimization

Метаевристичний алгоритм оптимізації Whale Optimization Algorithm

(WOA) імітує мисливську поведінку горбатих китів. Основна відмінність між проведеним дослідженням і недавно опублікованими статтями авторів (зокрема, GWO [1]) полягає в моделюванні мисливської поведінки з випадковим або кращим пошуковим агентом для переслідування здобичі і використанням спіралі для моделювання пухирчастої сітки [2].

Експлуатаційна модель:

$$D^- = |C^- \cdot X_p^-(t) - X^-(t)|, \quad (1)$$

$$X^-(t + 1) = \begin{cases} \vec{X}(t) - \vec{A} \cdot \vec{D}, & p < 0.5 \\ \vec{D} * e^{bt} \cos \cos(2\pi t) + \vec{X}^*(t), & p \geq 0.5 \end{cases} \quad (2)$$

Пошукова модель:

$$D^- = |C^- \cdot X_{rand}^- - X^-|, \quad (3)$$

$$X^-(t + 1) = |C^- \cdot X_{rand}^- - A^- \cdot D^-| \quad (4)$$

Алгоритм Genetic Algorithm

Генетичні алгоритми засновані на принципі природного відбору. Генетичні (або еволюційні) методи реалізують процес, пов'язаний з біологічною еволюцією, включають генерацію, рекомбінацію та мутацію хромосом, створення нової популяції з кращими генами та покращеною пристосованістю. У таких методах припускається, що змінні моделі відповідають хромосомам популяції і вони намагаються через деяку кількість поколінь отримати найкращі хромосоми, що представляють оптимальне рішення проблеми.

Алгоритм Particle Swarm Optimization

Оптимізація зграї частинок базується на поведінці птахів, які збираються навколо джерел їжі. Коли один птах бачить або відчуває запах їжі, він голосно цвірінькає, а інші птахи розгойдуються в його напрямку. Якщо будь-який інший птах наближається до їжі першим, він щебече голосніше, і інші птахи повертаються до нього. Цей процес триває до тих пір, поки один із птахів не приземлиться до їжі. У контексті оптимізації розташування їжі вказує на оптимальний вектор моделі, який можна використовувати для визначення оптимального значення цільової функції:

$$X^{i-}(t + 1) = X^{i-}(t) + V^{i-}(t + 1), \quad (5)$$

$$V_k^{i-}(t + 1) = w * V_k^{i-}(t) + c_1 * r_2 (\overline{p_{k,best}^i} - \overline{p_k^i}(t)) + c_2 * r_2 (\overline{g_{best}^i}(t) - \overline{p_k^i}(t)) \quad (6)$$

Розроблено рекомендаційну систему з матрицею факторизацій, де дані користувачів представляють собою психологічний портрет, а дані подій – параметри опису подій після співставлення обох матриць (рис.3).

	Fun	Relax	Intellectual	Sport	...	0.33	0.12	0.77	0.89
	Relax	0.1	0.5	0.42	0.17
	0.33	0.12	0.75	0.1	...	0.44	0.14	0.62	0.88
	0.6	0.2	0.3	0.5	...	0.33	0.72	0.47	?
	0.22	0.77	0.02	0.35	...	0.82	0.18	0.55	0.61
	0.53	0.5	0.13	0.22	...	0.18	0.55	0.1	0.25

Рис. 3. Матриця факторизацій.

Результати порівняння використання класичного методу оберненої похибки (MLP + BP, класичний метод є аналогічним запропонованому за виключенням множини метаевристичних алгоритмів оптимізації, замість якої використовується метод зворотного поширення помилки) та запропонованого методу надання рекомендацій за допомогою метаевристичної оптимізації (MLP + MOA) наведено в табл. 1. Запропонований метод має перевагу за рахунок того, що метаевристичні алгоритми оптимізації надають нейронній мережі більший пошуковий простір і мають можливість виходити з локальних пасток, на відміну від використання таких класичних методів, як стохастичний градієнт.

Таблиця 1

Порівняння методів

Метод	Точність
<i>Класичний (MLP + BP)</i>	76.68%
<i>Запропонований (MLP + MOA)</i>	81.37%

Висновки

У даному дослідженні розглянуто алгоритми, які дозволяють точніше створювати рекомендаційні системи, а отже, й покращувати конверсію користувачів та їхню взаємодію з тією чи іншою системою на основі отриманих даних, що, в свою чергу, дозволяє покращити точність аналізу даних для отримання цільової множини користувачів системи. Запропонований програмний метод забезпечує зменшення складності обчислень та забезпечує більшу точність у порівнянні з класичною реалізацією в середньому на 4-6%. Модульна структура програмного

забезпечення, модулі якої реалізують вищезазначені алгоритми, полегшує реалізацію запропонованого програмного методу в цілому.

Література

1. Mirjalili, Seyedali & Mirjalili, Seyed & Lewis, Andrew. (2014). Grey Wolf Optimizer. *Advances in Engineering Software*. 69. 46–61. [10.1016/j.advengsoft.2013.12.007](https://doi.org/10.1016/j.advengsoft.2013.12.007).
2. M. Cells, B. Rylander, Neural network learning using particle swarm optimization, *Advances in Information Science and Soft Computing*, 2002, pp. 224– 226.
3. Y. Tian, S. Peng, X. Zhang, T. Rodemann, K. C. Tan and Y. Jin, "A Recommender System for Metaheuristic Algorithms for Continuous Optimization Based on Deep Recurrent Neural Networks," in *IEEE Transactions on Artificial Intelligence*, vol. 1, no. 1, pp. 5-18, Aug. 2020, doi: [10.1109/TAI.2020.3022339](https://doi.org/10.1109/TAI.2020.3022339).
4. Mirjalili, S. and A. Lewis. "The Whale Optimization Algorithm." *Adv. Eng. Softw.* 95 (2016): 51-67.
6. L.D. Mech, "Alpha status, dominance, and division of labor in wolf packs", *Canadian Journal of Zoology*, vol. 77, pp. 1196- 1203, 1999.
7. Vasileios Perifanis, Pavlos S. Efraimidis, Federated Neural Collaborative Filtering, *Knowledge-Based Systems*, Volume 242, 2022, 108441, ISSN 950-7051, <https://doi.org/10.1016/j.knosys.2022.108441>.

Магістрант Мороз М.В., старш. викл. Сущук-Слюсаренко В.І.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

МЕТОД МОДЕЛЮВАННЯ ПРИМІЩЕНЬ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ ДОПОВНЕНОЇ РЕАЛЬНОСТІ

Abstract

Mykhailo Moroz, student; senior lecturer Suschuk-Sliusarenko V.I.

Method for Room Modeling Based on AR-Technology

Room modeling and interior design is a delicate and complex task. The use of a special method, which is based on augmented reality technologies, allows to significantly improve the efficiency and resource consumption of modeling tasks. The claim is based on the fact that the use of such technology introduces the extension of the real world with virtual objects, images, models, sounds, etc. Interaction of real objects with virtual ones and vice versa is possible, the general possibilities of such interactivity are limitless. The purpose of this work is to analyze basic and advanced methods of using augmented reality technologies when applying them to modeling tasks, implementing and developing one's own improved method with an overview of the tasks to be solved.

Вступ

Моделювання приміщень та дизайн інтер'єру є творчим і складним завданням. Використання методу, який ґрунтується на технології доповненої реальності, дозволяє покращити якість та зменшити ресурсозатратність 3D-моделювання. Використання технології доповненої реальності робить можливим розширення реального світу віртуальними об'єктами, зображеннями, моделями, звуками тощо. Ця робота має на меті аналіз методів використання технології доповненої реальності та розроблення вдосконаленого методу, спрямованого на застосування технології доповненої реальності для розв'язання задачі моделювання приміщень.

У контексті цієї роботи доповнена реальність буде розглядатись згідно із цим визначенням [1] про систему, яка:

- поєднує віртуальні та реальні об'єкти;
- дозволяє взаємодіяти з віртуальними об'єктами у реальному часі;
- дозволяє працювати з 3D-об'єктами.

Базовою ідеєю використання доповненої реальності є поєднання віртуальних об'єктів із зображенням реального середовища та об'єктів у ньому, а також реалістична інтеграція віртуальних об'єктів до реального середовища. Варто зазначити, що доповнена реальність дозволяє виконувати маніпуляції та впливати на об'єкти системи. Впровадження використання такої технології для процесів моделювання є вкрай ефективним [2].

Постановка задачі

Задача полягає у локалізації та побудові мапи моделі навколишнього простору і положення пристрою (Simultaneous Localization and Mapping, SLAM) [3] з метою автоматичного обчислення, побудови і оновлення мапи невідомого оточення з одночасним відстеженням місцеположення віртуального об'єкта у процесі руху. SLAM являє собою методологію для вирішення двох задач: побудови мапи дослідженого простору та побудови траєкторії руху пристрою на мапі.

Для вирішення поставленої задачі у дослідженні використовуються елементи реалізації деяких існуючих алгоритмів вирішення задачі локалізації та мапінгу: алгоритм розширеного фільтру Калмана та алгоритм DP-SLAM.

Розширений фільтр Калмана

Фільтр Калмана – рекурсивний фільтр, що оцінює вектор стану динамічної системи, використовуючи послідовність неповних і зашумлених вимірювань. Фільтр Калмана призначений для рекурсивного дооцінювання вектора стану апріорно відомої динамічної системи, тобто для розрахунку поточного стану системи необхідно знати поточне вимірювання, а також попередній стан самого фільтра. Стан фільтра визначається двома змінними:

X_k – оцінка вектора стану динамічної системи в момент часу k ;

P_k – коваріаційна матриця помилок (міра точності оцінювання вектора стану) в момент часу k .

Розширений фільтр Калмана (EKF) є схожим на простий фільтр Калмана, за тим винятком, що він може бути використаний в нелінійних процесах. Застосування EKF – це один з найбільш поширених методів рішення задачі SLAM. Він дозволяє не тільки уточнювати оцінку положення об'єкта на мапі, але і положення всіх виявлених орієнтирів. Зазвичай процес оцінки стану системи, в контексті SLAM, розбивають на три етапи: оновлення оцінки стану системи на основі одометричних даних, оновлення оцінки стану системи на основі повторно виявлених орієнтирів, додавання нових орієнтирів в систему.

Алгоритм DP-SLAM

DP-SLAM використовує дані лазерного далекоміра і фільтр частинок для зберігання гіпотез про позицію об'єкта і конфігурацію навколишнього середовища [5]. Фільтр частинок (послідовний метод Монте-Карло) – рекурсивний алгоритм для чисельного вирішення проблем оцінювання (фільтрації, згладжування), особливо для нелінійних випадків.

Алгоритм ґрунтується на тому, що в фільтрі аналізується безліч гіпотез про поточну позицію об'єкта. У початковий момент генерується випадковий набір гіпотез. В ході роботи алгоритму деякі з них будуть відсіюватися через невідповідності вторинними ознаками системи, які можуть бути виміряні більш точно. Після завершення циклу сканування вибирається найбільш вірогідна гіпотеза.

Для зберігання і структурування великого числа проміжних мап у дослідженні реалізована деревоподібна структура. Вся інформація, необхідна для вирішення обох підзадач, локалізації і коригування мапи, оброблюється одним з фільтрів частинок [6].

Запропонований метод

Запропонований метод ґрунтується на одночасному використанні модифікованих алгоритмів фільтру Калмана та DP-SLAM. Для обчислення задачі локалізації застосовується вибіркоче використання цих методів, базуючись на кількості та якості орієнтирів та середовища.

На першому етапі необхідно обчислити коефіцієнт Калмана для різних точок-орієнтирів для визначення та вибору оптимального методу побудови мапи та локалізації. Визначення динамічних точок: якщо деяка точка x_i у полі зору камери нерухома, то її проекції $x_{i,j} \in R^2$ задовольняють рівнянню

$$\pi(\theta_{c,j}, X) = x_{i,j} + v_{i,j},$$

де $v_{i,j}$ - випадковий шум у спостереженнях;

$\{\theta_{c,j}\}$ - положення камери у моменти часу $j = t_{i,1}, \dots, t_{i,n_i}$.

Точка класифікується як статична, якщо виконана умова щодо середньої похибки перепроєктування:

$$\exists X_i \in R^3 : \frac{1}{n_i} |\pi(\theta_j, X_i) - x_{i,j}| < \varepsilon, \quad (1)$$

для деякого параметра похибки ε .

Якщо зміщення орієнтирів має високу похибку, у порівнянні з показниками датчиків пристрою, обраний орієнтир не є достовірним, тому коефіцієнт Калмана буде низьким. Навпаки, якщо зміщення орієнтирів

обчислюється точно в порівнянні з показниками датчиків, то коефіцієнт Калмана буде високим.

На другому етапі залежно від кількості точок з коефіцієнтом, що задовольняють умові (1) про статичну точку, обираємо метод локалізації. Якщо кількість точок задовольняє певний поріг якості D , то використовується алгоритм реалізації розширеного методу Калмана; якщо кількість точок нижча за поріг якості, то використовується алгоритм DP-SLAM, який працює ефективніше для нестатичних середовищ. Поріг якості D може встановлюватися довільно як конфігураційна змінна. Реалізований алгоритм DP-SLAM одночасно підтримує до 9000 можливих гіпотез і відкидає ті з них, які в силу невиконання умов про статичну точку точно не можуть відповідати реальному стану системи. У такому алгоритмі для зберігання і структурування великого числа проміжних мап у цьому дослідженні реалізована деревоподібна структура, яка відображає усю інформацію, необхідну для вирішення обох підзадач, локалізації і коригування мапи. Також важливою інноваційною особливістю програмної системи, що реалізує запропонований підхід, є використання єдиної кодової бази реалізації методу, для застосування на основних платформах мобільних девайсів IOS та Android. Реалізація методу виконана мовою JS, що виконується всередині JS VM, що виконує JavaScriptCore. На iOS вона надається системою; на Android додаток використовує її у вигляді бібліотеки. Таким чином, єдина кодова база дозволяє імплементувати кросплатформну реалізацію методу на обох платформах.

Висновки

Таким чином, у цьому дослідженні запропоновано підхід до модифікації та спільного використання існуючих методів для розв'язання задач локалізації та побудови мапи середовища електронних документів з метою забезпечення високої ефективності при інтеграції у систему доповненої реальності. Запропоновано алгоритм побудови доповненої реальності з використанням модифікації DP-SLAM та одночасним використанням фільтру Калмана, що дозволяє розміщувати віртуальні об'єкти у невідомому середовищі без використання заздалегідь заданих міток.

Перспективним напрямком для продовження роботи над тематикою цієї статті є дослідження ефективності реалізованої модифікації методів для неоднорідних або нестатичних середовищ.

Література

1. Tolga Kilic (2019), Investigation Of Mobile Augmented Reality Applications Used In The Interior Design, pp. 304-305.
2. Understanding Augmented Reality: Concepts and Applications [Електронний ресурс] — Режим доступу: https://www.researchgate.net/publication/277306153_Understanding_Augmented_Reality_Concepts_And_Applications
3. Dan Munnerlay (2007), Toward's Real-Time Markerless Tracking Wizard Lens on Paper Maps, pp. 56-75.

К.т.н., доцент Онай М.В., студент Гришко Ю.І.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

МЕТОД МНОЖЕННЯ ТОЧКИ ЕЛІПТИЧНОЇ КРИВОЇ НА СКАЛЯР У СКІНЧЕННОМУ ПОЛІ

Abstract

Mykola Onai, PhD; Hryshko Yurii, student

Method multiplying point of elliptic curve by scalar in finite field

This paper concerns the task of analysis and modification of existing methods multiplying point of elliptic curve by scalar in finite field. The classical methods are studied and discussed. The modified method is proposed. The comparative analysis of efficiency of both the classical and the modified methods is fulfilled. The ways for further research are proposed as well.

Вступ

Задача забезпечення конфіденційності інформації та захист її від зловмисників стає дедалі складнішою і в той же час найбільш актуальною.

Більшість систем захисту інформації будується на основі асиметричних криптосистем. Найбільш обчислювально витратною операцією в еліптичних криптосистемах є операція множення точки еліптичної кривої на скаляр. У даній статті проводиться порівняльний аналіз швидкодії існуючих методів множення точки еліптичної кривої на скаляр у скінченному полі, а також пропонується модифікація визначеного найшвидшого із наведених методів, яка дозволяє виконувати операцію множення швидше.

Постановка задачі

Задача полягає в модифікації існуючого найшвидшого методу множення точки еліптичної кривої на скаляр у скінченному полі, для поліпшення часових характеристик алгоритму.

Аналіз існуючих методів множення точки еліптичної кривої на скаляр у скінченному полі

Double-and-add

Цей алгоритм вимагає $\log_2(d)$ ітерацій подвоєння та додавання точки для обчислення повного множення точки.

Даний алгоритм представлено на рис. 1.

```
let bits = bit_representation(s) # the vector of bits (from LSB to MSB) representing s
let res =  $\mathcal{O}$  # point at infinity
let temp = P # track doubled P val
for bit in bits:
    if bit == 1:
        res = res + temp # point add
        temp = temp + temp # double
return res
```

Рис. 1. Алгоритм double-and-add

Windowed method

Цей алгоритм має таку саму складність, що й підхід *double-and-add*, з перевагою використання меншої кількості додавання точок.

Даний алгоритм представлено на рис. 2.

```
Q ←  $\mathcal{O}$ 
for i from m to 0 do
    Q ← point_double_repeat(Q, w)
    if  $d_i > 0$  then
        Q ← point_add(Q,  $d_i P$ ) # using pre-computed value of  $d_i P$ 
return Q
```

Рис. 2. Алгоритм windowed method

Sliding-window method

Недолік цього алгоритму полягає в тому, що етап попереднього обчислення приблизно вдвічі складніший, ніж *windowed method*.

Цей алгоритм представлено на рис. 3.

```
Q ←  $\mathcal{O}$ 
for i from m downto 0 do
    if  $d_i = 0$  then
        Q ← point_double(Q)
    else
        t ← extract j (up to w - 1) additional bits from d (including  $d_i$ )
        i ← i - j
        if j < w then
            Perform double-and-add using t
            return Q
        else
            Q ← point_double_repeat(Q, w)
            Q ← point_add(Q, tP)
return Q
```

Рис. 3. Алгоритм sliding-window

wNAF

У цьому методі ми прагнемо використати той факт, що віднімання точок так само легко, як і додавання точок, щоб виконати менше, порівняно з методом *sliding-window method*.

Цей алгоритм представлено на рис. 4.

```

i ← 0
while (d > 0) do
  if (d mod 2) = 1 then
    di ← d mod 2w
    d ← d - di
  else
    di = 0
  d ← d/2
  i ← i + 1
return (di-1, di-2, ..., d0)

```

Рис. 4. Алгоритм wNAF

Montgomery ladder

Цей метод обчислює значення точки за фіксований проміжок часу. Це може бути корисним, коли вимірювання часу або енергоспоживання піддаються впливу зломисника, який виконує атаку по бічному каналу.

Даний алгоритм представлено на рис. 5.

```

R0 ← 0
R1 ← P
for i from m downto 0 do
  if di = 0 then
    R1 ← point_add(R0, R1)
    R0 ← point_double(R0)
  else
    R0 ← point_add(R0, R1)
    R1 ← point_double(R1)
return R0

```

Рис. 5. Алгоритм Montgomery ladder

У табл. 1 подано дані часу роботи представлених методів.

Таблиця 1

Час роботи методів

Виконання %	Windowed, мс	Sliding-window, мс	Double-and-add, мс	w-ary non-adjacent form(wNAF), мс	Montgomery ladder, мс
0,39	0,99	0,98	0,99	0,98	0,99
1,56	0,99	0,99	1	0,99	1
6,25	0,98	1,03	1,05	1,02	1,04
12,5	1,04	1,09	1,05	1,09	1,05
18,75	1,06	1,14	1,07	1,11	1,04
25	1,08	1,19	1,1	1,19	1,09
31,25	1,11	1,26	1,14	1,24	1,12
37,5	1,13	1,31	1,15	1,29	1,14
43,75	1,15	1,37	1,18	1,34	1,16
50	1,14	1,42	1,24	1,38	1,21
56,25	1,18	1,47	1,24	1,42	1,2
62,5	1,2	1,52	1,26	1,46	1,21
68,75	1,22	1,58	1,29	1,51	1,23
75	1,24	1,63	1,31	1,56	1,25
81,25	1,26	1,69	1,34	1,62	1,29
87,5	1,27	1,73	1,36	1,66	1,3
93,75	1,28	1,79	1,39	1,71	1,33
100	1,31	1,84	1,41	1,75	1,34

Порівняння часу роботи методів представлено на рис. 6.

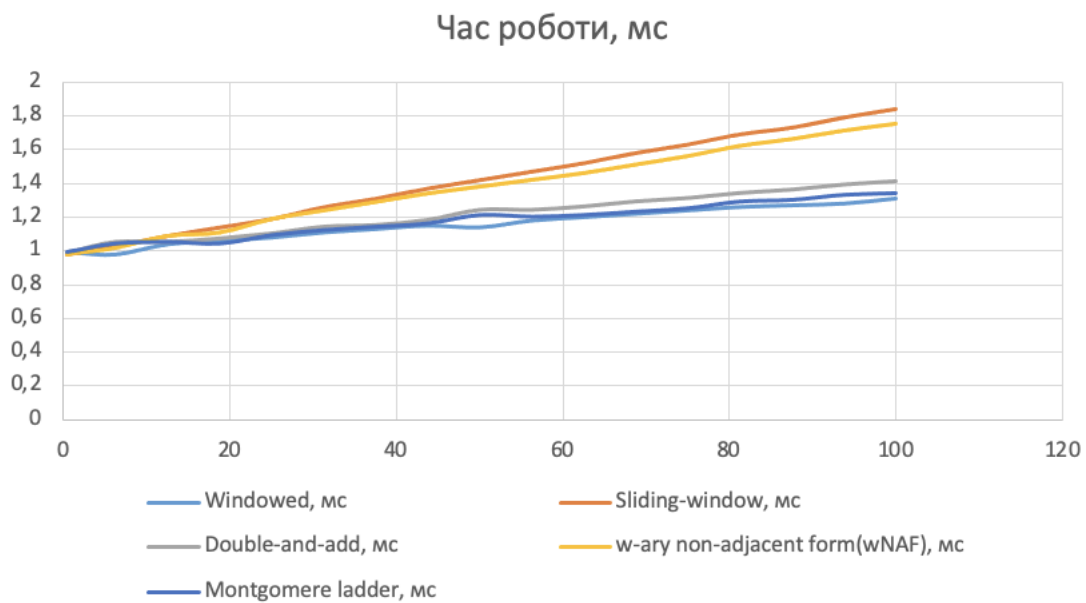


Рис. 6. Порівняння методів за часом роботи

Можемо зробити висновок, що найшвидший метод з усіх – *Windowed*.

Модифікація віконного методу

Статистично показано, що двійкові подання чисел довжиною понад 1000 біт містять довгі послідовності нулів. Тому, дана модифікація віконного методу буде виділяти вікно зі старшим одинарним бітом та іншими нульовими.

Таблицю передобчислень для $w = 5$, де P – точка еліптичної кривої, зображено на рис. 7.

№ елементу	Множник	Значення точки
0	2^0	$1 \cdot P$
1	2^1	$10 \cdot P$
2	2^2	$100 \cdot P$
3	2^3	$1000 \cdot P$
4	2^4	$10000 \cdot P$

Рис. 7. Вигляд таблиці передобчислень

У табл. 2 подано дані часу роботи звичайного та модифікованого методу.

Таблиця 2

Час роботи звичайного та модифікованого методу

Виконання %	Windowed, мс	Модифікований Windowed, мс
0,39	0,99	0,98
1,56	0,99	0,92
6,25	0,98	0,97
12,50	1,04	1,00
18,75	1,06	1,00
25,00	1,08	1,01
31,25	1,11	1,03
37,50	1,13	1,05
43,75	1,15	1,06

Продовження табл. 2

50,00	1,14	1,09
56,25	1,18	1,10
62,50	1,2	1,13
68,75	1,22	1,16
75,00	1,24	1,19
81,25	1,26	1,20
87,50	1,27	1,22
93,75	1,28	1,26
100,00	1,31	1,29

Порівняння часу роботи звичайного та модифікованого windowed методу представлено на рис. 8.

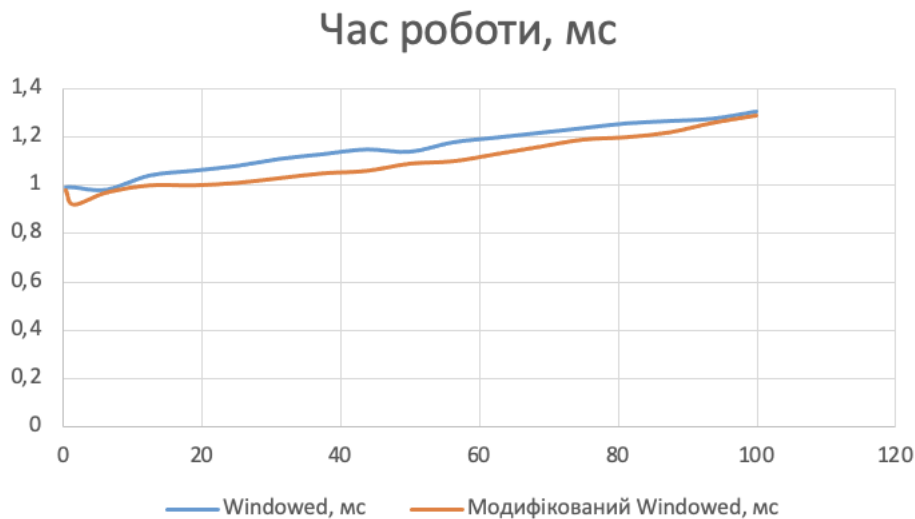


Рис. 8. Порівняння звичайного та модифікованого windowed методу за часом роботи

При використанні даної модифікації час виконання методу зменшився.

Висновки

Проведений аналіз найшвидшого існуючого і модифікованого методу множення точки еліптичної кривої на скаляр у скінченному полі показав, що запропонована модифікація дає вищу швидкодію, при правильному виборі довжини вікна та побудові на початковій стадії таблиці передобчислень.

Література

1. V. S. Dimitrov, L. Imbert, and P. Mishra. Fast elliptic curve point multiplication using double-base chains. 2005.
2. Василенко О. Н., Теоретико-числовые алгоритмы в криптографии/ Василенко О. Н.// М.: МЦНМО, 2003. – 178 с.
3. N.Koblitz, Elliptic Curve Cryptosystems [Text]/ Neal Koblitz// Mathematics of Computation – Volume 48 – Number 177 – January 1987 – pp. 203-209.

UDC 004.056.5:004.8

Volodymyr Pogorelov, PhD, Assoc. Prof.; Wenjiao Shi, student

National Technical University of Ukraine
“Kyiv Polytechnic Institute”

THE METHOD FOR DDOS ATTACKS RECOGNITION BASED ON ARTIFICIAL INTELLIGENCE

Abstract

К.т.н. Погорелов В.В., магістрант Ши Венцзяо

Метод розпізнавання DDos-атак на основі штучного інтелекту

Метою цієї статті є побудова простору ознак, що базується на основі кодів поведінки хоста та кодів функцій, щоб представити агреговані групи, пов'язані з операційними факторами та факторами поведінки програмного забезпечення в мережевому середовищі. Оскільки поведінка хоста та характеристики пакетів є хаотичною системою, що визначається багатьма факторами, для відстеження ненормальної поведінки в цій статті використовується побудова фазового простору для аналізу розміру області змін як вектора ознак часової області, щоб точніше визначити чи є поведінка хоста ненормальною в усіх варіантах простору ознак. Таким чином даний метод дозволяє краще й точніше попереджати про DDos-атаки.

Introduction

Distributed Denial of Service (DDoS) has always been a key research object in the field of network security, which seriously threatens the development of network security in daily work and has a huge impact on the network environment. DDoS attack refers to the use of client/server technology to unite multiple computers as an attack platform to launch DDoS attacks against one or more targets, thereby multiplying the power of denial-of-service attacks. When a network device suffers a DDoS attack, the most obvious phenomenon is that the network device receives a large number of unknown packets and data streams. For enterprises, a company's network egress usually corresponds to a single IP, if there is an abnormal host in the enterprise, it may affect other users, so the enterprise security gateway needs to suppress and warn of abnormal behavior. However, the business traffic within an enterprise is very diverse, and it is difficult to accurately and effectively filter abnormal traffic, and it is impossible to distinguish normal sudden traffic. The current traffic inspection method is mainly based on deep packet inspection (D e e p Packet Inspection, DPI), which can only determine the type of traffic, and has limitations in the analysis of the specific behavior of the client, so it cannot effectively determine whether the network equipment of the enterprise has

suffered a DDoS attack.

Research Objective

The main objective of the research is to solve the problem that existing methods cannot effectively judge whether an enterprise's network equipment has suffered a DDoS attack, this paper provides a network security early warning method based on artificial intelligence for DDoS attack identification

Method description

The ONE-Hot code [1] of the department corresponding to each host is merged with the ONE-Hot code of the browser fingerprint corresponding to each host to obtain the host behaviour code $V = Concatenate(V_A, V_B)$ where $Concatenate$ is the merge function, V_A is the One-hot code corresponding to the department of the host, and V_B is the one-hot code corresponding to the browser fingerprint of the host.

For the communication packets of any host in the target enterprise, the bag-of-words model is used to segment the communication packets, obtain all the words corresponding to the communication messages of the host, and exclude common words and words that are repeated due to common protocols in the field. Use the CountVectorizer technology to convert the remaining vocabulary after the above exclusion to obtain the packet feature code corresponding to the host. And then, according to the host behavior code and message signature code corresponding to each host corresponding to each host to calculate the local change domain of each host corresponding to each period. For any period of any host, calculate the neighborhood distance between it and the target period, and the target period is any period of the host or any period of any other host; Taking any period of any host as a sample, the K-local variation domain corresponding to any sample is calculated, and the process is as follows:

Based on the neighborhood distance, there is a K-reachable P_k distance and it radiates one sample outward in hypothetical space until it covers the distance when the Kth neighboring sample is covered. All samples y within the K-th reachable distance P_k of sample x form set $H_k(x)$ and the local variation domain corresponding to each sample is calculated according to the following formula:

$$J(x) = \frac{K}{\sum_{y \in H_k(x)} P(x, y)} \quad (1)$$

Among $J(x)$ them, is the local variation domain of a certain period x corresponding to a certain host, K is the number of adjacent samples, y is any $H_k(x)$

adjacent sample, and P_x is the set of adjacent samples of a certain period x corresponding to a certain host.

Next, use the following formula to calculate the neighbourhood distance between it and the target period:

$$P_{x,y} = (2 - \cos(F_x, F_y)) + \text{HAMMING}(V_x, V_y) \quad (2)$$

Where $P_{x,y}$ is the neighborhood distance F_x between a certain period x corresponding to a host and the target period y , F_y is the message signature code corresponding to a host, V_x is the message signature code of a certain period x corresponding to a host, V_y is the message signature code of the target period y , and HAMMING is the Hamming distance [3], and \cos is the cosine similarity.

According to the local variation domain of each period corresponding to each host, the standard deviation of the tracking index corresponding to each observation time period in the use process of each host is calculated, and one observation period includes multiple periods; According to the standard deviation of the tracking index corresponding to the observation time of each segment, the network behavior anomaly index corresponding to the analysis time of each segment is calculated; Determine whether the host has increased the abnormal network behavior index for more than two consecutive times during use, and if so, determine that the host has launched a DDoS attack and give an early warning.

The local change domain of each period of the host is recorded according to the chronological order, and the corresponding historical change domain of the host is constructed:

$$E = \{D_{\text{Reach}_{t-N}}, D_{\text{Reach}_{t-N+1}} \dots D_{\text{Reach}_t}\} \quad (3)$$

The D_{Reach_t} is the local variation domain of the current cycle of the host, $D_{\text{Reach}_{t-N}}$ is local variation domain of the first N cycles of the current cycle of the host, $D_{\text{Reach}_{t-N+1}}$ and is the local variation domain of the first $N-1$ cycles of the current cycle of the host

After a single acquisition, set the $t=0$ moment of the benchmark analysis and record the domain size of N parts:

$$E = \{D_{Reach_0(1)}, D_{Reach_0(2)} \dots D_{Reach_0(N)}\} \quad (4)$$

N is the number of samples sampled from history, that is, the size of the local variable domain since the present $D_{Range(1)}$ take a total of N forwards; The delay time parameter τ is selected by the mutual information method, and the embedded dimension parameter m is selected by the false proximity point method, and the phase space reconstruction method is as follows:

$$\begin{aligned} W_{0(1)} &= [D_{Reach_0(1)}, D_{Reach_0(1+\tau)}, \dots, D_{Reach_0(1+(m-1)\tau)}] \\ &\dots \\ W_{0(N)} &= [D_{Reach_0(N)}, D_{Reach_0(N+\tau)}, \dots, D_{Reach_0(N+(m-1)\tau)}] \\ &\dots \\ W_{0(N-(m-1)\tau)} &= [D_{Reach_0(N+(m-1)\tau)}, D_{Reach_0(N+(m-2)\tau)}, \dots, D_{Reach_0(N)}] \end{aligned} \quad (5)$$

At this point, the phase space composed of N local variation domains [2] is reconstructed, and the phase space is used as the reference phase w_0 space

Update the data in the form of a continuous sliding window to record the historical variation domain $E = \{D_{Reach_{t-\tau}}, D_{Reach_{t-2\tau}} \dots D_{Reach_{t-m\tau}}\}$ composed of N local change domains when the next moment comes, using the same delay time τ as the reference phase space and w_0 the embedded dimension m to reconstruct the phase space of the historical variation domain composed of N local change domains at t time: in subsequent recordings, reconstruct the current phase space in the same way as above, then get the phase space W that changes in real time.

For a vector $W_t(n)$ in the phase space at moment t, look $p(p > 2(t * m))$ for closest $\{W_{(k)}, W_{(k+1)}, \dots, W_{(k+p-1)}\}$ vector in the reference $k = \{1, \dots, N - (m-1)\tau\}$ phase space, based on the above processing method, all data D_{Reach} points are updated with each sample, and the network behaviour anomaly index B is obtained by taking T sampling as the length of the observation time window.

For phase space, there are:

$$\begin{aligned} A_k &= \{W_{0(k+t)} \dots W_{0(k+p+t-1)}\} \\ B_k &= \{\{W'_{0(k+t-1)}, \dots, W'_{0(k)}\} \dots W'_{0(k+p+t-2)}, \dots, W'_{0(k+p-1)}\} \end{aligned} \quad (6)$$

Design $w_{t(0)}$ trace function to predict phase space:

$$P_{(n)} = W_{(n+t)} - ((W'_{n+t-1}, \dots, W'_{(n)}) \frac{A_k}{B_k})' \quad (7)$$

Calculate the distance between w_0 and the vector $W_{0(k)}, W_{0(k+1)}, \dots, W_{0(k+p-1)}$ the farthest distance is Dr_n the phase space weight $q(n) = (Dr_n^m)^{-1}$ is set, and

$$P(n) \quad n = N - (m-1)t$$

m is the correlation dimension of the phase space at t. Let n increment by 1 and continue calculating until . Then, all tracking functions corresponding to all vectors in the t-time phase space are used and the tracking metrics for the t-time phase space are calculated:

$$P_t = \sqrt{\frac{\sum_{n=1}^M q(n) / P_{t(n)}^2}{\sum_{n=1}^M q(n)}} \quad (8)$$

Calculation of P_1, P_2, \dots, P_T Mean P_T and standard deviation σ_P of T tracking indicators.

Therefore, the variance σ_P of P is calculated by counting the P value of the host for M consecutive times, which indicates that the behavior of the host is difficult to be continuously and stably represented in the phase space. The network behavior anomaly index corresponding to each host is calculated as follows:

$$B = \frac{Max(\sigma_{Pr})}{Min(\sigma_{Pr})} \quad (9)$$

So far, the network behavior anomaly index B is obtained.

Conclusion

This paper constructs the host behavior code and message signature code based on the browser fingerprint, department and communication message information corresponding to each host of the target enterprise. According to the host behavior code and packet signature code during the use of each host, it can more effectively judge whether the host has launched a DDoS attack, when the network behavior anomaly index B corresponding to a host becomes Q times larger continuously, it means that the phase space can no longer track the behavior, that is, in the feature space composed of the host behavior code and the message signature code, the host behavior code is different or the message signature code is different, and the duration is different, resulting in a large deviation of the phase space prediction result P. Considering the energy and operational ability of people, it is determined that the host is launching a DDoS attack, and the following restrictions are imposed on the host to protect the access of the corporate network from being affected by the external blocked IP, then remind the management personnel to block the host that launched the DDoS attack to avoid affecting other users when abnormal hosts appear in the enterprise.

References

1. M.Antonakakis,T.April,M.Bailey,M.Bernhard,andE.Bursztein, Understanding the Mirai Botnet, p. 19.
2. C. Koliass, G. Kambourakis, A. Stavrou, and J. Voas, “DDoS in the IoT: mirai and other botnets,” *Computer*, vol. 50, no. 7, pp. 80–84, 2017.
3. Dou Jiawei. Hamming distance and editing distance calculation and application for privacy protection[J].*Computer Science*,2022,49(09):355-360.)

УДК 004.4

Магістрант Кривда П. С., старш. викл. Сущук-Слюсаренко В.І.

Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»

МЕТОД ПОШУКУ ОПТИМАЛЬНОГО ВАРІАНТУ ТРИАНГУЛЯЦІЇ

Abstract

Pavlo Kryvda, Master student; senior lecturer Suschuk-Sliusarenko V.I.

Method of searching for the most profitable option for triangulation

This article presents the research devoted to solving the problem of developing the method of optimal triangulation based on existing algorithms of triangulation to optimize the process of 3D visualization, which should work faster than existing ones.

The aim of this study is to implement a triangulation algorithm in the form of software, due to which rendering would be performed 10% faster.

Вступ

Як відомо, більшу частину інформації людина отримує саме візуально. У сучасному світі спостерігається посилення візуальних комунікацій, текстуальне сприйняття сенсу поступається місцем візуально активному його розумінню: комп'ютерна графіка, гіпертексти електронних посібників, візуальні моделі електромеханічних комплексів, а також широке коло наукових досліджень, результати яких неможливі висловити у вербальній формі, що вимагає розробки нових методологій моделювання візуального сприйняття.

Візуалізація є заключним етапом роботи над сценою, що моделюється. На цьому етапі комп'ютер перетворює математичну модель сцени на форму доступну для візуального сприйняття. Цей процес називається рендерингом (render).

Основними складнощами, які виникають під час виконання процесу 3D візуалізації є велика кількість обчислень координат положення 3D об'єктів, обчислень множини пікселів на екрані, які будуть використані під час виводу, обчислення процесу тріангуляції 3D об'єкту, обчислення процесу видалення невидимих поверхонь та відносно мала швидкість виведення зображень.

Об'єкт дослідження

Об'єктом дослідження є процес 3D-візуалізації; предметом дослідження є алгоритми тріангуляції.

У дослідженні вирішується задача розроблення за допомогою модифікації та/або об'єднання існуючих алгоритмів алгоритму тріангуляції для оптимізації процесу 3D візуалізації, який повинен працювати швидше за існуючи.

Отже, метою даного дослідження є реалізація у вигляді програмного забезпечення алгоритму тріангуляції, за рахунок якого рендеринг виконувався б на 10% швидше.

Термінологія

Тріангуляція – це планарне розбиття площини на M фігур, з яких одна називається нескінченністю, а інші – трикутниками.

Рендеринг – в комп'ютерній графіці це процес отримання зображення за моделлю з допомогою комп'ютерної програми.

Стартовий вектор – вектор, з початку якого проведено вектор, який розділяє багатокутник на дві частини – трикутник та іншу частину.

Наступний вектор – вектор, в кінець якого проведено вектор, який розділяє багатокутник на дві частини – трикутник та іншу частину. Також початок цього вектора – це кінцева точка стартового вектора.

Твірний вектор – вектор, який має спільний початок зі стартовим вектором та спільний кінець з наступним вектором та який розділяє багатокутник на дві частини – трикутник та іншу частину.

Попередній вектор – вектор, який має кінцеву точку з координатами початкової точки стартового вектора.

Допоміжний вектор – це зворотній для попереднього вектор.

Опис методу

Основною ідеєю запропонованого методу є обрахування всіх можливих варіантів тріангуляції багатокутника.

Початковою дією є задання вершини багатокутника, з якої розпочнеться процес тріангуляції. В процесі виявлення всіх варіантів тріангуляції треба щоб кожна вершина багатокутника побувала початковою. Для зручності сторони багатокутника подаються у вигляді векторів, які описують багатокутник у напрямку за годинниковою стрілкою. Вводяться поняття стартового, попереднього і наступного вектора.

Для побудови першого трикутника стартова вершина з'єднується з кінцем наступного вектора для стартового вектора – це перший спосіб. Існує другий спосіб. Стартова точка з'єднується з початком попереднього

вектора для попереднього вектора для стартового вектора. Також можна скористатись третім способом, який полягає у комбінації перших двох.

Для реалізації методу розроблено програмне забезпечення. На вхід програми надходять пари числових значень, які означають координати вершин багатокутника. З цих пар числових значень утворюється об'єкт класу *Vector*, який приймає у конструктор координати вісі x та y початкової точки та координати вісі x та y кінцевої точки. Також у конструкторі є необов'язковими параметрами посилання на наступний та попередній вектор, які відображають наступну та попередню сторони багатокутника відповідно. Кожна пара значень інтерпретується як координати початку вектора, а наступна пара після неї, як координати кінцевої точки. Для останньої пари значень координатами кінцевої точки вектора буде перша пара. Таким чином створюється масив векторів. Наступним етапом методу є побудова порядку у множині векторів. Відповідна функція встановлює значення параметрам *previous* та *next* класу *Vector*, які відповідають за посилання на попередній та наступний вектор для даного об'єкту класу *Vector* відповідно, посилання відповідно на попередній та наступний об'єкти у масиві об'єктів класу *Vector*. Для останнього об'єкту у масиві параметр *next* – це посилання на перший об'єкт у масиві, а поле *previous* для першого елемента у масиві – це посилання на останній елемент у масиві. Отриманий масив векторів після цього передається в конструктор класу *Polygon*, який встановлює параметру *vectors* класу *Polygon* цей масив.

Далі отриманий об'єкт класу *Polygon* передається як параметр у основну функцію програми, яка називається *Tri*. Спочатку функція перевіряє, чи відображає об'єкт класу *Polygon*, переданий до неї як параметр, трикутник. Для цього перевіряється, чи дорівнює кількість об'єктів класу *Vector* у параметрі *vectors*, переданого об'єкту класу *Polygon*, трьом. Якщо так, то функція повертає масив з об'єктом класу *ChainPart*, перший параметр конструктора якого є масив об'єктів класу *Polygon*, які відображають трикутники утворені на даному етапі триангуляції, другим параметром є масив об'єктів класу *ChainPart*, які відображають усі можливі варіанти розбиття багатокутника, який утворився в результаті відділення від багатокутника, відображеного об'єктом класу *Polygon*, переданим до функції, тієї частини, яка належить до трикутників, утворених на даному етапі триангуляції. Третій параметр конструктора для цього класу – це рядок, який відображає, на якому етапі був створений цей об'єкт. Даний об'єкт класу *ChainPart* першим параметром отримує масив з об'єктом, переданим до функції, другим параметром – масив без елементів, третім – рядок «».

Якщо кількість об'єктів у параметрі *vectors* переданого об'єкта у функцію не дорівнює трьом, то йде перевірка на правильність даного об'єкта. Проводиться перевірка, чи довжина параметру *vectors*, переданого об'єкту

менше трьох. Якщо ні, то об'єкт правильний. Якщо так, то функція повертає масив з об'єктом класу *ChainPart*, який і першим, і другим параметром у конструктор має масиви без елементів.

Якщо відображений об'єктом багатокутник не трикутник, а має більше трьох кутів, то для кожного об'єкту класу *Vector* у параметрі *vectors* запускається операція, яка складається з трьох етапів. Під час виконання операції вектор, над яким вона виконується, має ім'я «стартовий вектор».

Перший етап алгоритму реалізується в окремій функції *firstPart*, яка виконує розбиття багатокутника на трикутники та іншу частину першим способом.

Другий етап алгоритму реалізується в окремій функції *secondPart*, яка виконує розбиття багатокутника на трикутники та іншу частину другим способом.

Третій етап алгоритму реалізується в окремій функції *thirdPart*. Цей етап комбінує дії першого та другого етапів в плані відділення одразу двох трикутників, отриманих на перших двох етапах.

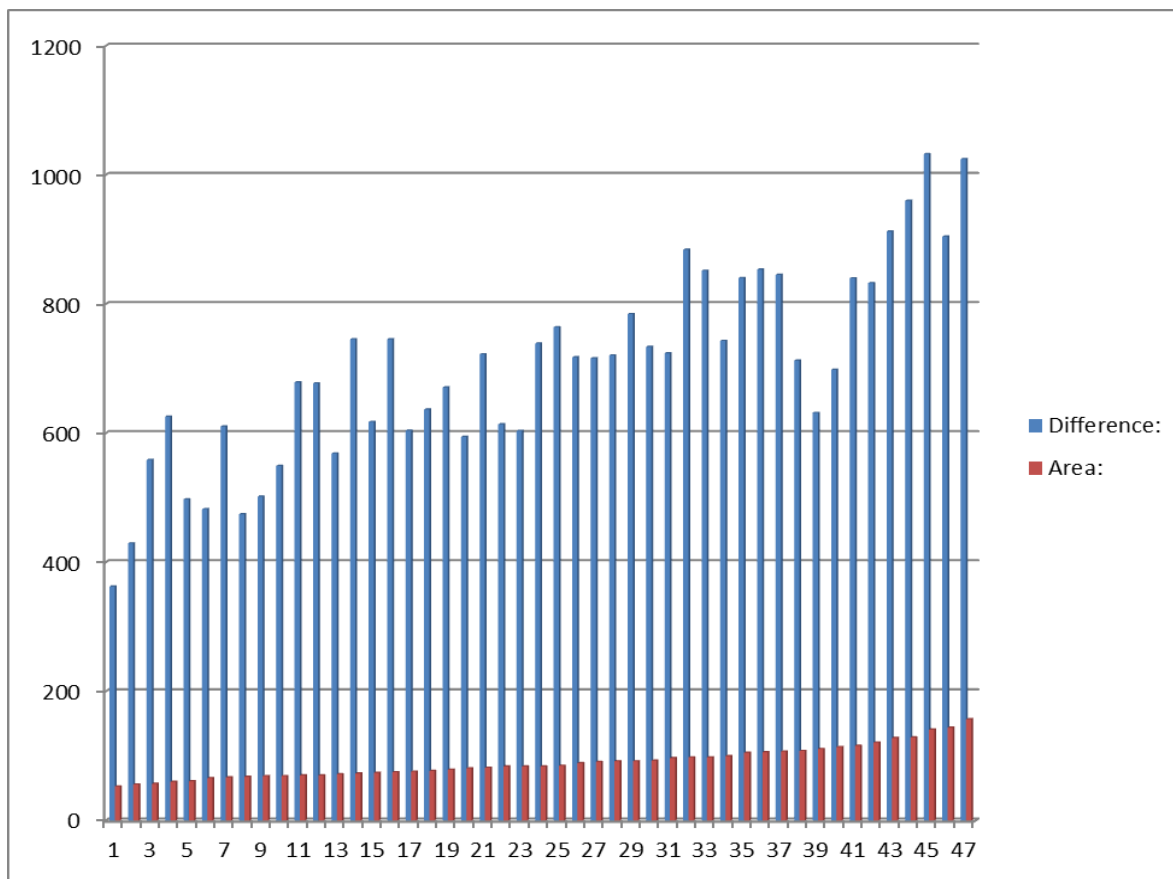
Результати

На діаграмі показано відповідність параметрів «Площа частини, яка буде перефарбована» – *Area* та «Степінь відмінності від правильного трикутника» – *Difference* для різних результатів триангуляції тестового багатокутника. З неї видно, що не завжди між цими параметрами існує пряма залежність.

В результаті проведених експериментів можна зробити висновок, що найвигідніша (найменша) площа відрізняється від найневигіднішої (найбільшої) майже втричі, а це означає, що можна значно скоротити тривалість часу, необхідного для перемалювання пікселів.

Висновки

У статті описаний метод, який дозволяє вдосконалити процес триангуляції для відображення полігону. Перевага запропонованого методу полягає в тому, що завдяки йому можна знайти найоптимальніший варіант триангуляції. Недоліком можна вважати триваліший час виконання алгоритму реалізації методу. Подальшу роботу доцільно зосередити на вдосконаленні алгоритму реалізації методу з метою скорочення часу виконання процесу триангуляції.



Література

1. Delaunay triangulation [Електронний ресурс]. – 2022 . – Режим доступу: <https://www.jstor.org/stable/43693155?seq=1>
2. Triangulation Algorithms and Data Structures [Електронний ресурс]. – 2022 . – Режим доступу: <https://www.cs.cmu.edu/~quake/tripaper/triangle2.html>
3. Triangulation Algorithm [Електронний ресурс]. – 2018 . – Режим доступу: <http://www.polygontriangulation.com/2018/07/triangulation-algorithm.html>

Д.т.н., доцент Сулема Є. С.; аспірант Песчанський В.Ю.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

АЛГОРИТМ ЗЧИТУВАННЯ ТА АНАЛІЗУ ДАНИХ МЕДИКО-БІОЛОГІЧНИХ ОБ'ЄКТІВ У ФОРМАТІ PDF НА ОСНОВІ ОПТИЧНОГО РОЗПІЗНАВАННЯ СИМВОЛІВ

*Yevgeniya Sulema, DSc, assoc. prof.; Peschanskyi Vladyslav, PhD student
Algorithm for reading and analysing data of medical and biological objects
in PDF format based on optical character recognition*

The paper proposes an algorithm for reading and analysing data of medical and biological objects in PDF format based on optical character recognition. This algorithm combines the use of a neural network to facilitate character recognition with further syntactic analysis of the received data and construction of an abstract syntactic tree based on it, which can be transformed into any common data format for use in electronic document circulation.

Вступ

У сучасному світі все більше галузей починають використовувати цифрові технології для полегшення та оптимізації своєї діяльності. Безумовно, навіть на перший погляд проста організація паперових даних у електронні сховища може значно полегшити документообіг та звітність.

Медична галузь також здійснює перевід своєї інфраструктури у цифровий простір. За останні роки було здійснено величезну кількість роботи по переведенню усіх документів та обладнання у єдину цифрову систему, що надає можливість значно підвищити ефективність обміну інформацією у сфері, яка має прямий вплив на якість життя людей. Проте, значна кількість даних все ще зберігається на фізичних носіях, що значно ускладнює процес цифровізації медичної галузі, адже більшість сучасних інформаційних систем вимагають представлення даних в специфічному форматі, а перетворення вже існуючої інформації нерідко являє собою досить довгий та складний процес. Отже, доцільним є розроблення системи полегшення процесу цифровізації даних з метою їх подальшого використання у системах електронного документообігу.

Мета дослідження

Головною метою дослідження є оптимізація процесу аналізу та уніфікації даних медико-біологічних об'єктів, збережених у форматі PDF за допомогою використання оптичного розпізнавання символів.

Практичною задачею дослідження є розробка програмного забезпечення для перетворення даних медико-біологічних об'єктів, збережених у форматі PDF у більш розповсюджений формат JSON.

Опис алгоритму

Згідно з документацією стандарту ISO 32000 [1] документи PDF в більшості випадків являють собою комбінацію векторної графіки, тексту та растрової графіки. Основні типи вмісту в PDF:

- текст, що зберігається як суцільний потік даних (тобто не закодований у вигляді звичайного тексту, наприклад вручну заповнені таблиці);
- векторна графіка для ілюстрацій і схем, які складаються з форм і ліній;
- растрова графіка для інших типів зображень.

У випадку медико-біологічних об'єктів найбільш розповсюдженим варіантом зберігання даних є перший пункт наведеного списку, так як більшість інформації являє собою заповненні рукописними або друкованими літерами таблиці. Отже, на основі вищезазначеної тези можемо сформулювати основні кроки алгоритму цифровізації даних медико-біологічних об'єктів, які будуть наведені на рис.1.

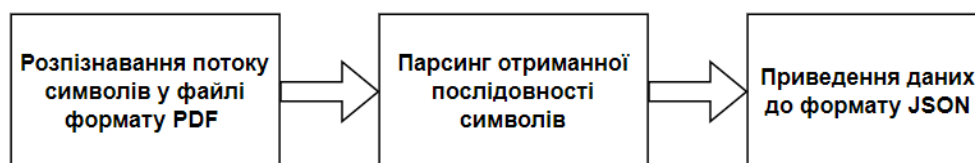


Рис. 1. Запропонований алгоритм цифровізації даних

Основним та найбільш складним з усіх етапів, безумовно, є розпізнавання потоку символів у вхідному файлі формату PDF.

Для вирішення поставленої задачі пропонується використати нейронні мережі. Адже здатність розпізнавати символи через абстракцію чудово підходить для більш-менш фіксованих документів, які доволі часто містять у собі нерозбірливий рукописний текст. Базовий алгоритм тренування нейронної мережі приведено на рис.2.

На тренована нейронна мережа на основі бібліотеки TensorFlow [2] з відкритим кодом, згідно з проведеними дослідженнями, результати яких представлені в табл. 1, показує дуже високу вірогідність розпізнавання друкованих та рукописних символів, що підтверджує доцільність її використання на цьому етапі.

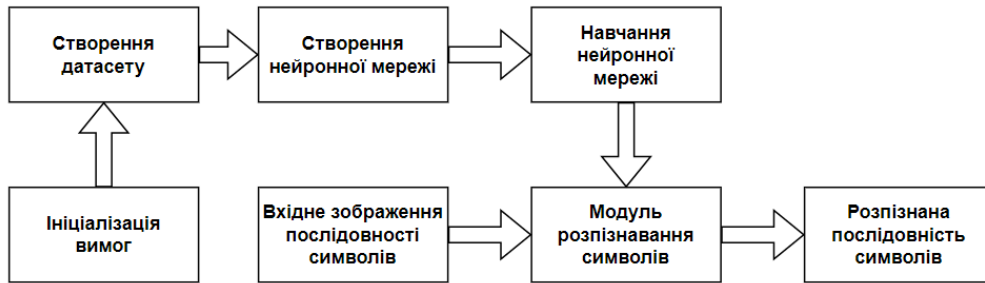


Рис. 2. Алгоритм створення нейронної мережі

Таблиця 1

Результати тесту нейронної мережі

Надруковані символи		
Розмір набору даних	Розмір вибірки	Результат розпізнання
11173 символів	2793 символів	98.44%
Рукописні символи		
Розмір набору даних	Розмір вибірки	Результат розпізнання
5407 символів	1351 символів	94.62 %

Наступним кроком алгоритму є перетворення послідовності символів, отриманої в якості результату роботи нейронної мережі у послідовність лексем, на основі яких буде можливо виконати перетворення у електронний формат.

Загальну схему роботи алгоритму аналізу вхідної послідовності символів можна побачити на рис.3.

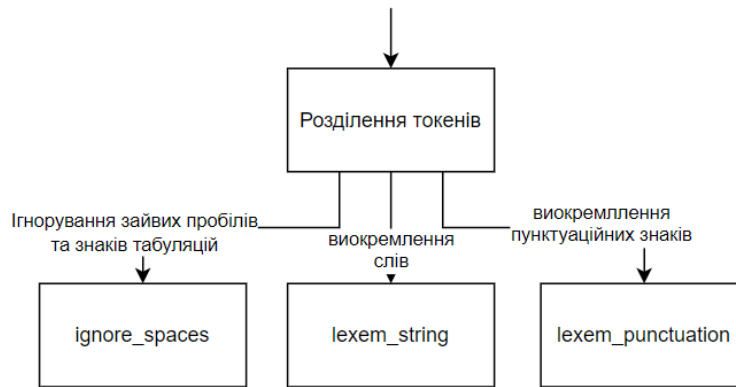


Рис. 3. Алгоритм лексичного аналізу

Оскільки, за припущенням, більша кількість вхідних даних являє собою варіації таблиць, результуючий алгоритм не потребує аналізу складних лексичних конструкцій та патернів зберігання інформації. А отже можна орієнтуватися на стандартне представлення даних у таблиці, де верхній рядок є заголовком, а усі інші – змістовними полями.

Таким чином, можна перетворювати дані, збережені у форматі PDF у формат JSON, який є одним із найбільш розповсюджених [3] електронних форматів передачі даних.

Header 1	Header 2	Header 3
Some data 1.1	Another data 1.2	And some data 1.3
Some other data 2.1	Some another data 2.2	Some Another data 2.3



```

[
  {
    "Header 1": "Some data 1.1",
    "Header 2": "Another data 1.2",
    "Header 3": "And some data 1.3"
  },
  {
    "Header 1": "Some other data 2.1",
    "Header 2": "Some another data 2.2",
    "Header 3": "Some Another data 2.3"
  }
]
  
```

Рис. 4. Результат роботи алгоритму

Висновки

Алгоритм, представлений у статті, розроблено для зчитування та аналізу даних медико-біологічних об'єктів у форматі PDF на основі

оптичного розпізнавання символів. Цей алгоритм комбінує використання нейронної мережі для полегшення розпізнавання символів з подальшим синтаксичним аналізом отриманих даних та перетворення їх у розповсюджений формат даних JSON, який значно полегшить їхнє подальше використання у електронному документообігу.

Література

1. Sugano, H., Fujimoto, S., Klyne, G., Bateman, A., Carr, W. and Peterson, J., 2004. Presence information data format. No. rfc3863.
2. Dillon, J.V., Langmore, I., Tran, D., Brevdo, E., Vasudevan, S., Moore, D., Patton, B., Alemi, A., Hoffman, M. and Saurous, R.A., 2017. Tensorflow distributions. arXiv preprint arXiv:1711.10604.
3. Pezoa, F., Reutter, J.L., Suarez, F., Ugarte, M. and Vrgoč, D., 2016, April. Foundations of JSON schema. In Proceedings of the 25th International Conference on World Wide Web, pp. 263–273.

UDC 004.627

Tang Jiaoyan, master student; senior lecturer Suschuk-Sliusarenko V.I.

**National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”**

CHINESE TEXT REPRESENTATION AS BGW 2D BARCODE

Abstract

Магістрант Тан Цзяоянь, старш. викл. Сущук-Слюсаренко В.І.

Алгоритм представлення китайського тексту як 2D штрих-коду BGW

У цій статті розглядається спосіб представлення машинописних текстів китайською мовою у вигляді BGW 2D Barcode – чорно-сіро-білих матричних штрихових кодів. Особливістю застосування цього методу є можливість ущільненого представлення тестових даних, поданих китайською мовою.

Introduction

With the continuous growth of the national economy and the continuous innovation of science and technology, barcode technology is widely used, creating a lot of practical value. As the most important part of the construction and management process of a modern digital library, the scientific application of barcode technology is an indispensable technology, which is directly related to whether it can provide convenient information services for library users, give full play to the role of resources, and promote the stable and sustainable development of library construction. Therefore, libraries should attach great importance to the innovation of barcode technology, and effectively integrate this technology into different links of the work content, to bring more convenience for managers and users. This paper will analyze and discuss the application of barcode technology in the library.

Research Objective

The main objective of the research is to increase the library information capacity. In the process of library management, the number of books is large, and the library's information storage is also relatively large. The method allows for more information to be encoded while still preserving the size of the graphical symbol.

Algorithm description

Adding more colors to a barcode and making it multicolor instead of black and white will increase the amount of information that can be encoded while preserving the size of the graphical symbol.

Using an RGB palette runs the risk of distorting colors or rendering incorrectly when printing. Black-Gray-White barcode, or grayscale tricolor 2D barcode [1] is an effective solution for representing multicolor barcodes in printed form.

A data barcoding method that encodes more information while preserving the area of the graphical symbol based on the Black-Gray-White barcode includes the following steps.

1. Theoretical Background.

Let an entering alphanumeric sequence is as follows [1]:

$$T = t_1 t_2 \dots t_n \quad (1)$$

where $t_i \in GB2312 - 80$ is an element of the entering sequence, $i = 1, 2 \dots n$, n is the T sequence length.

GB2312-80 is the national standard for Chinese character coding. The set of symbols can be presented as follows:

$$GB2312 - 80 = \{L \cup D \cup C\} \quad (2)$$

where L is a letters set, D is a digits set, and C is a special symbols set.

The sequence T can be divided into adjacent subsets that are consisted of elements belonging to one of the GB2312-80 subsets:

$$\begin{cases} W_1 = t_1 t_2 \dots t_{i-1} \in L \\ W_2 = t_i t_{i+1} \dots t_{n-i+1} \in D \\ \dots \\ W_k = t_{n-i} t_{n-i-1} \dots t_n \in C \end{cases} \quad (3)$$

Entering sequence acquires:

$$T = W_1 W_2 \dots W_k \quad (4)$$

where W_i is a subsequence of the entering sequence which contains elements of only one set of the GB2312-80 subsets. In the entering sequence T , the subsequences $W_1 W_2 \dots W_k$ can be situated in any order.

2. BGW 2D Barcode Construction.

As stated in [1], the procedure of constructing BGW 2D Barcode is the following.

I. The transformation of symbols into codes corresponds to the order of the symbols in the sequence. The symbols in the alpha-numeric sequence are being replaced by codes that correspond to the order number of the symbol in the alphabet. As a result, we have a sequence of order numbers.

II. This is the first stage of transforming the decimal number P_A to the new notation system $P_{\Omega_{inf}}$.

III. Transformation of the number in notation $P_{\Omega_{inf}}$ to a ternary number.

IV. The number in the ternary system is represented with a matrix that has each cell being one of three colors.

3. Practical Implementation.

The sequence of ternary numbers is being represented in a matrix of 7×10 , where white represents 0, gray represents 1, and black represents 2.

Conclusions

By using the BGW barcoding method [1], the amount of space needed to store textual data should be reduced. The method not only encodes data but also compresses it, which is a benefit. Depending on various parameters, compression can be significantly increased, up to 25%. This is especially true when considering the digital capacity of barcode symbols.

References

1. Onai M V, Sulema O K, Dychka A I. Data encoding based on tricolor matrix barcodes [J]. 2019.
2. Singh G, Sharma M. Barcode technology and its application in libraries and Information centers [J]. International Journal of Next Generation Library and Technologies, 2015, 1(1): 1-8.

УДК 004.912

Магістрант Фалілеєва Д.М., к.т.н. Хіцко Я.В.

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

СТВОРЕННЯ ПЕРСОНАЛЬНОГО ВЕБ-САЙТУ З ВИКОРИСТАННЯМ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ

Abstract

Daria Falileieva, student; Yana Khitsko, PhD

Creating a personal website using Machine learning algorithms

In this work the hybrid method of recommendation system is analyzed. Proposed method based on using k-nearest neighbors approach. The main steps of the method with its features and advantages are outlined in this paper.

Вступ

В епоху інформаційного перевантаження користувачам дуже важко отримати інформацію, що їх дійсно цікавить. Іноді ми отримуємо стрічку відео, новин, одягу тощо, що не відповідає нашим інтересам. Це робить клієнтський досвід жахливим, і в якийсь момент він просто покине застосунок чи сайт, адже не знайде цінності для себе. Також для контент-провайдера дуже важко зробити так, щоб їх продукт вирізнявся з натовпу. Ось чому багато компаній розробляють рекомендаційну систему. Така система передбачає інтереси користувачів і дає рекомендації відповідно до моделі інтересів користувачів. Головна задача такої системи полягає в тому, щоб з'єднати користувачів та інформацію, яка, з одного боку, допомагає користувачам знайти цінний для них контент і не витратити час, а з іншого – просувати інформацію конкретним користувачам, підвищувати популярність контенту і досягати гарних рейтингів. Це безпрограшна ситуація для обох сторін.

Загалом, усі чули про системи рекомендацій і мають дуже приблизне уявлення про їх базове функціонування, навіть із простого особистого

досвіду. Що, однак, не так добре відомо, так це роль машинного навчання в їхніх базових механізмах.

Машинне навчання (ML) — це підгалузь штучного інтелекту, яка зосереджується на створенні комп'ютерних алгоритмів, які можуть обробляти величезні набори даних, ідентифікувати повторювані закономірності та кореляції між кількома змінними та створювати математичні моделі, що їх зображають [1].

Це «навчання» в його назві не випадково, оскільки системи машинного навчання, а також додатки навчання з підкріпленням, що використовують ці алгоритми, можуть фактично розширити свої можливості з часом через досвід. Чим більше даних вони обробляють, тим більше зв'язків між точками даних вони помічають і тим краще вони налаштовуюватимуть свої моделі.

Такі моделі, які також зазвичай використовуються для прогнозування аналітики в маркетингу, прогнозують те, що думають клієнти, оскільки вони дозволяють нам сформулювати логіку конкретних моделей купівлі та пролити світло на поточні тенденції продажів або навіть передбачити майбутні події.

На віртуальних ринках системи рекомендацій відіграють подібну роль, замінюючи продавців-помічників у сегментуванні та рекомендаціях продуктів клієнтам. Основна відмінність полягає в тому, що люди-продавці керуються своєю інтуїцією та досвідом, щоб дослідити невелику частку вищезгаданих змінних під час короткої розмови з покупцями.

Натомість механізми рекомендацій покладаються на машинне навчання для обробки величезних наборів даних клієнтів і розглядають ширший діапазон параметрів для виконання цього процесу класифікації та націлювання. До них належать поведінка користувача, історія покупок, використання вмісту, особиста інформація з профілів користувачів, огляди продуктів і пристрої доступу. Тож система рекомендацій може отримати досить чітке уявлення як про окремих покупців, так і про аудиторію в цілому, а також з'ясувати базову динаміку продажів, яку людині-спостерігачу було б важко зрозуміти.

Постановка задачі

Створюючи систему рекомендацій з нуля ми стикаємося з кількома різними проблемами.

Зараз існує багато систем рекомендацій, заснованих на інформації користувача, тож виникає проблема, коли веб-сайт не отримав достатньо користувачів. Ще одне питання - формалізація якісних зв'язків між

користувачем і елементом на основі розріджених даних. Після цього потрібно вирішити головні параметри фільму, тобто те, як система може зрозуміти, як ідентифікувати цей фільм. Це передумова для порівняння подібності між двома фільмами. Фільм за такими ознаками, як жанр, актор і режисер, можна класифікувати, але для кожної функції фільму має бути різна вага і для кожного з них вони відіграють іншу роль для рекомендації. Наступна проблема - масштабованість, означає, що передбачити складно у величезній кількості елементів рейтингу.

Отже, ми отримуємо такі задачі, які нам треба вирішити:

- Як рекомендувати фільми, коли немає інформації про користувача.
- Які функції фільму можна використовувати для системи рекомендацій.
- Як обчислити схожість між двома фільмами.
- Як об'єднати результати кластеризації продукту та матрицю оцінок користувачів.
- Чи можна встановити вагу для кожної функції.
- Яку обрати систему рекомендаційної оцінки, як вибрати достатній рівень схожості.
- Як обчислювати точність рекомендації.

Гібридний метод рекомендацій

Розроблений гібридний рекомендаційний алгоритм базується на підході “к-найближчих сусідів” [2]. Модуль розрахований на роботу в два етапи. На першому етапі він обирає тих сусідів, яких вважає найбільш придатними. На другому етапі модуль створює список потенційних елементів вмісту, які можуть бути запропоновані системою. Потім цей список передається до гібридного модуля для подальшої обробки.

Модуль для вибору найближчого сусіда базується на коефіцієнті Пірсона. Було вирішено використати цю метрику відстані, оскільки вона досить часто використовується в підходах до моделювання користувачів, а також доведено, що вона дає хороші результати. Система розроблена для пошуку 10 найближчих сусідів, як видно з рис. 1 евклідова відстань подібності в цьому випадку має найменше RMSE, тож це число є найкращим компромісом між швидкістю та ефективністю.

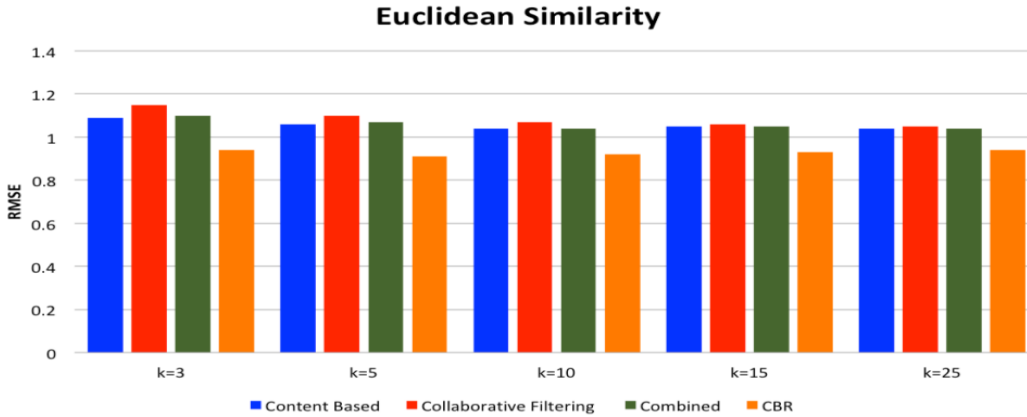


Рис. 1 RMSE евклідової відстані подібності в різних варіаціях параметра k в методі k-найближчих сусідів

Як подальше вдосконалення системи, було налаштовано показник відстані, щоб надавати перевагу користувачам, які зареєстрували більшу кількість оцінок у системі. Таким чином, можна переконатися, що система має достатньо великий пул можливих рекомендацій. Щоб знайти цих сусідів, система порівнює профіль (рейтинги) активного користувача з профілями всіх інших користувачів у базі даних, за умови, що у них зареєстровано достатній рейтинг. Мінімальна кількість оцінок для кожного користувача, необхідних для цього модуля, була встановлена рівною 20.

Після того, як система обере сусідів користувача, вона переходить до створення списку рекомендованих елементів. Для цього він спочатку складає список усіх предметів, які оцінили сусіди. Потім він видаляє зі списку всі елементи, які вже були оцінені активним користувачем. Усі решта позицій стають потенційно рекомендованими. Щоб звужити цей вибір, система обчислює прогнозований рейтинг для кожного з цих елементів. Для розрахунку цього рейтингу наша система використовує істинний баєсівський алгоритм прогнозування оцінки.

Після обчислення прогнозованих рейтингів вони сортуються від найкращого (найвищий рейтинг) до найгіршого (найнижчий). Оскільки нашою метою було розробити гібридну систему рекомендацій, ми жодним чином не фільтрували цей список, а натомість переслали його до модуля hybrid recommender [3].

Таким чином, була проведена загальна оцінка системи. Зазвичай така система рекомендує лише обмежену кількість найкращих рекомендацій. Загальна продуктивність прогнозування нашого спільного модуля є досить хорошою, але однією з найбільших проблем, з якою ми зіткнулися під час тестування, є його вимоги до обробки. У той час, як процес розрахунку

прогнозованих рейтингів вимагає відносно небагато часу, процес вибору найближчих сусідів вимагає багато часу та ресурсів. Тому була створена власна система для пошуку сусідів лише тоді, коли зареєстровано нового користувача або коли користувач зареєстрував достатньо нових оцінок, щоб було сенс повторити процедуру.

Рекомендувач на основі вмісту генерує прогнозовані рейтинги для фільмів, описаних за допомогою таких атрибутів метаданих: жанр(и), актори, режисер(и), сценарист(и) і нагорода(и). Кожен із атрибутів обробляється окремо, і придатність окремого фільму спочатку обчислюється окремо для кожного з цих атрибутів. Остаточне прогнозування рейтингу базується на придатності цих атрибутів і розраховується за допомогою алгоритму правил прийняття рішень M5Rules [4].

Слід зазначити, що доступні описи контенту також містили короткий синопсис, описуючи контент неструктурованим текстом. Проте, було вирішено не використовувати цей атрибут для персоналізації через незадовільні результати в наших попередніх експериментах. Ці експерименти показали, що ключові слова, отримані з конспекту, дають корисну інформацію про вподобання користувача лише у випадку спортивних програм, де ключові слова мають високу інформаційну цінність (спортивний жанр, назви команд, імена учасників тощо).

Друга фаза використовується для навчання алгоритму M5Rules, який повинен мати можливість передбачити рейтинг певного фільму на основі попередньо обчислених значень придатності атрибутів. Після того, як рекомендувач на основі вмісту буде оновлено/навчено, він готовий генерувати прогнози рейтингу для фільмів, на основі їхніх описів метаданих. [5]

Подібно до етапу навчання, прогнозування рейтингу також є двоетапним процесом. На першому етапі придатність атрибутів обчислюється на основі опису метаданих фільму та даних, що зберігаються в моделі користувача. Результатом цього етапу є 5 значень придатності атрибутів, у діапазоні від 0 до 1, які представляють придатність фільму для користувача, беручи до уваги список акторів, режисера(ів), жанр(ів), автора(ів) сценарію та нагороди.

Потім ці значення передаються в алгоритм M5Rules, який розраховує прогнозований рейтинг. Запропонований підхід було обрано серед низки доступних підходів (регресійна регресія, баєсівська регресія тощо) через його пряmlinійність (завжди можна отримати списки найпопулярніших акторів, жанрів, тощо), тому що вони прийнятні з точки зору обчислень і тому, що попередні випробування дали хороші результати.

Щойно прогнозовані рейтинги, згенеровані рекомендувачем на основі вмісту та спільним рекомендувачем, стають доступними, запускається гібридний рекомендувач.

Гібридний рекомендувач приймає передбачувані рейтинги як вхідні дані та на основі їх значень класифікує фільм до класу рекомендованих або нерекондованих елементів. Для поточної системи ми запровадили систему зваженого голосування. Спочатку встановлюється порогове значення, яке визначає межу між класами рекомендованих і нерекондованих елементів.

Процедура гібридного голосування виглядає наступним чином: якщо обидва рекомендувачі «погоджуються» щодо класифікації фільму, фільм безпосередньо класифікується до прогнозованої категорії. Якщо рекомендувачі не погоджуються, застосовується схема зваженого голосування. Прогнозоване голосування кожного з рекомендованих зважується з використанням ваги того, хто рекомендує, і розраховується середній рейтинг. Якщо це значення перевищує порогове значення, цей фільм класифікується як рекомендований, і навпаки. Вагові коефіцієнти, які використовуються в нашій поточній системі, були встановлені однаково для обох модулів рекомендацій.

Результати гібридного методу рекомендації

Порівняння помилок RMSE і MAE в методах колаборативної фільтрації і описаного вище гібридного методу на прикладі 10 довільних користувачах зображено на Рис. 2 і Рис. 3

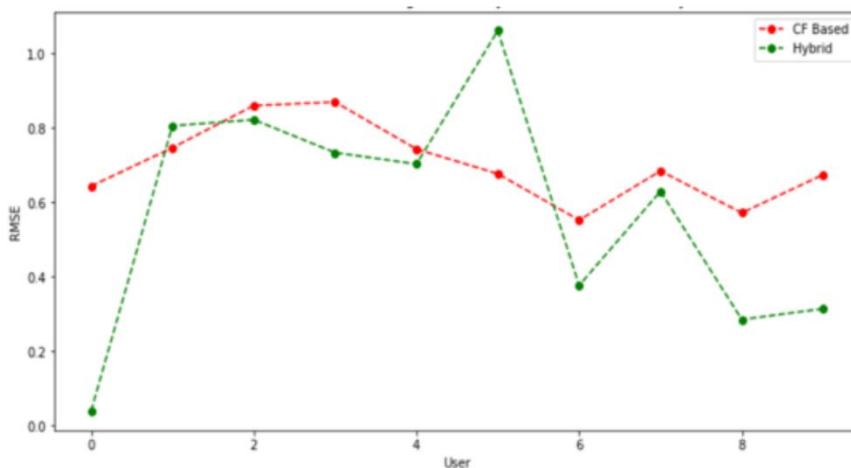


Рис. 2 RMSE методу колаборативної фільтрації і гібридного методу

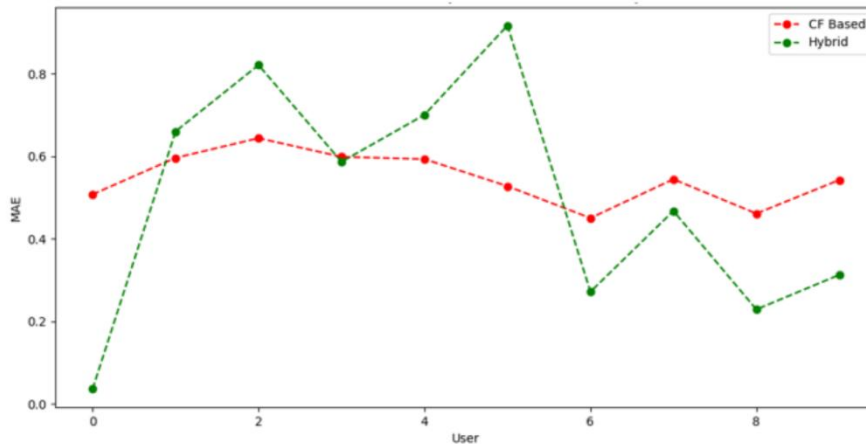


Рис. 3 MAE методу колаборативної фільтрації і гібридного методу

Висновки

Описане у даній статті дослідження присвячене розробці гібридного методу рекомендацій. Незважаючи на просту гібридну конструкцію, рекомендація дає значення високої точності. Крім того, підхід до вибору найближчих сусідів для модуля спільної роботи залишає простір для вдосконалення, як і методи розрахунку придатності атрибутів у рекомендувачі на основі вмісту. Цей документ не стосується питань безпеки та конфіденційності. Однак ми знаємо про них і вважаємо, що їх можна вирішити за допомогою стандартних підходів, які мають запобігти неправомірному використанню та компрометації даних користувача.

Література

1. Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. Knowledge and Data Engineering, IEEE Transactions on, 17(6):734–749, 2005.
2. Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. Modern information retrieval, volume 463. ACM press New York, 1999.
3. Shumeet Baluja, Rohan Seth, D Sivakumar, Yushi Jing, Jay Yagnik, Shankar Kumar, Deepak Ravichandran, and Mohamed Aly. Video suggestion and discovery for youtube: taking random walks through the view graph. In Proceedings of the 17th international conference on World Wide Web, pages 895–904.

- ACM, 2008.
4. Robert Bell, Yehuda Koren, and Chris Volinsky. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 95–104. ACM, 2007.
 5. Suvir Bhargav. Efficient features for movie recommendation systems. 2014.

УДК 004.05

**Магістрант Панков Т.С., к.т.н., доцент Потапова К.Р.,
асистент Кучмій О.О.**

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

СЕСІЙНИЙ АЛГОРИТМ СТИСКАННЯ ЗОБРАЖЕНЬ

Abstract

**Tymur Pankov, student; Potapova Kateryna, assoc. prof., PhD; Kuchmii Oksana, asisstant
*Sessional image compression algorithm***

The basics of image compression and mathematical models of data compression are considered. Based on the fact that learning occurs in context compression, an efficient website image compression algorithm is implemented.

Вступ

Більшість даних, що передаються по мережі - це фото та відео файли.

Для швидкої передачі цих файлів та компактного їх зберігання використовуються алгоритми стискання. Алгоритми мають бути ефективними, здатними швидко та якісно стискати окремі зображення та послідовності кадрів відео.

Головною проблемою при стисканні послідовності кадрів відео є обчислювальна складність пошуку областей кореляції зображень, яка проводиться з метою підвищення коефіцієнта стискання шляхом кодування посилення на схожу область зображення та різниці між попереднім та наступним зображенням, замість кодування повністю вихідного зображення. Проблемою при стисканні набору зображень є виділення загального контексту даних зображень з метою підвищення якості стискання.

Постановка задачі

Задача полягає у вивченні математичної бази алгоритмів стискання даних та зображень та побудові алгоритму стискання зображень, що спирається на контекст стискання та потенційно перевершує за ефективністю існуючі алгоритми стискання.

Термінологія

lossless (стискання без втрат) – стискання, при якому вихідні дані можуть бути точно відновлені після стискання;

Ентропія - міра невизначеності (невпорядкованості) системи.

Кодування серій - стиснення послідовностей, що поспіль йдуть символів.

Квантування (у термінах стиснення) — зниження ентропії інформації з допомогою виділення рівнів і округлення значень до цих рівнів. Застосовується переважно в lossy стисканнях, оскільки майже завжди призводить до втрати інформації.

Оптичний потік - технологія, що використовується в різних галузях комп'ютерного зору визначення зрушень, сегментації, виділення об'єктів, компресії відео.

Дельта кодування - перетворення, що замінює кожне наступне число різницею між ним та попереднім числом. Використовується з метою зниження ентропії вхідних даних у випадках, коли вхідні дані є зростаючою або спадною послідовністю.

Опис алгоритму

Розглянемо алгоритм стискання без втрат з погляду теорії множин. Стискання без втрат передбачає бієктивне (взаємно однозначне) відображення вихідних даних у стиснуті дані. Розглянемо як безліч вихідних даних - безліч всіх можливих зображень розміру $N \times M$ з кількістю ступенів яскравості D . Нехай безліч вихідних даних - X , безліч стиснутих даних - Y . Зважаючи на бієкцію, $|X| = |Y|$.

Припустимо тепер, що у вхід для стискання надходить випадкове зображення з вибраними параметрами, причому кожне зображення надходить рівноймовірно. Головна відмінність у міркуваннях нижче основних принципів стискання, описаних у першому розділі, у тому, що розглядатимемо все зображення як єдиний випадковий об'єкт. Зауважимо, що тоді за формулою Шеннона за умови, що зображення надходить з рівномірного розподілу, кількість інформації, яка потрібна для опису цього зображення дорівнює:

$$H(x) = - \sum_{|Y|} \frac{1}{|Y|} \log_2 \frac{1}{|Y|} = \log_2 |Y| = \log_2 |X| = NM \log_2 D \quad (1)$$

Виходячи з формули (1) можна помітити, що ця величина — є нічим іншим, як розмір зображення у форматі BMP, без стискання. Таким чином, незалежно від алгоритму стискання, якщо зображення, яке подається на вхід,

цілком випадкове, в середньому воно займатиме не менше місця, ніж у стиснутому вигляді. Звідси випливають такі наслідки.

1. Стискання даних не існує без припущення про нерівномірність розподілу даних, що стискаються.
2. Для будь-якого алгоритму стискання істинно твердження: для будь-якої безлічі вхідних даних X , сума біт, витрачених на збереження всіх цих даних у стиснутому вигляді не менше суми біт, витрачених на збереження цих даних у стиснутому, "сирому" вигляді, що дорівнює $|X| \log_2 |X|$.
3. У свою чергу з пункту вище випливає, що для будь-якого алгоритму стискання деякі дані після "стискання" зменшуються у розмірі, а деякі збільшуються, причому чим більше даних вимагають менше біт для зберігання, тим сильніше збільшується кількість біт, що витрачаються на зберігання інших даних у стиснутому вигляді.
4. Також з другого пункту випливає, що кожному алгоритму можна для кожної множини даних присвоїти коефіцієнт надмірності алгоритму стискання A на множині X , який дорівнює:
$$\frac{\sum_{x \in X} A(x) - |X| \log_2 |X|}{|X|}$$
, де $A(x)$ — кількість біт, необхідних для зберігання вхідних даних x , стиснутих алгоритмом A . Зауважимо, що це коефіцієнт завжди невід'ємний і що, чим він більше, тим гірше поводить ся аналізований алгоритм на рівномірно розподілених даних.
5. Немає поняття "ефективність алгоритму стискання" або "якість алгоритму стискання" без вказівки множини X , а також інформації про розподіл ймовірностей елементів цієї множини.

Припустимо тепер, що вхідні дані підпорядковуються деякому відомому нам розподілу $p(x)$. Тоді за формулою Шеннона маємо середню кількість біт для збереження стиснутих даних:

$$H(X) = - \sum_{x \in X} p(x) \log_2 p(x) \quad (2)$$

Зауважимо, що в даному випадку від функції ймовірності безпосередньо залежить якість стискання. До речі, якщо розробити такий алгоритм, який з одиничною ймовірністю вгадуватиме такі дані, кількість бітів, необхідних для передачі цієї інформації, буде дорівнювати 0.

Отже, потрібно знайти щільність ймовірності $P(x)$ щодо доступної інформації про природу розподілу вхідних даних. У найпростішому випадку доступна інформація — це попередні стиснуті зображення в даному

контексті. Іншими словами, потрібно знайти умовну ймовірність $P(x|X)$, де X - безліч попередніх зображень, що стискаються.

Як вже було зауважено, стискання можливе тільки в припущенні про те, що вхідні дані мають нерівномірний розподіл. Тож припустимо, для простоти, що маємо справу з одновимірними даними (двовимірні дані нескладно звести до одновимірних) і перетворювач - лінійна функція. В якості квантувальника розглянемо округлення до найближчого цілого. Тоді стискання можна розглядати як функцію виду:

$$Ax = b, c = \text{Coder}([b]) \quad (3)$$

В рівнянні (3) x - вхідні дані, A - матриця перетворення, b - вектор значень, отриманий після перетворення, $[b]$ - округлення кожного із значень вектора до найближчого цілого, Coder - кодувальник символів.

Властивість, яку повинен мати алгоритм стискання — можливість декодування. А отже, кожне з перетворень має бути оборотним.

Властивість, яку повинен мати перетворювач - стійкість до невеликих змін b . Тобто при невеликих змінах змінної b , змінна x , що знаходиться в рівнянні (3), також має змінюватися не сильно. Ця характеристика описується числом обумовленості матриці, яке обчислюється за такою формулою:

$$\mu(A) = \|A\| * \|A^{-1}\| \quad (4)$$

Перетворення A необхідне для забезпечення обнулення більшості останніх елементів вектора, інакше кажучи — необхідне для “ущільнення енергії”.

В якості кодувальника символів може розглядатися будь-який кодувальник, що спирається на відомі ймовірності символів. У найпростішому випадку можна розглянути кодувальник, який перед записом числа, що кодується записує, скільки біт потрібно для збереження даного числа без останніх нулів. Таким чином, якщо маємо числа від 0 до n , кількість біт, у найгіршому випадку необхідних для збереження одного з цих чисел:

$$\lceil \log_2(\log_2(n)) + \log_2(n) \rceil \quad (5)$$

Розглянемо процес декодування загалом:

$$b + e = \text{Decoder}(c), x_{dec} = A^{-1}(b + e) \quad (6)$$

Отже, отримали загальну схему стискання зображень. По-перше, неважко помітити, що при правильному доборі коефіцієнтів A і при правильному виборі кодувальника символів, дану схему можна порівняти зі стисканням JPEG і стисканням JPEG2000.

Для JPEG достатньо замінити матрицю A на коефіцієнти дискретного косинусного перетворення:

$$A = [\cos(k * (l + \frac{1}{2}) * \frac{\pi}{n})]_{0 \leq k, l \leq n} \quad (7)$$

В якості квантувача використовувати не округлення, а ділення націло на відповідні коефіцієнти з матриці квантування.

А в якості кодувальника символів достатньо використовувати сукупність RLE і кодів Хаффмана.

У свою чергу, для JPEG2000 достатньо замінити матрицю A на набір коефіцієнтів дискретного вейвлет-перетворення DWT-2D.

Опишемо алгоритм стискання зображення виходячи з отриманих збігів. Зауважимо, що алгоритм вибору ключових точок позитивно впливає на ефективність одержуваного алгоритму, оскільки ключові точки обираються в областях з високою ентропією, а саме ці області найскладніше стискаються будь-яким алгоритмом стискання, у тому числі й JPEG. У кращому випадку доведеться стискати лише одноколірні області та різниці між кешованими частинами та вихідним зображенням.

Отже, обираємо кращі збіги дескрипторів, відповідні ключові точки і замінюємо області навколо цих ключових точок на зображенні, що розглядається, на різницю між знайденою областю і аналізованої. Ділянки, що використовуються, і різниця, відповідно (розмір блоку навколо ключової точки — 64x64 пікселів):

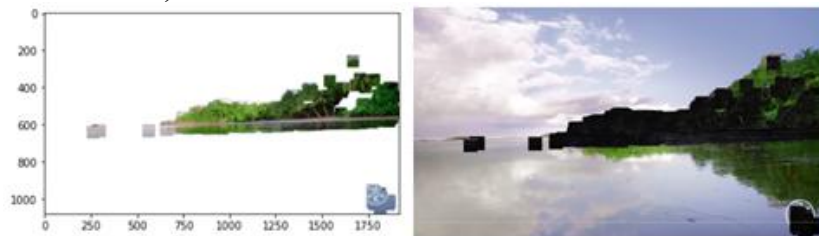


Рис. 1. Виділення та видалення областей схожих із областями з попереднього ключового кадру

Спробуємо стиснути тільки отримане зображення. Вихідне зображення у форматі JPEG важить 159.1 КВ, отримане - 147.9 КВ. Краще, але трохи.

Зауважимо, що JPEG розбиває зображення на фрагменти по 8x8, тому спробуємо вирівняти частини, що отримуються, по кратній сітці 8x8. Відобразимо отримане зображення:

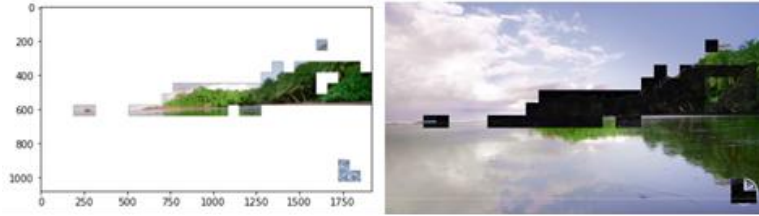


Рис. 2. Виділення та видалення областей схожих із областями з попереднього ключового кадру з вирівнюванням по сітці 8x8

Отриманий розмір - 141.1 KB.

Також зауважимо, що всі блоки 8x8 в кінці обробляються разом, тому об'єднаємо різниці в одне зображення, а все інше - окремо, тим самим розташувавши поруч схожі блоки 8x8. Пропорції зображення зберегти не вдасться, тому зберігати все будемо у зображення у вигляді рядка шириною рівною ширині блоку, що розглядається навколо ключової точки. У даному випадку – 64.

Отримані розміри — для різниць: 37KB, для частин зображення, що залишилися: 105KB. Сумарно вийшло більше, ніж було, тому що JPEG для кожного зображення зберігає додаткову інформацію, таку як таблиці Хаффмана, наприклад. Однак тепер ми маємо можливість варіювати якість окремо зображення та окремо — різниць. Так як різниці незначні, їх можна стискати без втрати якості. Збережемо зображення з різницею із параметром quality=10. Отримаємо розмір різниць 7.5KB. Разом сумарний розмір $7.5KB+105KB=112.5KB$ проти вихідних 159.1KB. Подивимося на вихідне та декодоване після нашого стиснення зображення відповідно:

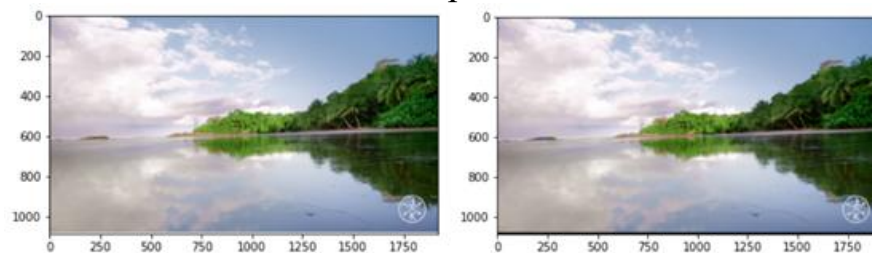


Рис. 3. Зліва – вихідне зображення, праворуч – збережене у новому форматі

Висновки

Так як існуючі алгоритми мають недоліки, було прийнято рішення про створення нового алгоритму, який зможе врахувати існуючі недоліки та усунути їх або, хоча б, їх частину.

Ті зображення, які погано стискаються звичайним алгоритмом стискання, тобто зображення, у яких високий коефіцієнт ентропії, стискаються розробленим алгоритмом.

Проте алгоритм не справляється із зображеннями, в яких немає великих подібних областей. Таким чином, він може бути корисний у випадках, коли маємо справу із зображеннями різних розмірів, в яких спостерігаються схожі фрагменти, або із зображеннями однакових розмірів, в яких спостерігаються схожі фрагменти, наприклад, відео з малим FPS.

Література

1. Р. Вудс. Цифровая обработка изображений / Р. Вудс. — Москва: Техносфера, 2005;
2. Стискання на основі передбачення значень пікселей. [Електронний ресурс].
https://www.csd.uwo.ca/~melsakka/publications/journals/pdfs/2007_jvcir_Nathan_ael.pdf
3. Стискання на базі передбачення збігу значень довколишніх в двумірному сенсі пікселей зображення. [Електронний ресурс].
<https://naun.org/main/NAUN/computers/17-679.pdf>
4. Вейвлет перетворення для стискання зображень. [Електронний ресурс]
https://www.researchgate.net/publication/266018963_Wavelet_image_compression
5. Стандарт JPEG. [Електронний ресурс]
<https://web.stanford.edu/class/ee398a/handouts/lectures/08-JPEG.pdf>.

УДК 004.4'24

магістрант Гулько Д.Т., к.т.н., доцент Заболотня Т.М.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

МЕТОД АВТОМАТИЗОВАНОГО ГЕНЕРУВАННЯ КОДУ ШАБЛОНІВ ПРОГРАМНИХ КОМПОНЕНТІВ КЛІЄНТСЬКОЇ ЧАСТИНИ ВЕБЗАСТОСУНКІВ

Abstract

Hulko Danylo, student; Tetiana Zabolotnia, assoc. prof., PhD

The method of automatic generation of program components templates for the frontend side of web applications

This work is devoted to the method of automatic code generation for the frontend program components. The developers spend most of their time to translate the requirements into the programming components, including markup, styles, and scripts for them. That task can be achieved by analyzing the description of the components written in natural human language and generating the code with a neural network approach. That will allow the developers to quickly and effectively implement the business tasks and features.

Вступ

Із розвитком галузі інформаційних технологій підтримка швидкого написання програмного коду стала надактуальною задачею з точки зору бізнесу, який завжди вимагає якнайскорішого отримання готового рішення своїх завдань. Зрозуміло, що жертвувати якістю готового програмного продукту задля пришвидшення його випуску не можна. Також для команди програмістів важливими є читабельність та модульність розроблюваного коду застосунку, оскільки робота в команді передбачає взаємодію з чужим кодом. Код, в такому випадку, має бути чітким та зрозумілим не лише для автора, а і для його колег по проєкту. Все це ставить розробника у ситуацію, коли є необхідність писати код швидко і якісно одночасно.

У практиці програмування «чистий» код та полегшення його подальшої підтримки забезпечується рядом підходів, як-от: винесення окремих методів, класів та об'єктів до незалежних утиліт; дотримання практик об'єктно-орієнтованого програмування; надання змістовних імен для змінних та функцій тощо. Також для того, щоб забезпечити вимоги щодо якості та швидкості створення коду, розробники часто звертаються до пошуку готового чи часткового рішення поставленої

перед ними задачі із подальшою його модифікацією. Але такий підхід несе в собі прихований ризик додавання до проєкту зайвої функціональності, що може виходити за рамки вирішуваної задачі.

Не менш популярним підходом до пришвидшення написання якісного коду є використання спеціальних програмних процедур для генерування або вставки шаблонів готового коду до структури проєкту. Так, наприклад, існують спеціальні пакети для генерування заготовок програмних компонентів, модулів і бібліотек для фреймворків фронтенд розробки.

Проте, на думку авторів, більш ефективним рішенням задачі підтримки швидкого розроблення якісного коду вебзастосунків може бути забезпечення можливості генерування окремих зручних для повторного використання шаблонних графічних компонентів, оскільки багато сайтів містять ряд схожих елементів інтерфейсу, які завжди створюються вручну: слайдери, вкладки, таблиці, елементи форм, фігурні елементи тощо. Складність реалізації такого рішення полягає у необхідності опрацювання вхідних даних, у ідеальному випадку, поданих у формі описів природною формалізованою мовою. На сьогоднішній день дана проблема не має однозначного і повного рішення у сфері розроблення клієнтської частини вебзастосунків, тому робота над її розв'язанням є актуальною задачею.

Мета дослідження

Метою даного дослідження є визначення методу автоматизованого генерування коду шаблонних прикладних компонентів для розроблення клієнтської частини вебзастосунків на основі вхідних даних, що подані формалізованим текстом.

Термінологія

Рекурентні нейронні мережі (англ. Recurrent neural networks; RNN) – вид штучних нейронних мереж, у яких зв'язки між вузлами утворюють направлену послідовність. Рекурентні мережі можуть використовувати внутрішню пам'ять для оброблення послідовностей довільної довжини [1].

Рекурсивні нейронні мережі (англ. Recursive neural network; RNN) – вид глибинних нейронних мереж, що працюють з даними змінної довжини. Такі нейронні мережі є зручними і ефективними у навчанні деревних структур при обробленні природномовних даних, головним чином неперервних представлень речень на основі векторного подання слів[2].

Механізм уваги – методика імітації когнітивної уваги для

підсилення важливих частин вхідних даних та пригнічення значення решти цих даних. Для нейронної мережі ця маленька частина даних є важливою, тому потребує ретельнішого оброблення [3].

LSTM (long short-term memory) – архітектура рекурентних нейронних мереж, яка передбачає наявність LSTM-вузлів. Можливість вузлів до запам'ятовування значень для довгих і коротких проміжків часу забезпечується використанням функції активації у межах їх рекурентних складових [4].

Розроблений метод автоматизованого генерування коду шаблонів програмних компонентів клієнтської частини вебзастосунків

Перетворення текстів природною мовою, написаних користувачем для опису майбутнього компоненту, у безпосередній код цих компонентів не може не нагадати задачу з перекладу тексту між двома мовами. Основна відмінність нашої ситуації полягає у тому, що вхідні і вихідні дані підпорядковуються абсолютно різним правилам побудови лексичних структур. На додаток, генерування коду програмних компонентів клієнтської частини вебзастосунків передбачає формування трьох складових: розмітки гіпертексту, набору правил стилів та функціонального коду поведінки компоненту.

Задача з перекладу тексту класично вирішується за допомогою використання нейронних мереж, що є доцільним і у випадку задачі генерування коду на основі опису компонента. Проте з метою спрощення даної задачі пропонується попередньо провести синтаксичний аналіз природномовного опису компонента для виявлення логічного зв'язку між значимими словами та підготовки вхідних даних. Таким чином, основною гіпотезою, що висувається в даному дослідженні, є забезпечення ефективного генерування програмного коду графічних компонентів шляхом поєднання попереднього синтаксичного оброблення природномовного опису компонента та подальшого опрацювання результатів аналізу за допомогою нейронних мереж. Використання такого підходу дозволить побудувати синтаксично правильні програмні компоненти.

Наведемо нижче кроки щодо реалізації висунутої гіпотези.

За допомогою синтаксичного аналізу можна програмно визначити ключові слова, які мають вказувати на те, які саме стилі, структура і логіка має бути у майбутнього компонента. Отже, перед безпосереднім опрацюванням тексту нейронними мережами необхідно виділити важливі для генерування компонента слова та прибрати все, що не має сенсу, як, наприклад, символи табуляції та пунктуації.

Після отримання результатів синтаксичного аналізу необхідно передати їх на подальше опрацювання нейронною мережею для отримання вихідного згенерованого коду.

Програмний код можна розглядати як послідовність символів або токенів, проте у такому випадку нейронна мережа має вчитись не лише перекладати текст у програмний код, але й писати код синтаксично правильно, що є неефективним з точки зору затрат часового ресурсу. Більш швидкий та ефективний підхід полягає у поданні програмного коду у вигляді абстрактних синтаксичних дерев (AST-структура). Таким чином нейрона мережа буде мати тип «Послідовність-до-Послідовності», причому вихідна послідовність буде деревовидною структурою. Як правило, для розв'язання задачі з перекладу тексту між двома мовами використовуються рекурентні нейронні мережі типу «Послідовність-до-Послідовності», але у даному випадку через роботу із деревовидною структурою більш доцільним є використання рекурсивних нейронних мереж. Така нейрона мережа передбачає використання на кожному вузлі структури певної функції:

$$g(x_i, s_1, \dots, s_n),$$

де x_i – значення i -го вузла, s_j – представлення, отримане шляхом рекурсивного застосування цієї ж нейронної мережі до дочірніх вузлів кожного вузла деревовидної послідовності. Як правило, у рекурсивних нейронних мережах ця функція будується на основі LSTM, або, у випадку деревовидної структури, – TreeLSTM.

Етап методу щодо генерування коду пропонується реалізувати з використанням шаблону «Кодувальник-Декодувальник». А для підвищення детальності аналізу вхідних даних пропонується використати механізм «уваги».

Кодувальник приймає на вхід оброблене бінарне синтаксичне дерево, проводить його через LSTM-подібну нейронну мережу, утворюючи на виході деревовидну структуру, кожен вузол якої представляє вектор фіксованої довжини. Цей вектор називається станом вузла.

Дана деревовидна структура далі опрацьовується Декодувальником, який пропускає його через власну нейронну мережу на основі LSTM, утворюючи абстрактне дерево програмного коду цільового компоненту.

Процес перетворення абстрактного синтаксичного дерева на код, складений цільовою мовою програмування, є класичною задачею, тому пропонується використання сторонніх бібліотек для реалізації цього кроку. Так, наприклад, з отриманого дерева може бути сформований текст розмітки, стилів та коду програмного компоненту мовою TypeScript у рамках фреймворку Angular. Даний текст буде записаний у відповідні файли, що знаходяться у директорії, вказаній користувачем.

Таким чином, на основі отриманих результатів дослідження можна сформулювати метод автоматизованого генерування коду компонентів клієнтської частини вебзастосунків на основі їх опису природною мовою користувача:

1. Перетворення вхідного тексту у деревовидну структуру значимих лексем за допомогою синтаксичного аналізу.
2. Перетворення вхідної структури на абстрактне синтаксичне дерево цільового програмного компоненту за допомогою рекурсивних LSTM нейронних мереж та шаблону «Кодувальник-Декодувальник».
3. Генерування розмітки, стилів та коду цільового компоненту клієнтської частини вебзастосунку за допомогою спеціалізованих бібліотек.

Висновки

У даному дослідженні запропоновано узагальнене формулювання методу автоматизованого генерування коду шаблонних прикладних компонентів клієнтської частини вебзастосунків на основі вхідних даних, що подані формалізованим текстом, чого не пропонують існуючі рішення з генерування шаблонних частин проєкту. Даний метод передбачає використання нейронних мереж для оброблення текстового вводу користувача та його переклад у відповідний код.

Використання даного методу приєє спрощенню процесу написання програмного коду за рахунок зменшення часу на залучення частково-готового рішення, яке потребує мінімальних змін для забезпечення вимог проєктних задач. Завдяки комбінації синтаксичного оброблення тексту і процесу перекладу з природньої мови на програмний код компонентів, суттєво збільшується ефективність написання додатку за критерієм витраченого на це часу.

У подальших дослідженнях пропонується провести аналіз відповідності згенерованого програмного коду до запиту користувача і пошук шляхів до покращення швидкодії отриманого метода генерування коду за допомогою використання нейронних мереж.

Література

1. *Allamanis, M., Barr, E.T., Devanbu, P., Sutton, C.: A Survey of Machine Learning for Big Code and Naturalness* – [Електронний ресурс] – Режим доступу: <http://arxiv.org/abs/1709.06182><http://arxiv.org/abs/1709.06182>.

2. Birch, A., Osborne, M., Koehn, P.: *Predicting success in machine translation. Proceedings of the Conference on Empirical Methods in Natural Language Processing* – [Электронный ресурс] – Режим доступа: doi.org/10.3115/1613715.1613809
3. Y. Oda, H. Fudaba, G. Neubig, H. Hata, S. Sakti, T. Toda, and S. Nakamura, Learning to generate pseudo-code from source code using statistical machine translation – [Электронный ресурс] – Режим доступа: <https://ieeexplore.ieee.org/document/7372045>
4. Sak, Haşim; Senior, Andrew; Beaufays, Françoise (2014). *Long Short-Term Memory recurrent neural network architectures for large scale acoustic modeling.* – [Электронный ресурс] – Режим доступа: <https://static.googleusercontent.com/media/research.google.com/en/pubs/archive/43905.pdf>

УДК 004.4'233

магістрант Демченко В.М., к.т.н., доцент Заболотня Т.М.

Національний технічний університету України
«Київський політехнічний інститут імені Ігоря Сікорського»

МЕТОДИ СТАТИЧНОГО АНАЛІЗУ ВИХІДНОГО КОДУ КОРИСТУВАЧА ЗАСОБАМИ МОВИ C#

Abstract

Vladyslav Demchenko student; Tetiana Zabolotnia, assoc. prof., PhD

Methods of static analysis of user source code using C# programming language

This work is devoted to the methods of static analysis of the source code using the C# language. Developers spend a lot of time conducting code reviews, looking for code that may have defects. Modern static analysis tools allow you to find vulnerabilities and shortcomings of the source code automatically in a couple of seconds. Using these systems, you can save developers' time and, as a result, the customer's money. In this work, we will consider the existing methods of static analysis and also propose a hypothesis for improving the characteristics of already existing methods.

Вступ

З року у рік розвиток ІТ-технологій є постійним явищем у нашому інформаційному суспільстві: все більше людей приходять у сферу розроблення програмного забезпечення, а програми стають все більш комплексними. Задля спрощення та пришвидшення розроблення програмного забезпечення створюються бібліотеки та фреймворки, що реалізують необхідні складові функціональності розроблюваної програми. Можна сказати, що, на сьогоднішній день, використання сторонніх бібліотек та фреймворків вже стало невід'ємною частиною роботи розробника. Проте неможливо гарантувати, що використана бібліотека не має недоліків: що її код реалізує найефективніші алгоритми, що вона стійка до некоректного вводу даних тощо. Це призводить до того, що використання ненадійного стороннього програмного забезпечення робить розроблювану систему також вразливою. І це в той час, коли одними з найважливіших критеріїв якості програмного забезпечення були та залишаються надійність та відмовостійкість.

На допомогу розробникам у таких ситуаціях приходять інструменти для проведення статичного та семантичного аналізу програмного коду. Кожен з них використовує свої механізми знаходження погано написаного коду та надає можливість отримати детальний звіт, у якому відображені потенційні проблеми розроблюваної системи, рекомендації

для покращення існуючої кодової бази та інші характеристики (в залежності від того, які саме методи аналізу використані). Також методи статичного аналізу вихідного коду можуть ефективно використовуватись в рамках проведення процедури код-рев'ю (коли більш кваліфіковані спеціалісти рецензують код менш кваліфікованих): наприклад, вони допомагають заощаджувати час при огляді нововведень у кодовій базі системи. Таким чином, рецензент (reviewer) може зосередитись на архітектурній складовій або читабельності коду, в той час як проблеми іменування, відсутність крапок з комою, недосяжні блоки коду тощо може взяти на себе система статичного аналізу.

Звісно, існуючі системи статичного аналізу програмного коду не можуть повноцінно замінити рецензента, адже вони нездатні слідкувати за тим, наскільки читабельним є код, або за тим, наскільки архітектурно правильно побудована розроблювана система. Тим не менш, швидке та коректне виконання ними оброблення програмного коду безперечно сприяє продуктивній подальшій роботі фахівців з цим кодом, дозволяючи економити час на виконанні рутинних перевірок. Таким чином, дослідження способів підвищення ефективності методів статичного аналізу програмного коду, зокрема за показником швидкості їх виконання, є актуальною задачею.

Мета дослідження

Мета даної роботи полягає у обґрунтованому формулюванні гіпотез щодо шляхів прискорення виконання статичного аналізу вихідного програмного коду за рахунок дослідження існуючих методів статичного аналізу коду, виявлення їх переваг та недоліків та вивчення можливих способів їх комбінування чи модифікації.

Термінологія

Абстрактне синтаксичне дерево (англ. abstract syntax tree) – це скінченна множина вершин, позначене і орієнтоване дерево, в якому внутрішні вершини співставлені з відповідними операторами мови програмування, а листя – з відповідними операндами [1].

Taint-аналіз (англ. taint checking) – це метод аналізу коду, що дозволяє відстежувати поширення неперевіраних зовнішніх даних програми під час її роботи. Потрапляння таких даних у деякі ключові точки вихідного коду може призводити до виникнення різних безпекових вразливостей, серед яких SQL injection, cross-site scripting (XSS), path traversal та інші. Ці вразливості можуть бути використані зловмисником для порушення коректної роботи програмної системи, отримання конфіденційних даних або виконання інших несанкціонованих операцій [2].

Data-flow аналіз — це метод, що використовується для аналізу, яким чином значення змінних змінюються під час виконання програми. Отримані результати дають можливість оптимізувати або налагодити роботу програмного забезпечення [3].

Існуючі методи статичного аналізу програмного коду

Data-flow аналіз

Одним з методів статичного аналізу, що дає досить детальні результати, є метод, який називається data-flow аналіз. Основними етапами цього методу є [3]:

- 1) парсинг вхідного коду;
- 2) побудова графу зв'язків між класами;
- 3) аналіз побудованого графу зв'язків, зокрема пошук використаних шаблонів проєктування.

На етапі аналізу побудованого графу зв'язків в якості даних використовується граф вхідного коду, тобто оголошення класів, оголошення інтерфейсів, публічні методи, властивості та їх типи, а також граф колекції шаблонів. Отже, структуру графа складають вершини, що представляють класи, та ребра, що представляють їх залежності. Через ребра (залежності) з одного класу до іншого можуть передаватися стани системи, наприклад змінні. Також у кожній вершині (кожного класу) є деякий стан, наприклад приватні змінні, константи, методи та інше.

Одним зі способів проведення data-flow аналізу є застосування ітераційного алгоритму. Алгоритм починається з апроксимації стану кожного блоку коду. Далі обчислюється деякий вихідний стан шляхом застосування функції трансформації (виклику методів, операції розгалуження тощо). Внутрішні стани оновлюються, застосовуючи операції об'єднання змінних для різних потоків виконання (наприклад в блоках *if-else*), тобто формуються всі можливі значення, які може приймати змінна. Простіше кажучи, можна сформулювати рівняння[4]:

$$\begin{aligned} out_b &= trans_b(in_b) \\ in_b &= join_{b+prev(b)}(out_b) \end{aligned}$$

- b – блок коду;
- in/out – стан змінних на вході та виході відповідно;
- $trans$ – функція, що змінює стан змінних у блоці;
- $join$ – функція, що об'єднує стан змінних в різних потоках виконання.

Останнє рівняння повторюється, поки ми не досягнемо так званої «точки фіксації»: ситуації, в якій внутрішні стани не змінюються.

Наведений вище алгоритм має суттєвий недолік, який полягає у тому, що для перебору всіх можливих варіантів вхідних параметрів та їх

об'єднання у поточний стан коду потрібно виконання багатьох ітерацій, і тому алгоритм може працювати досить повільно. Для вирішення цього недоліку розроблена модифікація, яка називається «зворотний аналіз». Суть зворотного аналізу майже така сама, як і у прямого: він працює аналогічно, використовуючи ітераційний алгоритм. Відмінність полягає у тому, що функція передачі застосовується до стану виходу, що дає стан входу, а операція приєднання працює над станами входу наступних блоків (вершин графу), щоб отримати стан виходу.

Taint-аналіз

Аналіз на «чистоту» - це аналіз того, як ненадійні (або «брудні» - від англ. «taint») дані обробляються програмою. Ненадійними (taint) даними називаються дані, що приходять від користувача через поля вводу, читання завантаженого користувачем файлу чи іншими шляхами. Протилежні ним – надійні (trusted) дані. Іншими словами, до змінних додаються властивості (прапорці) taint та trusted. Taint означає, що дані прийшли з ненадійного ресурсу. В контексті безпеки, аналіз на «чистоту» використовується для відслідковування чутливої чи ненадійної інформації. Важливо проаналізувати, чи можуть ненадійні дані досягти виходу, тобто споживачів інформації. Споживачами інформації називають зазвичай місця виводу даних користувачу. У разі, якщо ненадійні дані можуть досягти таких виходів, можна говорити про імовірність наявності вразливостей в тій частині коду. Прикладом такої ситуації може слугувати випадок, коли користувач може змусити застосунок виводити довільні чи заборонені системою дані іншому користувачу. Інший приклад: якщо ненадійні дані розглядаються як чутлива інформація, то вони можуть бути втрачені. Дані можуть втратити taint властивість, якщо вони проходять через sanitation функцію (функцію очистки), у такому разі вони стають надійними.

Отже, даний метод дозволяє знаходити вразливості в системі, зокрема SQL-ін'єкції та можливий витік конфіденційних даних. Зазвичай виділяють такі його етапи:

- 1) парсинг вхідного коду;
- 2) побудова графу потоку управління – Control-flow graph (CFG);
- 3) впровадження прапорців (taint та trusted) до змінних згідно із заданими правилами
- 4) розповсюдження taint та trusted прапорців у системі.

Алгоритм Хіндлі-Мілнера

В загальному випадку алгоритм Хіндлі-Мілнера створено для виводу найбільш відповідних поліморфних типів в мовах програмування,

заснованих на системі типів Хіндлі-Мілнера, таких як: ML, Standard ML, Haskell, F# та інші.

Система типів визначається моделлю Хіндлі-Мілнера таким чином:

1. Примітивні типи ν є типами виразів.
2. Параметричні змінні типів α є типами виразів.
3. Якщо σ_1 і σ_2 — типи виразів, то тип $\sigma_1 \rightarrow \sigma_2$ є типом виразів.
4. Символ \perp є типом виразів.

Вирази, типи яких обчислюються, визначаються стандартним чином:

1. Константи є виразами.
2. Змінні є виразами.
3. Якщо e_1 і e_2 – вирази, то $(e_1 e_2)$ – вираз.
4. Якщо v – змінна, а e – вираз, то $\lambda v. e$ вираз.

Говорять, що тип σ_1 є екземпляром типу σ_2 , коли є певне перетворення p таке, що: $\sigma_1 = p(\sigma_2)$.

При цьому, зазвичай вважається, що у перетворення типів накладаються обмеження:

1. $p(\sigma_1 \rightarrow \sigma_2) = p(\sigma_1) \rightarrow p(\sigma_2)$.
2. $p(\nu) = \nu$.

Сам алгоритм виведення типів складається з двох кроків — генерації системи рівнянь і наступного розв'язання цих рівнянь.

Формулювання гіпотези щодо шляхів прискорення виконання статичного аналізу вихідного програмного коду

Якщо звернути увагу на той факт, що перші два кроки першого та другого методів збігаються, то логічно припустити, що, об'єднавши ці методи, можна отримати збільшення швидкодії роботи комбінованого метода. Ще одним гіпотетичним способом підвищення ефективності методів статичного аналізу коду є використання алгоритму Хіндлі-Мілнера для виведення типів, що забезпечить зменшення кількості варіантів поліморфних викликів в data-flow аналізі. Простіше кажучи, чим точніше зазначений вхідний стан, тим менше варіантів потрібно перебрати для знаходження точки фіксації.

Якщо розглядати вищезазначену гіпотезу та її реалізацію для мови програмування C#, то є зрозумілим, що ця мова не заснована на системі типів Хіндлі-Мілнера, а отже, можливі хибні та помилкові спрацювання алгоритму виводу типів. Тому застосування цього алгоритму при опрацюванні мови C# інколи може бути недоречним, або тільки збільшить кількість можливих варіантів для проведення data-flow аналізу. Зважаючи на все вищезгадане, потрібно модифікувати алгоритм Хіндлі-Мілнера згідно синтаксису та системи типів мови C#, та після його відпрацювання

перевіряти, чи у результаті не було виведено ще більш загальний тип, ніж зазначено у вихідному коді програми.

Таким чином, можна узагальнено сформулювати модифікований метод статичного аналізу вихідного коду користувача з покращеною швидкістю:

- 1) парсинг вхідного коду;
- 2) побудова графу зв'язків між класами;
- 3) робота алгоритму Хіндлі-Мілнера та внесення змін в граф зв'язків між класами;
- 4) застосування ітераційного алгоритму для проведення data-flow аналізу;
- 5) застосування алгоритму знаходження вразливостей.

Важливо відмітити, що останні два кроки можуть працювати паралельно маючи спільний граф зв'язків, адже жоден з алгоритмів не модифікує граф зв'язків. Проте задля покращення результатів алгоритм Хіндлі-Мілнера слід виконати перед застосуванням алгоритмів з інших методів.

Висновки

У даній статті розглянуто методи статичного аналізу коду, а також алгоритм виводу типів Хіндлі-Мілнера, проаналізовано їх переваги та недоліки, а також сформульована гіпотеза щодо побудови модифікованого методу статичного аналізу коду мови C# за рахунок поєднання існуючих методів.

За основу пропонується взяти метод data-flow аналізу, тому що він містить необхідну логіку для виконання статичного аналізу. В парі з taint-аналізом, в якому ми бачимо складову додавання taint та trusted властивостей до змінних, ми отримуємо комбінований метод, який дозволить отримати звіти з результатами taint та data-flow аналізів за менший проміжок часу. Ще одним варіантом для підвищення швидкодії комбінованого методу є застосування алгоритму Хіндлі-Мілнера, що дозволить зменшити кількість поліморфних викликів під час data-flow аналізу.

Література

1. ASTs - What are they and how to use them. – [Електронний ресурс] – Режим доступу: <https://www.twilio.com/blog/abstract-syntax-trees>;
2. Taint analysis (taint checking) – [Електронний ресурс] – Режим доступу: <https://pvs-studio.com/en/blog/terms/6496/>;

3. Data-flow analysis – [Электронный ресурс] – Режим доступа: <https://www.geeksforgeeks.org/data-flow-analysis-compiler/>;
4. Pavel Belicov (2018) How Data Flow analysis works in a static code analyzer – [Электронный ресурс] – режим доступа: https://www.slideshare.net/Andrey_Karpov/data-flow-95827404.

УДК 004.021

Д.т.н., професор Легеза В.П., магістрант Атаманюк О.В.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

АЛГОРИТМІЧНИЙ МЕТОД ПОБУДОВИ ЦИФРОВОГО ДВІЙНИКА ОДНОГО ВІБРОЗАХИСНОГО ПРОЦЕСУ

Abstract

Viktor P. Legeza, full prof., doctor of sciences (D.Sc.); Oleksii V. Atamaniuk,
Undergraduate

Algorithmic Method of building of the digital twin of one vibration process.

This paper concerns the task of building of the digital twins of a vibro-protective system for the tall flexible structure. The method of determination of optimal settings for ball vibration absorber as a part of digital twin is proposed as well. The performance of the method was investigated and the modification for the speed up was proposed. It is shown that the use of BVA makes it possible to four times reduce the level of amplitudes of forced vibrations of high-rise thin structures.

Вступ

Технологія цифрових двійників набуває все більшого значення, як для виробничих підприємств, так і для побутового життя. Попри всю потужність технології та велику кількість переваг вона не завжди може бути застосована через ряд різноманітних причин. Одним із таких випадків є створення цифрового двійника віброзахисного процесу для висотних гнучких споруд. При експлуатації висотних гнучких зазвичай виникають вимушені коливання, боротьба з якими перетворюється на велику технічну проблему. До останнього часу для розв'язання цієї проблеми використовувались динамічні гасники маятникового типу на підвісі [1]. Такий тип гасників доволі легко моделюється, тому створення цифрового двійника для них не викликало труднощів. З часом склалася ситуація, коли маятникові гасники не можуть бути застосовані через свої фізичні характеристики. В цих умовах найбільш ефективним є новий метод віброзахисту низькочастотних висотних споруд із використанням гасителів коткового типу [1-3]. У ході дослідження було розроблено математично обґрунтований метод оптимального налаштування основних параметрів кульових гасників [2]. Попри це відсутність розробленого алгоритмічно-програмного методу побудови цифрових двійників на основі створеного

методу із використанням гасителів коткового типу стримує їх впровадження у сучасну практику віброзахисту висотних гнучких об'єктів

Постановка задачі. В роботі розглядається процес придушення вимушених коливань віброзахисної системи з використанням кульових гасників (BVA - Ball Vibration Absorber). Принципова схема однієї із таких систем показана на рис. 1.

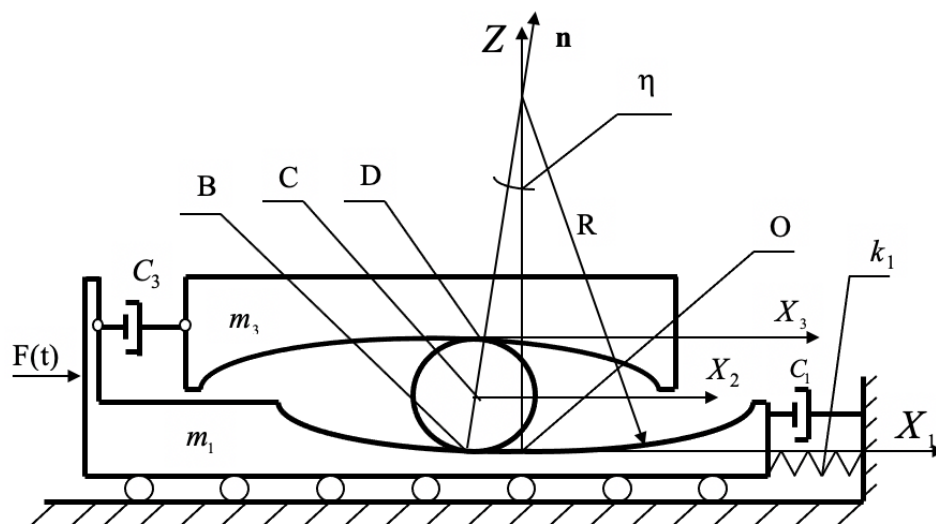


Рис. 1. Принципова схема кульового гасника

У ході попереднього дослідження [2] для такої системи було отримано рівняння амплітудно-частотної характеристики $A(\omega)$ системи “несуче тіло – котковий гасник”:

$$A(\omega) = \frac{\bar{F}_0 \sqrt{F_6(\omega)}}{\sqrt{F_6(\omega)F_7(\omega) + F_9(\omega) + F_{10}(\omega)}}, \quad (1)$$

де $F_1(\omega) = \omega_0^2 - \omega^2(1 + \nu)$; $F_2(\omega) = 2n_x \omega$; $F_3(\omega) = g - 2\bar{R}\omega^2$; $F_4(\omega) = 4n_r \bar{R}\omega$;

$F_5(\omega) = 2\nu \bar{R}\omega^4$; $F_6(\omega) = (F_3(\omega))^2 + (F_4(\omega))^2$; $F_7(\omega) = (F_1(\omega))^2 + (F_2(\omega))^2$;

$F_8(\omega) = F_2(\omega)F_4(\omega) - F_1(\omega)F_3(\omega)$; $F_9(\omega) = 2F_5(\omega)F_8(\omega)$; $F_{10}(\omega) = (F_5(\omega))^2$.

Для регулювання процесу віброзахисту цифровий двійник має передбачати процедуру пошуку оптимальних параметрів налаштування гасника. Виходячи з цих міркувань, сформулюємо мету даного дослідження.

Мета дослідження – розробити алгоритмічний метод створення цифрових двійників віброзахисного процесу, який забезпечить можливість визначення оптимальних регулюючих параметрів кульового гасника.

Метод створення цифрових двійників віброзахисного процесу.

Визначимо параметри віброзахисної системи, базуюсь на принциповій схемі системи (рис. 1) та отриманому рівнянню амплітудно-частотної характеристики (1).

У результаті отримуємо наступний набір параметрів:

- m_1 – маса несучого тіла;
- m_3 – маса робочого тіла;
- g – прискорення вільного падіння;
- ω_0 – кругова або циклічна частота;
- F_0 – максимальна амплітуда гармонічної збуджуючої сили;
- k_1 – коефіцієнт пружності несучого тіла;
- R – радіус виїмок;
- r – радіус кулі, що знаходиться всередині виїмок;
- ν – відношення мас робочого та несучого тіл;
- C_1 – коефіцієнт в'язкого опору демпфера, пов'язаного з несучим тілом;
- C_3 – коефіцієнт в'язкого опору демпфера, пов'язаного з робочим тілом;
- n_x – коефіцієнт демпфірування демпфера, пов'язаного з несучим тілом;
- n_η – коефіцієнт демпфірування демпфера, пов'язаного з робочим тілом;
- \bar{R} – різниця радіусів сферичних виїмок і кулі, що перекочується між ними.

Беручи до уваги те, що з будівельної точки зору віброзахисна система найлегше буде конфігуруватися через параметри саме робочого тіла гасника, тобто коефіцієнт демпфірування n_η та різницю радіусів виїмок та кулі \bar{R} , оберемо їх за вихідні параметри методу. У такому випадку ми будемо мати чотири вхідних параметри: \bar{F}_0 – максимальна амплітуда зовнішнього силового збудження; ν – відношення мас робочого та несучого тіл; n_x – коефіцієнт демпфірування демпфера, пов'язаного з несучим тілом; ω_0 – кругова частота власних коливань несучого тіла, яка є найменшою (або головною) з його спектру частот.

Оскільки було визначено вхідні та вихідні параметри, далі потрібно здійснити пошук оптимальних значень вихідних параметрів. Найпростішим способом знаходження оптимальних значень є звичайні ітеративні операції, де обидва вихідні параметри поступово інкрементуються, а отримані значення амплітуди при них зберігаються у певному словнику, а потім порівнюються. Такий спосіб є доволі очевидним та незалежним від умов, проте дуже повільним. З огляду на це

можемо виконати певну модифікацію, щоб пришвидшити метод побудови цифрового двійника. Для цього застосуємо графічний метод знаходження оптимальних параметрів настроювання гасників коткового типу [4-5]. В основі цього графічного методу лежить принцип «рівності двох максимумів», які досягаються на двох частотах ω_1 і ω_2 в околі резонансної частоти ω_0 . Як параметри настроювання розглядаються геометрична характеристика $\bar{R} = R - r$ гасника та коефіцієнт демпфірування n_η його робочого тіла.

Графічний метод оптимального налаштування гасника по частоті полягає у побудові фрагментів функціональних залежностей амплітуди A несучого тіла від характеристики \bar{R} гасника в околі двох частот ω_1 і ω_2 за умови фіксації інших параметрів системи. Перетин двох, знайдених таким чином фрагментів кривих АЧХ, і визначає оптимальну величину характеристик \bar{R} чи n_η гасника. Розглянемо детальніше графічний метод.

Спочатку параметри \bar{R} і n_η визначаються із умови рівності двох максимумів графіка функції $A = A(w)$, один з яких відповідає зведеній масі m_1 несучого тіла (споруди), а другий – масі m_2 робочого тіла гасника. Для початку роботи з методом знаходимо початкові частоти ω_1 і ω_2 , на яких досягаються два максимуми амплітудно-частотної характеристики для віброзахисної системи, за формулою (2) [5]:

$$\gamma_{1,2}^2 = \frac{1}{1+v} \left(1 \pm \sqrt{\frac{v}{2+v}} \right), \text{ де } \gamma_{1,2} = \frac{\omega_{1,2}}{\omega_0} \quad (2)$$

Далі графічно визначаємо точку перетину двох графіків A_1 та A_2 побудованих для фіксованих частот $\omega_1 = \gamma_1 \omega_0$ та $\omega_2 = \gamma_2 \omega_0$ при змінюванні параметра \bar{R} гасника, на яких функція $A = A(w)$ досягає двох рівних максимумів. Після визначення параметра \bar{R} гасника застосовуємо такий же підхід з побудовою двох графіків функції $A = A(w)$ для коефіцієнта демпфірування n_η .

Аналіз отриманих результатів застосування створеного методу

Запропонований метод створення цифрових двійників був реалізований за допомогою мови програмування Python та протестований на наборі різних даних, зібраних на основі реальних об'єктів [1]. Для виконання операцій з математичними функціями була використана бібліотека `sympy`. Побудований цифровий двійник віброзахисного процесу показав в середньому у чотири рази менші значення максимуму амплітуди коливань споруди, ніж цифровий двійник без гасника. З огляду на пришвидшення роботи методу для знаходження нулів математичних функцій були застосовані чисельні методи бібліотеки `sympy`. Окрім цього,

були виконані емпіричні заміри швидкодії оригінального методу та його модифікації з використанням графічного методу пошуку оптимальних параметрів. Графічний метод був реалізований за допомогою бібліотеки matplotlib. Такі модифікації дозволили пришвидшити метод в середньому втричі, зменшити навантаження на процесор та обробляти випадки, коли знаходження оптимальних параметрів цифрового двійника є доволі складною аналітичною задачею.

Висновки

Під час даного дослідження було розроблено алгоритмічний метод зі створення оптимального цифрового двійника, що відображає процес придушення небезпечних вимушених коливань висотних гнучких споруд на основі використання кульових гасників. Було описано та проаналізовано вхідні та вихідні параметри методу, способи їх отримання. Окрім цього, були виконані дослідження швидкодії методу за допомогою програмної реалізації цифрового двійника на мові Python. В результаті оцінки швидкості методу була запропонована модифікація методу, що дозволила скоротити час пошуку оптимальних параметрів цифрового двійника. Порівняльний аналіз показав високу ефективність функціонування запропонованої віброзахисної системи з оптимально налаштованими параметрами цифрового двійника.

Подальші дослідження будуть спрямовані на оцінку точності знаходження оптимальних параметрів цифрового двійника та оцінку швидкості його роботи.

Література

1. Легеза В.П. Теория виброзащиты систем с применением изохронных катковых гасителей: модели, методы, динамический анализ, технические решения. Lambert Academic Publishing, Saarbrücken, Deutschland. 2013.–108с.
2. В.П.Легеза, О.В.Атаманюк. Амплітудно-частотна характеристика віброзахисної системи та метод визначення оптимальних параметрів її кульового гасника // Доповідь на 14 науковій конференції ПМК-2021, КПІ, «Політехніка», 2021, С. 4 - 8.
3. Z.-L.Zhang, J.-Bing Chen, Jie Li. Theoretical study and experimental verification of vibration control of offshore wind turbines by a ball vibration absorber (BVA) // Structure and Infrastructure Engineering, V.10, #8, 2014, P.P. 1087-1100

4. Den Hartog, J.P. Mechanical Vibrations. Courier Corporation, 2013. – 464 p.
5. W.Weaver Jr., S.P.Timoshenko, D.H.Young. Vibration Problems in Engineering, 5th Edition, Wiley, 1991. - 624 p.

УДК 517.2

Д.т.н., професор Легеза В.П., студент К. Здоровенко,
к.ф.-м.н., доцент Нещадим О.М.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

РОЗВ'ЯЗАННЯ ОДНІЄЇ ЗАДАЧІ ПЕРЕСЛІДУВАННЯ В ЗАМКНЕНІЙ ФОРМІ

Abstract

The theme of these theses is about one of the classical pursuit problems is considered, when the point which chased, moves in a straight line. Solving this problem, we depicted the movement of the boats, the pursuit curve, created the differential equation of the second order. Pursuit was considered under the condition that one boat has a speed 2 times greater than the other, determined the meeting point, the time of pursuit, the route traveled. Finally, we generalized this problem to the case of different speed ratios of the boats.

Вступ

Розглядається одна з класичних задач переслідування, коли точка, яку переслідують, рухається по прямій лінії. Вперше її описав П'єр Бугер у 1732 році [1-3]. Пізніше П'єр Луї Мопертюї розглянув криву переслідування для інших випадків [4,5].

Постановка задачі. Розглянемо стартову позицію двох рухомих точок, одна з яких переслідує іншу (рис.1). На одному березі (позначеному на рис. 1 як вісь OY) водойми з паралельними берегами на відстані $y_0 = 1$ один від одного стоять два катери – катер берегової охорони U і катер порушників кордону V (рис.1). В початковий час $t=0$ катер V починає рухатись перпендикулярно до осі OY на протилежний берег водойми (вздовж осі OX , рис.1) зі швидкістю v . Катер U стартує одночасно з катером V і в процесі переслідування обирає напрям руху на поточну точку розташування катера V . При цьому катер U має швидкість u , яка в два рази більша за швидкість катера V : $u = 2v$.

Потрібно визначити: 1) траєкторію L руху катера U ; 2) час T , за який він наздожене катер V ; 3). точку на осі OX , в якій він наздожене катер V . Швидкість течії водойми не враховується.

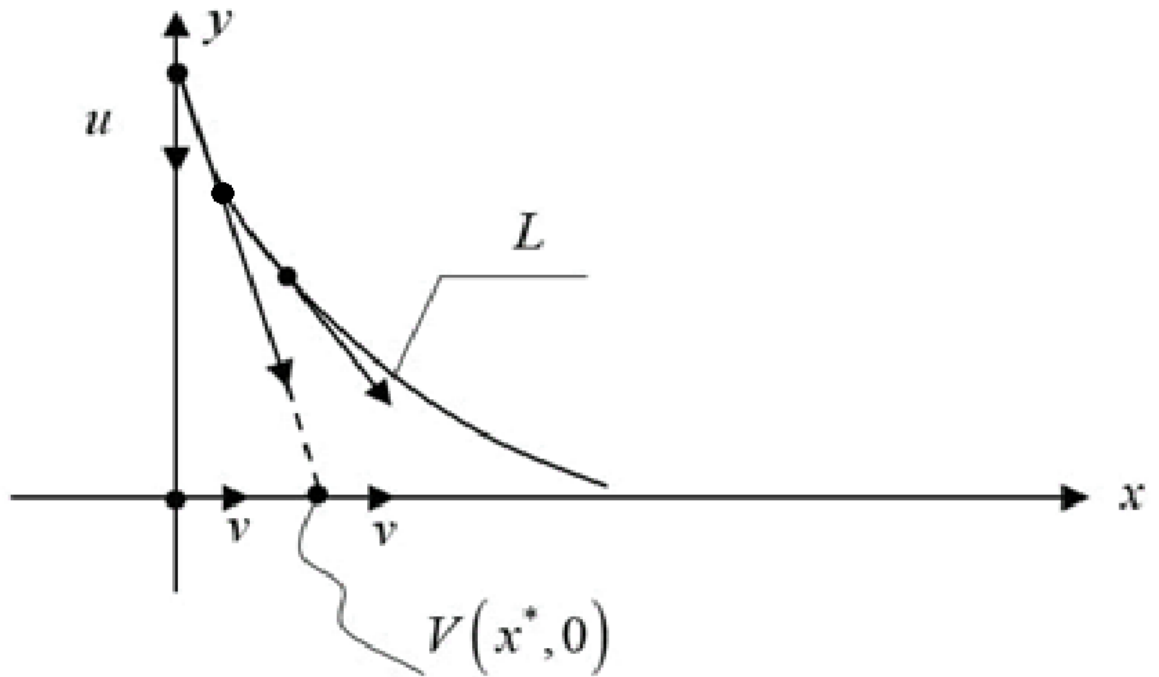


Рис. 1

Побудова і розв'язання диференціального рівняння траєкторії руху. Траєкторія L руху катера називається **кривою переслідування**.

Ототожнимо місцеположення катерів на площині OXY з точками U і V . Позначимо через (x, y) координати точки $U(x, y)$ і через x^* абсцису точки V в поточний момент часу t . Проведемо дотичну в точці $U(x, y)$ до кривої L , яка з'єднує точки U і V . Позначимо поточні координати дотичної через (ξ, η) і запишемо її рівняння:

$$\eta - y = y'(\xi - x). \quad (1)$$

З рівняння (1) дотичної до кривої L за умови $\eta = 0$ знаходимо абсцису точки її перетину з віссю OX , яка, за умовою, має дорівнювати x^* .

Отже, з рівняння (1) маємо

$$x^* = v \cdot t = x - \frac{y}{y'}. \quad (2)$$

Встановимо формулу для зв'язку між формою $y(x)$ траєкторії L та швидкістю u точки U :

$$u = \frac{dl}{dt} = \sqrt{1 + (y')^2} \cdot \frac{dx}{dt}. \quad (3)$$

Тепер знайдемо вираз для $\frac{dx}{dt}$, диференціюючи співвідношення (2):

$$v = \frac{dx}{dt} - \left(\frac{y}{y'}\right)' \frac{dx}{dt} = \frac{dx}{dt} \left(\frac{y''y}{(y')^2}\right) \Rightarrow \frac{dx}{dt} = v \frac{(y')^2}{y''y}. \quad (4)$$

Використовуючи співвідношення між швидкостями $u = 2v$, побудуємо шукане диференціальне рівняння кривої L переслідування:

$$u = \sqrt{1+(y')^2} \cdot v \frac{(y')^2}{y''y} \Rightarrow \frac{u}{v} = \sqrt{1+(y')^2} \cdot \frac{(y')^2}{y''y} = 2.$$

Таким чином, рівняння кривої переслідування має такий вигляд:

$$y'' = \sqrt{1+(y')^2} \cdot \frac{(y')^2}{2y}. \quad (5)$$

До рівняння (5) потрібно додати дві граничні умови:

$$y(x)|_{x=0} = y_0 = 1, y'(x)|_{x=0} = y'_0 = -\infty \quad (6)$$

В результаті отримали замкнену крайову задачу, яка складається з диференціального рівняння (5) та граничних умов (6).

Розв'яжемо рівняння (5). Оскільки рівняння (5) не містить змінну x в явній формі, то порядок рівняння можна знизити заміною $y' = p$:

$$p'p = \sqrt{1+p^2} \cdot \frac{p^2}{2y}.$$

Скорочення на множник p не призведе до втрати потрібних розв'язків, оскільки нас цікавить нетривіальний розв'язок, тому маємо остаточно:

$$p' = \sqrt{1+p^2} \cdot \frac{p}{2y}. \quad (7)$$

Розв'яжемо рівняння (7) відокремленням змінних:

$$\frac{dp}{p\sqrt{1+p^2}} = \frac{dy}{2y}. \quad (8)$$

Проінтегруємо ліву частину рівняння (8):

$$\int \frac{dp}{p\sqrt{1+p^2}} = \left| \begin{array}{l} p = \operatorname{tg} z \\ dp = \frac{dz}{\cos^2 z} \end{array} \right| = \int \frac{\frac{dz}{\cos^2 z}}{\operatorname{tg} z \cdot \frac{1}{\cos z}} = \int \frac{dz}{\sin z} = \ln \left| \operatorname{tg} \left(\frac{z}{2} \right) \right| + C = \ln \left| \operatorname{tg} \left(\frac{\operatorname{arctg} p}{2} \right) \right| + C.$$

Отже, маємо диференціальне рівняння першого порядку:

$$\ln \left| \operatorname{tg} \left(\frac{\operatorname{arctg} p}{2} \right) \right| + C = \ln(\sqrt{y}). \quad (9)$$

Знайдемо константу C з другої граничної умови (6), яка вказує на те, що дотична до кривої в початковий момент $t=0$ в точці $y(0) = y_0 = 1$ напрямлена вздовж осі OY вниз. Оскільки в початковій точці $|y'|_{t=0} = |p|_{t=0} \rightarrow \infty$, тому $C=0$. Далі, враховуючи, що в подальшому русі виконується нерівність $y' = p < 0$, знімемо логарифми в рівнянні (9) і отримаємо таке диференціальне рівняння першого порядку:

$$\operatorname{tg} \left(\frac{\operatorname{arctg} y'}{2} \right) = -\sqrt{y} \Rightarrow y' = -\operatorname{tg}(\operatorname{arctg}(2\sqrt{y})) \Rightarrow y' = \frac{2\sqrt{y}}{y-1}. \quad (10)$$

Рівняння (10) теж розв'язуємо відокремленням змінних:

$$\int \frac{(y-1)dy}{\sqrt{y}} = 2x + C_1.$$

Після інтегрування останнього рівняння дістанемо:

$$\frac{2}{3}y^{\frac{3}{2}} - 2\sqrt{y} = 2x + C_1 \Rightarrow \frac{1}{3}y^{\frac{3}{2}} - \sqrt{y} = x + C_0.$$

Константу C_0 знайдемо з першої з граничних умов (б) $y_0 = 1$: $C_0 = -\frac{2}{3}$

$$\frac{1}{3}y^{\frac{3}{2}} - \sqrt{y} = x - \frac{2}{3}.$$

Таким чином, маємо остаточне алгебраїчне рівняння кривої L переслідування:

$$x(y) = \frac{1}{3}y^{\frac{3}{2}} - \sqrt{y} + \frac{2}{3}. \quad (11)$$

Тепер з формули (11) можна знайти всі характеристики руху: точку зустрічі катерів на осі OX , час переслідування T та шлях s , який пройде катер U під час переслідування.

1). Точка зустрічі на осі OX визначається з умови, що $y = 0$: $x = \frac{2}{3}$.

2). Час переслідування $T = \frac{2}{3v}$.

3). Нескладно обчислити шлях, який пройде катер охорони U під час переслідування. Очевидно, що він дорівнює подвоєному шляху, знайденому в п. 1), оскільки швидкості катерів пов'язані співвідношенням $u = 2v$. Тобто шуканий шлях дорівнює: $s = 2x = \frac{4}{3}$.

В подальшому розв'язок цієї задачі було узагальнено на випадок довільного співвідношення швидкостей катерів $\alpha = u/v$, де α - дійсне число. Наведемо остаточний результат, опустивши проміжні викладки ($\alpha \neq 1$):

$$x(y) = \frac{\alpha}{2} \left(\frac{y^{\frac{\alpha+1}{\alpha}}}{\alpha+1} - \frac{y^{\frac{\alpha-1}{\alpha}}}{\alpha-1} \right) + \frac{\alpha}{\alpha^2 - 1}. \quad (12)$$

За умови $\alpha = 2$ формула (12) збігається із формулою (11).

Формула (12) також дає можливість встановити необхідну величину співвідношення швидкостей α в залежності від заданої точки затримання $x^* = x(0)$ катера-порушника V на осі OX . Величину α знайдемо, поклавши в формулі $y = 0$ та розв'язавши таке квадратне рівняння (підходить лише

один корінь, який більше одиниці): $x^* = \frac{\alpha}{\alpha^2 - 1} \Rightarrow \alpha = \frac{1 + \sqrt{1 + 4(x^*)^2}}{2x^*}$.

Наприклад, щоб наздогнати катер V в заданій точці $x^* = 0,5$, катеру U потрібно мати швидкість, в $\alpha = 1 + \sqrt{2}$ рази більше за швидкість катера V .

Для ілюстрації на малюнках рис.2-4 наведено графіки кривої переслідування L для трьох випадків: $\alpha = 2$, $\alpha = 5$, $\alpha = 10$. З них випливає той логічний факт, що для затримання катера-порушника V потрібно, щоб величина $\alpha = u/v$ задовольняла нерівність $\alpha > 1$. За умови збільшення величини α перетин кривої переслідування L з віссю OX відбувається значно раніше, тобто ближче до точки старту. За умови $\alpha \leq 1$ крива переслідування L взагалі не перетне горизонтальну вісь OX і факт затримання порушника не відбудеться.

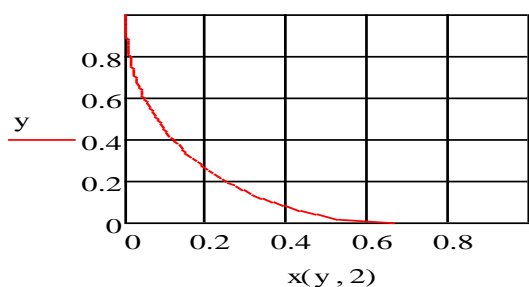


Рис. 2

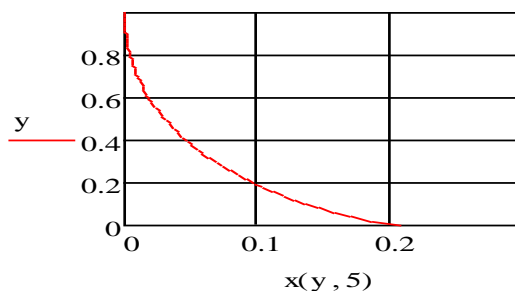


Рис. 3

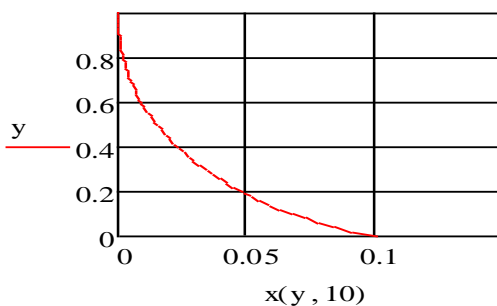


Рис. 4

Висновки

На основі узагальнення цієї задачі на випадок різних співвідношень $\alpha = u/v$ та виконаних числових експериментів зроблені наступні висновки.

За умови $\alpha > 1$ крива переслідування L обов'язково перетне пряму OX руху катера-втікача V і катер-переслідувач U наздожене катер V .

За умови $\alpha \leq 1$ крива переслідування асимптотично наближається до прямої руху OX катера V і катер U не наздожене катер V .

За раціонального значення $\alpha \neq 1$ крива переслідування є алгебраїчною кривою, за $\alpha = 1$ (при однаковій швидкості катерів V і U) крива переслідування перетворюється на трактрису.

Подальші дослідження у цьому напрямку доцільно проводити із врахуванням швидкості течії водойми.

Література

1. Бернхарт, А. «Криві переслідування-II». *Scripta Math.* **23**, 49-65, 1957.
2. Бернхарт, А. «Криві загального переслідування». *Scripta Math.* **24**, 189-206, 1959.
3. Архів історії математики. MacTutor. «Крива переслідування». <http://www-groups.dcs.st-and.ac.uk/~history/Curves/Pursuit.html> .
4. Вайштайн, Ерік В. «Крива переслідування». *MathWorld* -- веб-ресурс Wolfram. <https://mathworld.wolfram.com/PursuitCurve.html>.
5. Вайштайн, Ерік В. «Проблема траулера». *MathWorld* -- веб-ресурс Wolfram. <https://mathworld.wolfram.com/TrawlerProblem.html>.

UDC 004.415.2

Oleshchenko L. M., assoc. prof., PhD; Chao Wang, student

National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”

MACHINE LEARNING METHODS ANALYSIS FOR HUMAN ACTIVITY RECOGNITION

Анотація

К.т.н., доцент Олещенко Л. М., магістрант Ванг Чао

Аналіз методів машинного навчання для розпізнавання людської активності
У даній роботі проаналізовано ефективність поєднання класичних підходів машинного навчання для розпізнавання людської активності. Початкові вхідні вектори, отримані від датчиків акселерометра та гіроскопа смартфона, навчаються окремо різними алгоритмами машинного навчання та передаються в інший алгоритм машинного навчання для остаточних оцінок. Ця система використовує набори даних, створені із записів 30 осіб, які виконували повсякденні дії, маючи на поясі смартфон із вбудованими інерційними датчиками. Результати експерименту показують, що запропонована комбінація методів машинного навчання має кращу точність класифікації для розпізнавання активності людини порівняно з існуючими методами, застосованими окремо, а процес навчання є простим та ефективним.

Introduction

This experiment primarily focuses on the usage of classical algorithms that can significantly enhance performance on the time series data. Many deep learning algorithms suffer from the problem of overfitting. Sometimes the training data may not be a good representation of real time data, the ensemble methods proposed in this paper, not only increase the accuracy of prediction, but also the reliability of the performance on generalization.

Terminology

HAR – Human Activity Recognition has multifaceted applications due to its worldly usage of acquisition devices such as smartphones, video cameras, and its ability to capture human activity data.

SVM – Support Vector Machines are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis.

KNN – k-nearest neighbors algorithm (k-NN) is a non-parametric supervised learning method.

Problem statement

Motion sensors (accelerometer and gyroscope) on smartphones provide important information to facilitate the identification and monitoring of user movement [1]. Human activity recognition using smartphones is emerging as a trendy mobile application in healthcare monitoring and indoor location estimation. Analyzing the time-series sensor signals is challenging due to its spatial and temporal variances. There are numerous deep neural networks used for research on this topic. However, the drawback of deep neural networks is the hardness of explanation of the networks logic to achieve the output, and a huge training sample size is required to reach good performances.

The proposed method

Bagging, boosting, and stacking are the three major machine learning ensemble methods. For stacking method, usually we choose some different models as base algorithms, then a meta model at the end combines the base models. In this research a hybrid ensemble method is used. Some features in the processed dataset might contain redundant or irrelevant information that can negatively affect the recognition accuracy [2]. The structure of the modified ensemble method consists of two parts, the feature extraction unit and classification unit. The features of sensor data were extracted by composed of a nonensemble method and a bagging or boosting ensemble method to feed in a meta classification unit for identifying human activities. First, the major machine learning algorithms are used to train and test on the dataset, then the best algorithm is chosen as the nonensemble method. After that, feature selection is applied to the nonensemble method, then bagging and boosting methods are used to train and test on the trivial features, then the best bagging or boosting method is chosen. At the final, a meta classifier which is a nonensemble method is searched for the best performance.

Overall, it's a modified stacking method, but unlike stacking method, nonensemble method, bagging, boosting, and stacking are combined into one structure (fig. 1).

Algorithm for hybrid ensemble method:

1. Define T classifiers for selection.
2. Learn T classifiers on dataset.
3. Pick best classifier B from T and do feature selection for B and get feature set F and unselected feature set CF.
4. Define V classifiers for bagging and boosting.
5. Learn V classifiers on feature set CF.
6. Pick best classifier H from V.
7. Define S classifiers for another level.
8. Learn S classifiers on outputs of B classifier and classifier H.

9. Pick best classifier K from S.

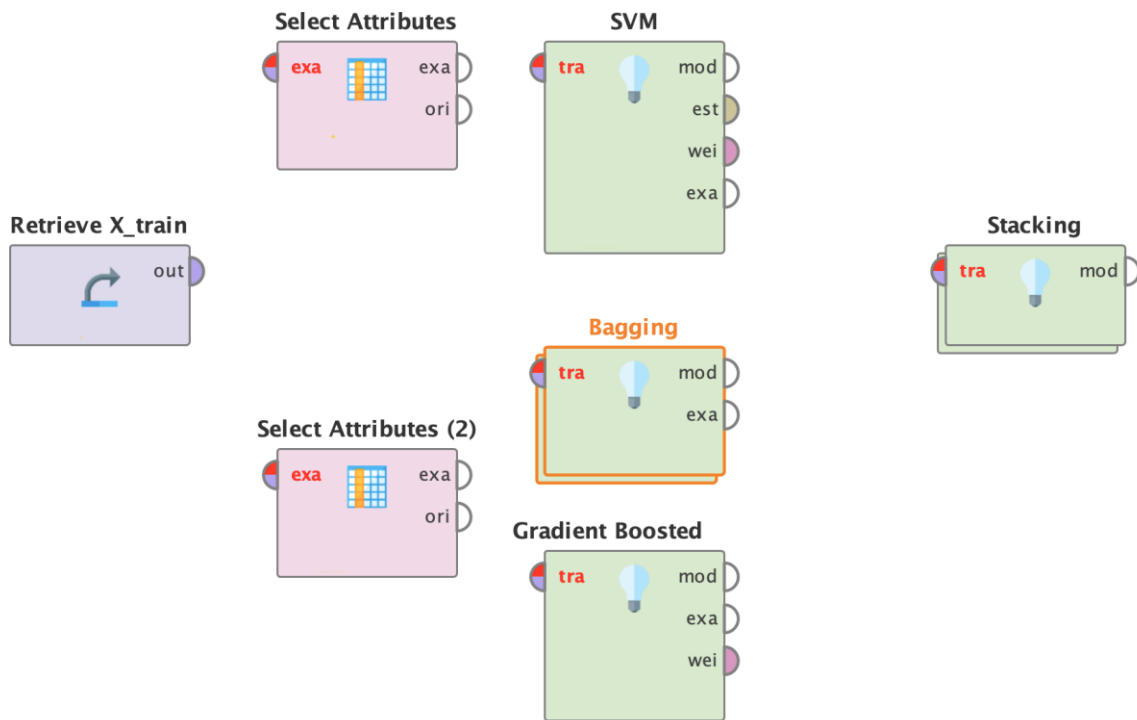


Fig.1. The specific structural diagram.

In research [3] the authors proposed a new approach for improving human activity recognition combined with principal component analysis, linear discriminant analysis and improved weighted Support Vector Machine (SVM) to overcome the problems of non-informative sequence features and class imbalance.

In research [4], fusion of various sensors for human activity recognition is explored, and the significance of the accelerometer and gyroscope is effectively analyzed.

In research [5], accelerometer and gyroscope data about 13 activities are collected, and machine learning algorithms such as multilayer perception, naive Bayes, logistic regression, and k-nearest neighbor are used to infer the human.

Based on these experiments, the set of the T, V, and S classifiers is chosen as follows:

- k-nearest Neighbors classifier;
- Decision Tree Classifier;
- Support Vector Machine;
- Gaussian Naive Bayes;
- Logistic Regression;
- Ensemble classifier candidates:
- Bagging Classifier;

- Random Forest Classifier;
- Extra-trees Classifier;
- Gradient Boosting;

Accuracy, balanced accuracy, weighted f1 score, weighted precision score and weighted recall score are chosen for the multiclass classification experiment analysis. The first key step for this experiment is to pick a best base classifier, the metrics details are as follow:

Table 1

Comparison of classifiers

Classifier	Accuracy	Balanced Accuracy	Weighted f1	Weighted Precision
KNN	0.89247	0.81456	0.89050	0.89715
Decision Tree	0.81151	0.72277	0.81196	0.81437
SVM	0.93674	0.83159	0.93605	0.93778
Gaussian Naive Bayes	0.74731	0.71089	0.74273	0.79360
Logistic Regression	0.90560	0.80811	0.90518	0.90639

Based on the outcomes above, the classifier SVM is chosen. The second key step for this experiment is to do feature selection for the chosen classifier, decision tree can help on this requirement, but a grid search is performed for this task. After this step, we get the CF feature subset. Then the ensemble classifiers get trained on the trivial feature subset.

Table 1

Comparison of classifiers

Classifier	Accuracy	Balanced Accuracy	Weighted f1	Weighted Precision
Bagging	0.74235	0.71635	0.76250	0.76517
Random Forest	0.71183	0.68258	0.71142	0.71532
Extra-trees	0.73674	0.73076	0.73104	0.73007
Gradient Boosting	0.75981	0.74627	0.74028	0.77867

Based on the outcomes above, the classifier Gradient Boosting is chosen.

The final step is to select the best classifier for the final predictions which dominates the overall performance, the complete classification report performances of the hybrid model is shown in the following table:

Table 1

Comparison of classifiers

Classifier	Accuracy	Balanced Accuracy	Weighted f1	Weighted Precision
Decision Tree	0.83657	0.82652	0.81628	0.82876
SVM	0.93876	0.85623	0.93765	0.93088
Gaussian Naive Bayes	0.84635	0.79620	0.84528	0.83916
Logistic Regression	0.94798	0.89631	0.95001	0.94625

Conclusions

In this research, a hybrid machine learning ensemble method with classical algorithms is developed for HAR using smartphones. The data features are efficiently and effectively extracted from the accelerometer and gyroscope data obtained by the mobile and wearable device, and the classification accuracy of HAR is improved. The hybrid machine learning structure achieved the best classification accuracy performance compared with the individual classifiers including k-nearest Neighbors classifier, Decision Tree Classifier, Support Vector Machine, Gaussian Naïve, and Bayes, Logistic Regression. Also, it surpassed the individual ensemble classifiers including Bagging,

Random Forest, Extra-trees Classifier and Gradient Boosting. More importantly, this hybrid model algorithm could generalize across different circumstances and suite different datasets well. And theoretically, any number of levels can be added into the hybrid model algorithm, two levels are found to be sufficient to solve this classification problem in this experiment. In practice, domain knowledge on the data will be helpful in configuring the model based on the nature of data and capacity of individual models for feature selection and model combing. The following can be taken up as future works:

- 1) Careful choice of individual models and their respective parameters could significantly help in improving the performance.
- 2) Finding an optimal way to do the feature selection to get the important feature subset, the trivial feature subset and the feature subset which should not be considered.
- 3) Apply this hybrid algorithm on more complex data and deep learning methods.

References

1. Incel, O. Analysis of Movement, Orientation and Rotation-based Sensing for Phone Placement Recognition. *Sensors* 2015, 15, 25474–25506.

2. Lara, O.D.; Labrador, M.A. A Survey on Human Activity Recognition Using Wearable Sensors. *IEEE Commun. Surv. Tutor.* 2013, 15, 1198
3. Abidine, B.M.H.; Fergani, L.; Fergani, B.; Oussalah, M. The Joint Use of Sequence Features Combination and Modified Weighted SVM for Improving Daily Activity Recognition. *Pattern Anal. Appl.* 2018, 21, 119–138
4. Shoaib, M.; Bosch, S.; Incel, O.; Scholten, H.; Havinga, P. Fusion of Smartphone Motion Sensors for Physical Activity Recognition. *Sensors* 2014, 14, 10146–10176
5. Wu, W.; Dasgupta, S.; Ramirez, E.E.; Peterson, C.; Norman, G.J. Classification Accuracies of Physical Activities Using Smartphone Motion Sensors. *J. Med. Internet Res.* 2012, 14, e130.

УДК 004.4:004.92

К.т.н., доцент Олещенко Л.М., магістрант Вовченко Д.С.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

ПРОГРАМНИЙ МЕТОД ПІДБОРУ ВАКАНСІЙ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ МАШИННОГО НАВЧАННЯ

Abstract

Oleshchenko Liubov, assoc. prof., PhD; Vovchenko Dmytro, student
Software method of selection of vacancies with the use of machine learning technologies

This paper concerns the analysis of common existing methods for machine learning. Their main characteristics are present. Also, requirements for their optimization are formulated. They were analyzed and among them was selected the most suitable one to use. Besides this, in this article were explained the ways to further optimize selected method.

Вступ

Майже кожен програміст за свою кар'єру зустрічався з процесом пошуку роботи, для деяких програмістів цей процес може повторюватись багато разів. Найкращим рішенням для такого пошуку є використання спеціальних вебсайтів, на яких роботодавці розміщують свої вакансії. Серед усього різновиду доступних вакансій зазвичай обирають декілька і уже далі шукачі зв'язуються з представниками компаній для подальшого обговорення пропозиції. У наш час на ринку праці з'являється все більше нових ІТ-компаній, яким потрібні нові працівники. Спостерігається зростання кількості вакансій, що, в свою чергу, може ускладнити процес пошуку необхідних спеціалісту пропозицій серед них. На вебсайтах для пошуку роботи вже є налаштовані фільтри, які допомагають оптимізувати процес пошуку, але навіть з їх допомогою нерідко можна знайти десятки пропозицій. В таких умовах виникає потреба у створенні програмної системи для виявлення найбільш перспективних вакансій з отриманого списку.

Постановка задачі

Метою даної є роботи оптимізація процесу відбору вакансій при пошуку роботи.

Об'єкт дослідження – процес відбору вакансій при пошуку роботи, предмет дослідження – методи машинного навчання.

Термінологія

ML (Machine Learning) – це підгалузь штучного інтелекту в галузі інформатики, яка застосовує статистичні прийоми для надання комп'ютерам здатності «навчатися».

TP (true positive) – параметр, який рахує, скільки разів модель машинного навчання правильно передбачає позитивний результат;

TN (true negative) – параметр, який рахує, скільки разів модель машинного навчання правильно передбачає негативний результат;

FP (false positive) – параметр, який рахує, скільки разів модель машинного навчання неправильно передбачає позитивний результат;

FN (false negative) – параметр, який рахує, скільки разів модель машинного навчання неправильно передбачає негативний результат;

SVM (support vector machines) – метод опорних векторів.

Аналіз вимог для відбору вакансій з використанням методів машинного навчання

Для розуміння вимог, необхідних для відбору вакансій, потрібно проаналізувати інформацію, з якою буде виконуватись робота. Більшість вакансій складається з декількох важливих пунктів: назва посади, необхідні навички, бажана характеристика майбутнього працівника, заробітня плата та привілегії роботи саме в цій компанії. Отже, ми маємо розділений на певні групи текст і нашою головною задачею є аналіз кожної з цих груп на предмет наявності бажаної інформації. Тобто завдання розробленої програмної системи зводиться до аналізу текстової інформації, і тому однією із головних вимог є швидкість та точність оброблення інформації.

Одним із найбільших недоліків застосування методів машинного навчання є швидкість при роботі з ними. Усі, хто у своїй роботі використовували хоч один із різновидів методів, стикався з потребою їх оптимізації. Це загальновідома проблема і тому наразі існує багато рішень, які можуть допомогти із пришвидшенням використання методів, і саме тому подібні рішення необхідно імплементувати у створювану систему.

Іншим недоліком методів машинного навчання є певна неточність у їх роботі. Загалом вона характерна у тих випадках, коли при роботі нехтується аспект навчання штучного інтелекту або, впливаючи з минулого пункту, йде акцент саме на швидкості роботи. При таких вхідних даних нерідко можна виявити, що кінцевий результат відрізняється від

бажаного. У розроблюваній системі потрібно зберігати баланс між швидкістю та точністю аналізу тексту.

Серед вищевказаних вимог варто зазначити загальну простоту у роботі з системою. Для цього потрібно розробити графічний інтерфейс, який допоможе користувачу використовувати доступну функціональність, не затрачуючи при цьому багато зусиль на розуміння того, як запустити ту чи іншу функцію. Розроблений графічний інтерфейс позбавить необхідності при роботі з системою використовувати сторонні засоби. Хорошою практикою програмування вважається відокремлення клієнтської та серверної частини програмного забезпечення. Це означає, що ці два аспекти системи по свої суті будуть двома різними об'єктами, які майже не будуть перетинатися один з одним. Цей підхід можна описати одним словом – інкапсуляція, що являє собою один з принципів об'єктно-орієнтованого програмування та є однією з вимог до розроблюваної системи.

Наявні методи машинного навчання

До найбільш розповсюджених методів машинного навчання можна віднести навчання на основі дерев рішень; навчання на основі асоціативних правил; штучні нейронні мережі; глибинне навчання; індуктивне логічне програмування; метод опорних векторів; використання Баєсової мережі.

Принцип дії усіх цих методів можна звести до простого оброблення вхідної інформації за допомогою певних алгоритмів. Важливою частиною усіх методів машинного навчання є механізм самого «навчання». Методи можуть відрізнятися способом їх імплементації, але це не змінює їх суті, при кожному аналізі даних система вносить в себе певні корективи, які в подальшому можуть пришвидшити процес отримання результату або зробити його більш точним. Крім алгоритмів аналізу та навчання можна виділити методи за типом даних, які вони приймають на вхід.

Метод навчання на основі дерев рішень передбачає отримання результату з вхідного набору даних. Від кожного вхідного параметру йде розгалуження по всім його можливим результатам. Від всіх розгалужень йдуть свої змінні до кінцевого елемента («листка» дерева). Цих листків зазвичай дуже багато і поступово, навчаючись, можуть з'являтися нові листки або шлях до старих може бути оптимізованим. В результаті маємо структуру, дуже схожу на реальне дерево з гілками та листям. Математично дерево рішень можна відобразити за допомогою формули:

$$(x, Y) = (x_1, x_2, x_3, \dots, x_k, Y) \quad (1)$$

Змінну Y представляє значення, яке ми намагаємось отримати, використовуючи дерево рішень. Вектор x складається з інших векторів x_1, x_2, x_3 , та інших векторів, або вузлів, які використовуються в задачі.

Навчання асоціативних правил – один з методів машинного навчання, основна ідея якого базується на знаходженні зв'язків між даними у великих їх наборах. Математична задача методу асоціативних правил полягає в наступному:

$$I = \{i_1, i_2, \dots, i_n\} \quad (2)$$

змінна I являє собою множину n двійкових атрибутів, крім неї є інша змінна D – база даних, множина операцій та транзакцій:

$$D = \{t_1, t_2, \dots, t_m\} \quad (3)$$

Кожна операція в D має унікальний номер ID та свою підмножину I . Правило визначається наступним чином:

$$X \Rightarrow Y, \text{ де } X, Y \subseteq I \quad (4)$$

Елементи X та Y – це дві множини, які представляють собою набори предметів.

Метод штучних нейронних мереж – один з найпоширеніших та найвідоміших методів машинного навчання. Він був створений як певна копія роботи біологічного мозку, який використовує нейронні зв'язки. Обчислення вхідної інформації відбувається в групах штучних нейронів, які, в свою чергу, використовують конективістський підхід обчислення. Такі системи ідеальні для обчислень при роботі з розрізненими наборами даних та при необхідності знаходження закономірностей. В математиці нейронну модель можна зобразити наступною функцією:

$$f: X \rightarrow Y \quad (5)$$

де нейромерева функція f являє собою поєднання інших функцій, які, в свою чергу, можуть бути розділені на свої функції. Суть їх використання полягає у тому, що перш ніж з вхідних даних вийде результат, ці дані пройдуть через множину обчислювальних функцій, аж поки не буде дана остаточна відповідь.

Запропонований метод

Для розроблення програмного забезпечення для підбору вакансій був проведений аналіз серед деяких методів машинного навчання з метою виявлення найбільш вдалого для імплементації і подальшого вдосконалення розробки. Під час даного дослідження було проаналізовано:

- метод опорних векторів;

- Бассові мережі;
- навчання на основі дерев рішень.

Ці методи були обрані, оскільки вони більше всього підходять для аналізу текстових даних і пошуку ключових слів у них.

Маючи перелік методів, варто також зазначити, за якими параметрами їх варто оцінювати. У даному дослідженні було використано 4 характеристики: TP (true positive), TN (true negative), FP (false positive), FN (false negative). На основі цих метрик можемо вивести наступні параметри, які будуть використовуватись для знаходження оптимального методу:

1) Чутливість (далі Sensitivity) – параметр, який показує, з яким відсотком система вірно виявляє позитивні результати:

$$Sensitivity = \frac{TP}{TP+FN} \quad (6)$$

2) Специфіка (далі Specificity) – параметр, який показує, з яким відсотком система вірно виявляє негативні результати:

$$Specificity = \frac{TN}{TN+FP} \quad (7)$$

В ідеальних умовах обидва ці параметри мають бути наближені до одиниці (або 100%), але подібного результату досягти неможливо, тому необхідно зберігати баланс між ними.

Крім того, існують інші параметри: точність (Accuracy) та влучність (Precision).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (8)$$

$$Precision = \frac{TP}{TP+FP} \quad (9)$$

Інколи при аналізі також їх враховують, але головними показниками все ж являються Sensitivity та Specificity.

Для аналізу вищезгаданих методів машинного навчання була використана підготовлена база текстових публікацій. За отриманими результатами дослідження найбільш оптимальним виявився метод опорних векторів. При роботі з п'ятьма наборами даних він продемонстрував середнє значення параметру Sensitivity близько 93% і Specificity близько 72% (табл.1).

Таблиця 1

Порівняння показників Sensitivity та Specificity для різних наборів даних.

Номер набору даних	Sensitivity, %	Specificity, %
1	94	59
2	89	81
3	91	80
4	96	67
5	95	73

Для подальшого удосконалення методу опорних векторів варто ознайомитися з такими поняттями, як пониження дискретизації (Downsampling), збільшення ваги (Upweighting) та генерація ознак (Feature generation), за допомогою яких планується оптимізувати існуючий метод.

Downsampling та Upweighting – це підходи, які застосовуються в процесі навчання моделі, основною їх суттю є маніпуляції з набором вхідних даних для навчання. Набори даних можна розділити на класи позитивних та негативних результатів.

Дуже часто складається ситуація, коли на один випадок позитивного результату припадає близько 200 негативних. Така незбалансованість негативно впливає на результат навчання і тут в силу вступають зазначені підходи. Downsampling передбачає навчання на наборі даних з заниженим значенням переважаючого класу, а Upweighting – навпаки, у наборі зі збільшеним значенням класу з малим представленням.

Feature generation передбачає вибірку найбільш релевантних ознак даних у наборі. Часто дослідники нехтують цим підходом, оскільки вважається, що модель сама визначить важливі ознаки між даними для подальшого отримання результату, але генерація нових ознак може зменшити час роботи системи та її загальну складність.

Поєднання зазначених підходів та їх коректне налаштування повинно позитивно відобразитися на значеннях Sensitivity та Specificity і в результаті збільшити ці показники на 5%.

Висновки

Під час дослідження було проаналізовано наявні методи машинного навчання та способи їх аналізу для задачі пошуку вакансій.

Були виявлені основні характеристики, на які варто опиратися при розрахунку оптимальності тієї чи іншої моделі. До найбільш важливих параметрів відносяться: чутливість, влучність, час виконання.

При аналізі наявних методів машинного навчання було виявлено оптимальний метод, а саме метод опорних векторів. Були отримані значення його базових параметрів Sensitivity та Specificity.

Проаналізовані можливості для подальшого вдосконалення базового методу шляхом використання підходів Downsampling, Upweighting та Feature generation, за допомогою яких можливе збільшення значень Sensitivity та Specificity у середньому на 5% та скорочення часу виконання моделі на 5-10%.

Література

1. Drew Conway and John Myles White “Machine Learning for Hackers” (2012).
2. Machine Learning course – Режим доступу: <https://developers.google.com/machine-learning>
3. E. Popoff, M. Besada, J. P. Jansen, S. Cope & S. Kanters “Aligning text mining and machine learning algorithms with best practices for study selection in systematic literature reviews” – Режим доступу: <https://systematicreviewsjournal.biomedcentral.com/articles/10.1186/s13643-020-01520-5#Sec8>.
4. What is Text Mining, Text Analytics and Natural Language Processing? – Режим доступу: <https://www.linguamatics.com/what-text-mining-text-analytics-and-natural-language-processing>.
5. К.Р. Soman, R. Loganathan, V. Ajay “Machine Learning with SVM and Other Kernel Methods” (2009).

К.т.н., асистент Погорелов В.В., магістрант Труш А.О.

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

КОМБІНОВАНИЙ МЕТОД РОЗПІЗНАВАННЯ REDOS ВРАЗЛИВОСТЕЙ В РЕГУЛЯРНИХ ВИРАЗАХ

Abstract

Volodymyr Pogorelov, PhD; Artem Trush, student

A combined approach for detection ReDoS vulnerabilities

Regexes are widely used in almost each big project on Github, so ReDoS poses a serious security threat. Early detection of ReDoS-vulnerable regexes in software is vital. Automatic tools for detecting ReDoS vulnerabilities are in great demand. Existing approaches has a few disadvantages: existing of the false positive results and requiring too much time to check. In this work methods related to ReDoS detection are analyzed and the approach to their combination, that solves above disadvantages, based on a few latest static methods is proposed.

Вступ

Регулярні вирази є незамінним інструментом щоб перевірити чи відповідає введений набір символів конкретному шаблону. Сучасні мови програмування підтримують регулярні вирази на рівні стандартних бібліотек і тим самим заохочують розробників використовувати всі їхні переваги. Проте, завжди існує ризик, створення таких алгоритмів, що обчислення співпадінь матиме в гіршому випадку експоненціальну часову складність. Зловмисник може створити таку строку, щоб змусити обчислювальний пристрій обійти всі можливі співпадиння за експоненціальний час і тим самим здійснити ReDoS атаки.

ReDoS є досить недооціненими атаками на відмову в обслуговуванні (DoS). В нещодавньому дослідженні щодо використання регулярних виразів в більше 3000 Python проєктів на Github було виявлено, що 43% серед них містять регулярні вирази [1]. Інше дослідження [2] показало, що близько 10% Node.js веб-застосунків виявились вразливими до ReDoS атак. Також було здійснене дослідження [3], яке виявило, що менше половини розробників знають про існування та методи захисту від ReDoS атак. Багато великих компаній відчули вплив таких атак, серед них: Stack

Overflow, Cloudflare, iCloud. Виявити вразливість вручну є складним процесом, тому є необхідність у спрощенні та автоматизації цього процесу.

Постановка задачі

Основною метою даного дослідження є підвищення ефективності процесу розпізнавання ReDoS вразливостей в коді, що містить регулярні вирази шляхом створення покращеного комбінованого методу в якого точність розпізнавання буде більше, а час перевірки буде менше ніж в аналогів.

Термінологія

Регулярний вираз – це рядок, що описує або збігається з множиною рядків, відповідно до набору спеціальних синтаксичних правил. Кожен регулярний вираз можна представити у вигляді визначеного автомата.

Regular Expression Denial of Service (ReDoS) – це атака на складність алгоритмів, яка створює відмову в обслуговуванні, спричинена введенням шкідливого набору символів в місця де використовуються регулярні вирази. В найгіршому випадку час може зростати експоненціально по відношенню до розміру вхідних даних.

Аналіз існуючих способів

Існує два головних підходи для визначення ReDoS:

1. Статичне розпізнавання. Існує багато різних підходів до семантично-базованого статичного розпізнавання ReDoS вразливостей, але більшість з них використовує автомати. В цих фреймворках регулярні вирази спершу трансформуються в автомат, котрий потім аналізує і визначають чи вразливі вирази чи ні. Головна проблема в тому, що трансформування регулярного виразу в автомат може як приховати, так і додати вразливостей. Це зазвичай є джерелом FP та FN результатів. Але, в останніх роботах на цю тему автори почали уникати побудов автоматів і натомість використовують підходи з використанням відомих шаблонів вразливостей в регулярних виразах та базовані на семантичних деревах.
2. Динамічне розпізнавання. Під час динамічного підходу відбувається генерація набору символів, яку потім тестують на регулярному виразі. Під час цього заміряють час за який відбувається пошук співпадінь і з огляду на це визначають чи є вразливість. Такі застосунки є відчутно повільнішими за застосунки зі статичними аналізаторами так, як тестування експоненціальних алгоритмів потребує багато часу.

Опис алгоритму

Загалом запропонований алгоритм для пошуку вразливостей в регулярних виразах складається з наступних кроків:

1. Стандартизація регулярного виразу.
2. Виконання статистичного аналізу наступними методами:
 - 2.1. Метод базований на семантичному дереві
 - 2.2. Метод базований на відомих шаблонах з CVE.
3. Виконання динамічного аналізу – для впевненості в тому, що в регулярному виразі дійсно наявна вразливість.

Комбінований метод полягає в поєднанні двох статистичних методів розпізнавання задля найкращої швидкодії та виокремленні шаблону з допомогою якого відбувається конструювання шкідливого тексту з динамічною перевіркою регулярного виразу.

Дане рішення:

- уникає трансформації регулярних виразів до автомата, щоб уникнути нових вразливостей і не пропустити наявних
- підтримує розширені регулярні вирази
- доповнює метод базований на семантичному дереві, так як він розрахований лише на експоненціальні випадки
- покращує та пришвидшує метод базований на шаблонах у експоненціальних випадках
- завдяки динамічній перевірці уникає False Positive.

Стандартизація

Регулярні вирази підтримують декілька корисних розширень. Серед них: `lazy quantifiers`, `anchors`, `lookarounds`, `backreference`. В [1] автори виявили, що у майже 4000 проєктах Python 4% регулярних виразів використовують `lookarounds` та до 0,4% використовують `backreference`.

Наявність розширень в регулярних виразах ускладнює процес аналізу. Тому, для вирішення даної проблеми було застосовано підхід з [4], перетворюємо даний регулярний вираз у стандартний регулярний вираз із деякими обмеженнями. Мета полягає не в тому, щоб дати еквівалентне перетворення, а натомість у спробі здійснити перетворення зі збереженням ReDoS вразливостей. Тобто, щоб вихідний регулярний вираз α мав ReDoS вразливості і трансформований цільовий вираз β мав ті ж самі ReDoS-уразливості. Якщо «той самий ефект» важко досягти, ми шукаємо послаблену умову, яка дозволяє цільовому виразу генерувати більше ReDoS вразливостей, ніж вихідний регулярний вираз.

Метод базований на семантичному дереві

Вперше цей метод був запропонований [5] і реалізований в застосунку Rat. Він зарекомендував себе, як один з найкращих та найточніших для визначення експоненціальних вразливостей. Хоча цей метод продемонстрував чудові результати розпізнавання та майже за всіма показниками є кращим за попередні статичні методи, він має ряд недоліків.

Одним з недоліків є те, що не передбачена підтримка розширень. Таких, як lookarounds, backreferences. Автори схилились до думки, що розробники їх використовують зрідка і вирішили залишити цю частину для майбутньої роботи.

Іншим недоліком є те, що виявляються лише вразливості експоненціальної часової складності.

Також даний метод з певною ймовірністю може визначити, що не вразливий регулярний вираз є вразливим (FP).

Метод базований на відомих шаблонах

В роботі [4] було виокремлено 5 загальновідомих шаблонів, на які були відкриті CVE. Цими шаблонами є:

1. Nested Quantifiers (NQ) – CVE-2015-9239.
2. Exponential Overlapping Disjunction (EOD) – CVE-2020-7662.
3. Exponential Overlapping Adjacency (EOA) – CVE-2018-3738.
4. Polynomial Overlapping Adjacency (POA) – CVE-2018-3737.
5. Starting with Large Quantifier (SLQ) – CVE-2019-1010266.

Стандартизований регулярний вираз перевіряється реалізованими алгоритмами на кожен з цих шаблонів.

Динамічна валідація

В процесі перевірки статичними методами наш алгоритм буде шаблон шкідливого набору символів (attack string):

1. Знаходження ядра attack string – s1.
2. Знаходження префікса attack string – s2.
3. Знаходження суфікса attack string – s3.
4. Утворення attack string – s, що будується за принципом $s2+s1^n+s3$.

Кроки динамічної валідації є наступними: виміряти час для вихідного регулярного виразу за який він надасть результат для attack string, перевірити чи досягнутий відповідний поріг для поліноміальної чи експоненціальної вразливості, якщо так то повернути true, якщо ні повернути false.

Висновки

Таким чином, у даній роботі запропоновано підхід в якому відбувається комбінування двох статичних методів, що доповнюють одне одного та динамічна перевірка правильності розпізнавання. Цими методами є: метод, що розпізнає вразливості на основі відомих шаблонів та метод, в якому на основі регулярного виразу будуємо семантичне дерево. Попередньо відбувається стандартизація для охоплення більшої кількості регулярних виразів.

Тестування було проведене на наборі з понад 70000 вразливих регулярних виразів. Метод продемонстрував якісне розпізнавання не зважаючи на мову програмування, що змінює обчислювальний двигун. Точність розпізнавання склала 98%, що краще за існуючі статичні методи, в той час, як, швидкість розпізнавання перевершила очікування і склала в середньому 1 секунду на регулярний вираз, що відповідно, краще за існуючі динамічні методи.

Перспективним напрямком для продовження роботи над тематикою даної статті автори вважають дослідження нових шаблонів вразливостей та покращення алгоритму генерації набору символів для тестування вразливості, так як створений алгоритм має близько 70% точності.

Література

1. Chapman, C., Stolee, K.T.: Exploring regular expression usage and context in Python. In: International Symposium on Software Testing and Analysis, ISSTA. pp. 282-293. ACM (2016)
2. Staicu, C.A., Pradel, M.: Freezing the web: A study of redos vulnerabilities in javascript-based web servers. In: 27th USENIX Security Symposium, USENIX Security 2018, Baltimore, MD, USA, August 15-17, 2018. pp. 361-376 (2018)
3. IV, L.G.M., Donohue, J., Davis, J.C., Lee, D., Servant, F.: Regexes are hard: Decision-making, difficulties, and risks in programming regular expressions. In: International Conference on Automated Software Engineering, ASE. pp. 415-426. IEEE (2019).
4. Yeting Li, Zixuan Chen, Jialun Cao, Zhiwu Xu, Qiancheng Peng and Shing-Chi Cheung. ReDoSHunter: A Combined Static and Dynamic Approach for Regular Expression DoS Detection. In Proceedings of the 30th USENIX Security Symposium, Aug. 2021
5. Francesco Parolini, Antoine Miné. Sound Static Analysis of Regular Expressions for Vulnerabilities to Denial of Service Attacks. 2022. hal-03685057

К.т.н., доцент Рибачок Н.А., студент Манохін А.В.

**Національний технічний університет України
«Київський політехнічний інститут»**

МЕТОДИ РОЗПІЗНАВАННЯ АНГЛОМОВНИХ АКЦЕНТІВ

Abstract

Andrii Manokhin, student; Nataliia Rybachok, PhD

Methods of recognition of English accents

In the process of speech recognition, several modern popular methods are used to extract important details from audio, resulting in a mathematical representation of sound. However, upon closer examination of each of them, certain shortcomings can be identified. Thus, it is logical to develop a combined method.

Вступ

В рамках попереднього дослідження було розроблено програмне забезпечення, яке визначає відсоток належності аудіозапису до 8 найбільш розповсюджених англомовних акцентів. Згорткова нейронна мережа, що складається із 2 згорткових шарів, 1 шару max pooling, а також 2 щільних шарів була натренована протягом 2 епох на множині 5 516 аудіозаписів, взятих із ресурсу English Multi-speaker Corpus for Voice Cloning. Досягнуто точність прогнозування 89.07% на тестових даних, що представлялися 11 тис. матрицями MFCC розмірністю 50x87 [1].

Однак, отримані результати спонукали до знаходження методів покращення розпізнавання раніше невідомих мережі користувачів, оскільки характеристики деяких акцентів є досить схожими навіть для людського вуха. Для вирішення цього завдання необхідно додаткове дослідження щодо кроку передобробки аудіо, а саме, видобування корисних для мережі його характеристик.

Аналіз доступних робіт за цією тематикою показав, що на даний момент існують кілька популярних методів видобування характеристик із аудіо, а саме MFCCs, LPC та DWT.

Постановка задачі

Задача, результати вирішення якої представлено у цій статті, полягала в аналізі існуючих методів видобування особливостей із аудіозапису та розробленні комбінованого програмного методу, який має підвищити точність визначення англомовного акценту при подальшому аналізі даних, не змінюючи модель визначення.

Термінологія

Формант — термін фонетики, що позначає акустичну характеристику звуку слова.

Вейвлет — це хвиля фактично обмеженої тривалості, середнє значення якої дорівнює нулю.

Банк фільтрів — масив смугових фільтрів, який розділяє вхідний сигнал на кілька компонентів.

Мел — психофізична одиниця висоти звуку.

Аналіз існуючих методів

MFCC (Mel frequency cepstral coefficients)

Частотні кепстральні коефіцієнти Mel (MFCC) спочатку були запропоновані для ідентифікації односкладових слів у безперервних промовах, але не для ідентифікації мовця. Обчислення MFCC є реплікацією системи слуху людини, яка має намір штучно реалізувати принцип роботи вуха з припущенням, що людське вухо є надійним розпізнавачем мовця. Функції MFCC ґрунтуються на визнаній невідповідності критичних смуг пропускання людського вуха з частотними фільтрами, розташованими лінійно на низьких частотах і логарифмічно на високих частотах, які використовуються для збереження фонетично важливих властивостей мовного сигналу. Мовні сигнали зазвичай містять тони різних частот, кожен тон має фактичну частоту f (Гц), а суб'єктивна висота тону обчислюється за шкалою Мела. Мел-частотна шкала має лінійний інтервал частот нижче 1000 Гц і логарифмічний інтервал вище 1000 Гц. Висота тону 1 кГц і 40 дБ вище перцептивного чутного порогу визначається як 1000 мел і використовується як контрольна точка.

MFCC заснований на дезінтеграції сигналу за допомогою банку фільтрів. MFCC дає дискретне косинусне перетворення (DCT) дійсного логарифма короткочасної енергії, що відображається на частотній шкалі Mel. MFCC використовується для ідентифікації бронювання авіа квитків, номерів, промовлених у телефон, і системи розпізнавання голосу з метою безпеки.

Кепстральні коефіцієнти вважаються точними в певних проблемах розпізнавання образів, пов'язаних із людським голосом. Вони широко використовуються для ідентифікації мовця та розпізнавання мовлення.

Інші форманти також можуть бути вищими за 1 кГц і не враховуються ефективно через великий інтервал фільтрів у високочастотному діапазоні

LPC (Linear Predictive Coding)

Коефіцієнти лінійного передбачення імітують людський голосовий тракт і забезпечують надійну функцію мовлення. Даний метод оцінює мовний сигнал шляхом апроксимації формант, позбавляючись від їх впливу на мовний сигнал і оцінюючи концентрацію та частоту залишку, що залишився позаду. У результаті кожен зразок сигналу є прямим включенням попередніх зразків. Коефіцієнти різницевого рівняння характеризують форманти, тому LPC потребує апроксимації цих коефіцієнтів. LPC є потужним методом аналізу мовлення, і він отримав популярність як метод оцінки формант.

Частоти, на яких виникають резонансні гребені, називаються формантними частотами. Таким чином, за допомогою цієї методики позиції формант у мовному сигналі можна передбачити шляхом обчислення лінійних коефіцієнтів прогнозування над ковзним вікном і знаходження вершин у спектрі наступного фільтра лінійного прогнозування. LPC допомагає кодувати високоякісну мову з низькою швидкістю передачі даних.

Характеристики LPC:

- ефективно вибирає інформацію голосового тракту з заданої мови;
- відомий швидкістю обчислень і точністю;
- представляє поведінку джерела, яка є стабільною та послідовною;
- можна використовувати в системах розпізнавання мовців, де основною метою є вилучення властивостей голосового тракту;
- дає точні оцінки мовних параметрів і є порівняно ефективним для обчислень;
- страждає від псевдокореляційних коефіцієнтів;
- оцінки LPC мають високу чутливість до шуму квантування і можуть не підходити для узагальнення.

WT (Wavelet Transform)

Теорія вейвлетного перетворення зосереджена навколо аналізу сигналу з використанням різних масштабів у часовій і частотній областях. Багато вейвлетів також демонструють ортогональність, ідеальну властивість компактного представлення сигналу. WT — це метод обробки сигналів, який можна використовувати для представлення реальних нестационарних сигналів з високою ефективністю. Він має можливість отримувати інформацію з перехідних сигналів одночасно як у часовій, так і в частотній областях.

Дискретне вейвлет-перетворення є розширенням WT, яке підвищує гнучкість процесу декомпозиції. Він був представлений як дуже гнучкий і ефективний метод розбиття сигналів на піддіапазон.

Параметри DWT містять інформацію різних частотних масштабів. Це покращує мовну інформацію, отриману у відповідному діапазоні частот. Здатність DWT розподіляти дисперсію елементів вхідних даних на шкалу за шкалою є додатковою перевагою. Це розбиття призводить до думки про залежну від масштабу вейвлет-дисперсію, яка багато в чому еквівалентна більш знайомому частотно-залежному спектру потужності Фур'є. Класичні схеми дискретної декомпозиції, які є діадицичними, не відповідають усім вимогам для прямого використання в параметризації. DWT дійсно забезпечує достатню кількість діапазонів частот для ефективного аналізу мови. Оскільки вхідні сигнали мають кінцеву довжину, вейвлет-коефіцієнти матимуть небажано великі варіації на границях через розриви на границях.

Результати порівняння методів

Кожен із вище названих методів має як недоліки, так і переваги. MFCC було використано в процесі написання системи розпізнавання англійських акцентів через його точність та популярність. Хоча метод точний в оцінці саме людського голосу, має посередній рівень шумостійкості, а також загалом погано узагальнюється, що унеможлиблює подальше його використання. В свою чергу лінійні коефіцієнти передбачення (LPC) мають високу чутливість до шуму квантування і можуть також не підходити для узагальнення, хоча і є дуже ефективним в обчисленні. Здатність дискретного вейвлет-перетворення (DWT) розподіляти дисперсію елементів вхідних даних на шкалу за шкалою є додатковою його перевагою. Оскільки вхідні сигнали мають кінцеву довжину, вейвлет-коефіцієнти матимуть небажано великі зміни на границях через розриви на границях.

Запропонований комбінований метод

Враховуючи недоліки та переваги вище вказаних методів, було прийнято рішення відмовитися від використання MFCC та поєднати два інші методи – LPC та DWT (рис. 1).

Метод лінійного прогнозування використовується для отримання коефіцієнтів фільтра, еквівалентних голосовому тракту, шляхом зменшення середньоквадратичної помилки між вхідним мовленням та оціненим мовленням:

$$\hat{s}(n) = \sum_{k=1}^p a_k s(n - k)$$

де \hat{s} – прогнозований зразок, s – вибірка мовлення, p – коефіцієнти предиктора.

Отримані коефіцієнти LP не можуть вловити високочастотні піки, присутні в мовному сигналі, а також не можуть точно проаналізувати локалізовані події, які може аналізувати вейвлет-перетворення (DWT). Однак LPC може краще розрізняти слова, які мають різні голосні звуки, ніж ті, що мають спільні голосні. DWT здатний моделювати деталі глухої звукової частини мови, на відміну від LPC. Отже, ми можемо застосувати техніку LPC до кожного піддіапазонного сигналу після вейвлет-декомпозиції, що дає комбіновані переваги LPC та WT.

Декомпозиція DWT попередньо оброблених і віконних кадрів мовлення виконується за допомогою вейвлет-фільтрів. Фактичні вейвлет-коефіцієнти зберігають інформацію про час, отже, характеристики LPC оцінюються за коефіцієнтами DWT у часовій області. Характеристики LPC порядку p були виділені з кожного піддіапазону вейвлет-декомпонованого мовного сигналу.

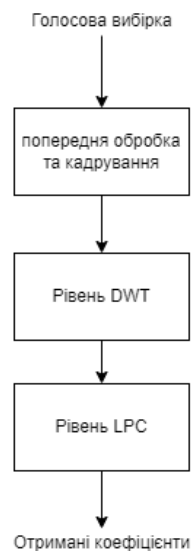


Рис. 1. Схема поєднання двох алгоритмів в єдиний

Висновки

У роботі описано комбінований метод видобування характеристик із аудіо метою якого є зменшення так званого шуму в аудіо. В подальшому заплановано дослідження ефективності запропонованого методу на тестовій системі.

Література

1. Manokhin A.V., Rybachok N.A. English Accent Recognition Using Deep

- Machine Learning. Управляющие системы и машины. 2021. № 4. p. 52-59. <https://doi.org/10.15407/csc.2021.04.028>
2. Linear predictive coding [Электронный ресурс]. Режим доступа: https://en.wikipedia.org/wiki/Linear_predictive_coding
 3. MFCC Technique for Speech Recognition. [Электронный ресурс]. Режим доступа: <https://www.analyticsvidhya.com/blog/2021/06/mfcc-technique-for-speech-recognition/>
 4. Distance measures for speech recognition, psychological and instrumental. [Электронный ресурс]. Режим доступа: <https://www.semanticscholar.org/paper/Distance-measures-for-speech-recognition%2C-and-Mermelstein/a086b8bc7a3ef2eee498ada4481c33a5e43fcf>
 5. Audio Feature Extraction – Devopedia. [Электронный ресурс]. Режим доступа: <https://devopedia.org/audio-feature-extraction>

К.т.н., доцент Рибачок Н.А., студент Тютюнник П.Б.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

МЕТОД ВИЗНАЧЕННЯ МОЖЛИВИХ РИЗИКІВ ТА ЗАТРИМОК В AGILE ПРОЕКТАХ

Abstract

Nataliia A. Rybachok, assoc. prof., PhD; Tiutiunnyk Petro, student
Identifying possible risks and delays in AGILE projects

The task of identifying possible risks and delays is important for AGILE projects, because risks and delays not found in time are the most frequent causes of failed projects. This work presents the results of a research aimed at identifying risks and delays in AGILE projects.

To solve this problem, it is proposed to use the Random Forest machine learning method and the Prophet time series data forecasting procedure. The developed method is obtained by entering a coefficient based on a model trained by the Prophet method using time data at the voting stage of the Random Forest method.

Вступ

Сьогодні одним із найпопулярніших методів управління проектами програмного забезпечення є методологія AGILE, яку використовують до 95% технологічних компаній і 86% команд розробників програмного забезпечення [1], оскільки це одна з найефективніших методологій управління проектами. При цьому AGILE-проекти часто провалюються через ризики, які не були виявлені на етапі планування, і великі затримки. Існуючі методи виявлення ризиків і затримок не є автоматизованими і, отже, більш схильні до людських помилок. Часто проектні команди не можуть вчасно визначити необхідні ризики, що призводить до провалу проєктів.

Для вирішення цієї проблеми можна використовувати методи машинного навчання, оскільки це проблема бінарної класифікації, де необхідно з'ясувати, чи належить проблема AGILE проекту до набору можливих ризиків чи ні. При цьому пропонується використовувати історичні дані, такі як невиконані проєкти JIRA з відкритим кодом.

Аналіз доступних робіт за цією тематикою показав, що на даний момент існує досить багато методів виявлення можливих ризиків і

затримок в проектах AGILE, які використовують методи машинного навчання. Наприклад, наївний Баєсів класифікатор, штучні нейронні мережі, Random Forest та інші [2]. Вони демонструють високу точність передбачень. Найефективнішим методом класифікації ризиків проблем AGILE наразі є Random Forest із найвищими результатами продуктивності та середньою точністю 0,8 [2].

Основною проблемою вказаних вище методів є неможливість використання функцій на основі дат, що може вплинути на статус ризику проблеми AGILE. Для вирішення цієї проблеми можна використати процедуру Prophet [3].

Постановка задачі

Задача полягає в розробці програмного методу ідентифікації ризиків і затримок у проектах AGILE, який має підвищити точність визначення можливого фактора затримки певного завдання в проекті AGILE за рахунок включення особливостей проблем, які пов'язані із датою.

Термінологія

AGILE – це клас методологій розробки програмного забезпечення, що засновані на ітераційній розробці, у якій вимоги та рішення розвиваються завдяки співпраці між багатофункціональними командами, здатними до самоорганізації.

Ризик – це невизначена подія або умова, яка, якщо вона відбудеться, позитивно або негативно впливає на цілі проекту.

Класифікація – це проблема визначення, до якої з набору категорій (наборів) належить спостереження.

Опис алгоритму

Random Forest

Random Forest прогнозує значення відповіді шляхом усереднення прогнозованих значень відповіді по багатьох деревах рішень [4]. Кожне дерево вирощується на початковому зразку навчальних даних. Завантажувальна вибірка – це випадкова вибірка спостережень, зібрана із заміною. Крім того, предиктори відбираються під час кожного розбиття в дереві рішень. Дерево рішень підбирається за допомогою методології рекурсивного розбиття. Процес підгонки навчального набору відбувається наступним чином:

- для кожного дерева обирається початкова вибірка спостережень;
- встановлюється індивідуальне дерево рішень за допомогою рекурсивного розбиття таким чином:

- вибирається випадковий набір предикторів для кожного розбиття;
- розділення продовжується, доки не буде виконано правило зупинки, вказане у вікні специфікації випадкового лісу;
- кроки 1 і 2 повторюються, доки не буде досягнуто кількості дерев, указаних у вікні специфікації Random Forrest, або доки не відбудеться рання зупинка.

Отримані дерева рішень потім використовуються для класифікації даних, де результатом випадкового лісу є клас, що обраний більшістю дерев.

Як було зазначено раніше, Random Forrest показав найкращі результати для результатів класифікації ризиків проблем AGILE, таких як точність і продуктивність. Але цей алгоритм не включає функції, що пов'язані з датою, які можуть впливати на ймовірність затримки проблеми, наприклад час, коли проблема була подана, або над нею розпочато роботу. Щоб включити функції, пов'язані з датою, можна використати процедуру Prophet і розрахувати коефіцієнт, який вплине на кінцевий вихід алгоритму Random Forest.

Prophet

Prophet або «Facebook Prophet» – процедура прогнозування часових рядів на основі адитивної моделі, призначена для обробки генеральних рядів, у яких зручно коригуються нелінійні часові параметри [3].

Має такі характеристики:

- найкраще працює з часовими рядами, які мають сильний сезонний вплив і кілька періодів історичних даних;
- стійкий до пропущених точок і змін трендів і добре справляється з відхиленнями;
- підтримує тренди, сезонність і свята;
- відповідна бібліотека надає інструменти для автоматичного скорингу та побудови графіків;
- прогноз можна робити автоматично, а потім коригувати вручну;
- результат прогнозування також можна покращити за допомогою інтерпретованих параметрів (знання предметної області та період дослідження).

Для навчання двох вказаних методів машинного навчання потрібні історичні дані. Для створення навчальних наборів даних дані про проблеми AGILE було зібрано з 4 проєктів програмного забезпечення з відкритим кодом, які використовують Atlassian JIRA як систему відстеження проблем: Apache, Red Hat, Spring і Moodle. Загалом із систем відстеження проблем було зібрано близько 1,5 мільйона питань.

Лише невелику частину з них можна було використати для навчання методу через відсутність необхідних даних для визначення статусу затримки цих питань.

Отриманий набір даних складається з 50 тисяч попередньо оброблених питань. Після попередньої обробки вони включають наступні характеристики: час обговорення, час очікування, тип проблеми, кількість повторів, відсоток відкладених проблем, які виникли у розробника, робоче навантаження розробника, пріоритет, кількість коментарів, кількість змін пріоритету, кількість версій виправлень, кількість уражених версій, кількість посилань на проблеми, кількість проблем із блокуванням тощо. Набори даних використовувалися для навчання моделі методу Random Forest, а функція простого режиму використовувалася для визначення результату набору дерева рішень, яке показувало наступні точні результати для кожного проекту:

- 0.8 для Apache;
- 0.73 для Moodle;
- 0,96 для Red Hat;
- 0,83 для Spring;
- 0,84 для всіх проектів разом.

Отримані результати високі, їх можна вважати корисними для AGILE, але вони мають великий розкид від 73% до 96%, що може вказувати на переобладнання для проекту Red Hat. Крім того, ця модель не включає дані, пов'язані з датою.

Щоб включити дані, пов'язані з датою, модель Prophet була навчена передбачати ймовірність того, яка проблема стане ризикованою, якщо робота починається в дату. Середня ймовірність того, що випуск буде відкладено, становить приблизно 21%, але для випусків, які почалися наприкінці кварталів або під час святкових сезонів, такий шанс вищий, ніж середній. Таким чином, можна припустити, що використання коефіцієнта, який дорівнює відсотковій різниці між середнім шансом і прогнозованим шансом ризику моделі Prophet, для зміни результатів розподілу в наборах дерева рішень.

Використовуючи набори даних, було навчено моделі Prophet та Random Forest і використано нову функцію з використанням коефіцієнта для визначення результату набору дерева рішень, яке показало наступні точні результати для кожного проекту:

- 0.82 для Apache;
- 0.74 для Moodle;
- 0,93 для Red Hat;
- 0,84 для Spring;

- 0,85 для всіх проектів разом.

Отримані результати мають точність у середньому на 1-2% вищу, ніж раніше, і нижчу точність для проекту Red Hat, що може вказувати на нижчу переобладнаність моделі для цього проекту.

Висновки

Виявлення ризиків і затримок в проектах AGILE є дуже важливим завданням, оскільки це основна причина невдач проектів. Запропонований метод класифікатора для ризиків і затримок AGILE показує результат високої точності для проектів з відкритим кодом із середньою точністю 0,85, що можна вважати корисним для управління ризиками в проектах AGILE та показує вищі результати в середньому на 1-2% на відміну від не модифікованого методу Random Forrest.

Тим не менш, запропонований метод також демонструє великий розкид результатів для різних проектів, з відмінностями до 20%, що вказує на переобладнання для деяких випадків або великі відмінності між кожним проектом AGILE. Для більш високої точності результатів може знадобитися більше навчальних і тестових даних, а також додаткові модифікації для врахування відмінностей між різними проектами AGILE.

Література

1. State of AGILE report [Електронний ресурс]. – Режим доступу: <https://stateofagile.com/>.
2. Choetkiertikul, M., Dam, H.K., Tran, T. *et al.* Predicting the delay of issues with due dates in software projects // *Empirical Software Engineering*. Vol. 22. 2017. - P. 1223–1263. <https://doi.org/10.1007/s10664-016-9496-7>.
3. Taylor SJ, Letham B. Forecasting at scale [Електронний ресурс]. – Режим доступу: <https://doi.org/10.7287/peerj.preprints.3190v2>.
4. Breiman L. *Random Forests* // *Machine Learning*. Vol. 45. 2001. P. 5–32. [doi:10.1023/A:1010933404324](https://doi.org/10.1023/A:1010933404324).

УДК 004.912

магістрант Чумак М.С., к.т.н., доцент Заболотня Т.М.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

МЕТОД ВИЯВЛЕННЯ НЕЧІТКИХ ДУБЛІКАТІВ ТЕКСТОВИХ ДОКУМЕНТІВ

Abstract

Tetiana Zabolotnia, assoc. prof., PhD; Mykhailo Chumak, student
Method for detecting fuzzy duplicates of text documents

This work is devoted to the method for detecting fuzzy duplicates of text documents. The detection of near-duplicate documents is an important problem and difficult task in many data mining and information filtering applications. When faced with massive quantities of data, traditional duplicate detection techniques relying on direct inter-document similarity computation are often not efficient enough. In this work, the existing methods for detecting fuzzy duplicates of text documents will be considered. As a result, a hypothesis for improving already existing methods will be proposed.

Вступ

З розвитком технологій обміну інформацією, зокрема мережі Інтернет, виникла та стрімко поширилась проблема незаконного запозичення текстових даних різних форматів. Звісно, використання копій дозволяється безкоштовно за умови, що вони не створюються та не розповсюджуються з метою отримання прибутку чи комерційної вигоди, і обов'язково містять посилання на першоджерело. Але численні порушення вищезазначених принципів стали причиною зростання необхідності розроблення методів та відповідних програмних засобів для виявлення нечітких дублікатів текстів. Вирішення цієї задачі є вкрай актуальним і для представників сучасного бізнесу.

З метою розв'язання задачі з виявлення нечітких дублікатів текстових документів розроблено багато різноманітних методів, що базуються на аналізі лексичних або синтаксичних ознак. Усі ці методи є наближеними, оскільки дозволяють уникнути повного попарного порівняння аналізованих документів і отримати прийнятні результати при допустимій обчислювальній складності. В цілому можна сказати, що існуючі методи дозволяють з достатньо високою точністю визначити, чи є два документи нечіткими дублікатами один одного. Проте цим рішенням досі притаманні проблеми помилкової ідентифікації близьких за змістом

документів як дублікатів, нестійкості методів до незначних змін в документах, прийнятих до аналізу, і не завжди високої точності оброблення документів різних розмірів.

Таким чином, дослідження та вдосконалення методів виявлення нечітких дублікатів є актуальним завданням, вирішення якого дозволить підвищити ефективність програмних систем з виявлення плагіату.

Постановка задачі

Задача даного дослідження полягає в модифікації існуючого частотного методу виявлення нечітких дублікатів текстових документів, з метою підвищення міри повноти на 2-3%. Частотні методи мають найвищі показники точності і посередні показники повноти, саме тому досягнення цілей в рамках поставленої задачі дасть змогу підвищити ефективність роботи існуючих методів.

Гіпотеза – розбиття текстових документів на більшу кількість блоків забезпечить виконання поставленої задачі.

Термінологія

Нечіткий дублікат документа – це документ, частково змінений у змістовній частині і / або в частині форматування [1].

F-міра – це одна з мір точності тесту. Її обчислюють через влучність та повноту тесту, де влучність є числом правильно визначених позитивних результатів, поділеним на число всіх позитивних результатів тесту, включно з визначеними неправильно, а повнота є числом правильно визначених позитивних результатів, поділеним на число всіх зразків, які мали бути визначені як позитивні [2].

Детектор дублікатів – це інструмент, який використовується для порівняння будь-яких двох веб-сторінок для виявлення тексту, який було скопійовано з однієї на іншу [3].

Хешування – перетворення вхідного масиву даних довільної довжини у вихідний бітовий рядок фіксованої довжини. Хеш-функція – функція, що перетворює вхідні дані будь-якого (як правило великого) розміру на дані фіксованого розміру [4].

IDF (англ. inverse document frequency – зворотна частота документа) – це інверсія частоти, з якою деяке слово зустрічається у документах колекції.

Стемінг (англ. stemming) - це процес скорочення слова до основи шляхом відкидання допоміжних частин, таких як закінчення чи суфікс.

Частотний словник — словник, у якому кожне слово характеризується певним числом, що вказує на кількість вживань цього слова в обстеженому масиві текстів, тобто на його абсолютну частоту появи в цих текстах. У багатьох словниках частота кожного слова в

обстежених текстах оцінюється статистично: обчислюється відносна похибка, коефіцієнт стабільності чи інші статистичні показники.

Текстовий документ – документ, який складається з текстових об'єктів (символів, слів, абзаців) та, можливо, інших об'єктів (графічних, мультимедійних тощо).

Аналіз існуючих методів виявлення нечітких дублікатів текстових документів

Методи виявлення нечітких дублікатів поділяються на такі категорії:

- 1) частотні методи;
- 2) методи, засновані на шинглюванні;
- 3) методи, засновані на використанні хешування;
- 4) сигнатурні методи.

До частотних відносяться такі методи: TF, TF*IDF, TF*RIDF. Відмінною від інших ознакою даного класу методів є розрахунок показника «ваги» кожного слова в прийнятих до аналізу документах. На основі цього показника складається словник, який містить інформацію про частоту появи кожного слова в документі. Значення в словнику мають бути відсортовані в порядку спадання. Порівняння документів за їх словниками може відбуватися різними способами. Як правило, обирають n слів з найбільшим показником «ваги» і для них обчислюють контрольну суму CRC32. Документи вважаються схожими, якщо ці суми близькі за значенням.

До переваг частотних методів відносять високі показники точності та швидкодію. Недоліками є: посередні показники повноти, низькі показники точності при порівнянні документів, що значно відрізняються за розмірами один від одного.

До методів, побудованих на шинглюванні відносяться: Log Shingles, Megashingles. Методи даного класу часто використовуються пошуковими системами через свою простоту та ефективність. Текст поділяється на послідовні фрагменти тексту, які називають «шинглами». Далі для кожного шингла визначають контрольну суму, на основі якої відбувається порівняння текстових документів. Даний клас методів не використовує побудову словників, що позитивно впливає на швидкодію. Проте, контрольні суми шинглів дуже чутливі до найменших змін, що може негативно вплинути на точність.

До переваг методів, побудованих на шинглюванні, відносять швидкість виконання алгоритму. Недоліками є: коротка вибірка шинглів або її відсутність при невеликих розмірах документу, низькі показники повноти.

До методів, побудованих на використанні хешування, відносяться: MD5, SHA256, SHA512. Визначення, чи є документи дублікатами, відбувається шляхом порівняння значень звичайного хешування кожного з документів в повному обсязі заданим алгоритмом, наприклад MD5.

Використання методів даного класу орієнтовано радше для виявлення чітких дублікатів, і вони мають найвищі показники точності серед усіх існуючих методів. Проте, показник повноти є дуже низьким в порівнянні з іншими класами методів.

До сигнатурних відносяться наступні методи: I-Match, Long Sent, Heavy Sent. Розглянемо принцип роботи сигнатурних методів на прикладі I-Match. Спочатку для вихідної колекції документів будується словник L. Даний словник містить в собі значення слів з середніми значеннями IDF, оскільки такі слова, зазвичай, дозволяють досягти більш точних результатів при пошуку нечітких дублікатів. Тобто слова з високими і низькими значеннями IDF до уваги не беруться. Наступний крок – це створення множини слів U для кожного документа і пошук значення перетину множин U та L ($S=L \cap U$). Сигнатура – це захешоване значення перетину. Два документи вважаються схожими, якщо в них співпадають I-Match сигнатури.

До переваг сигнатурних методів відносять високу точність виявлення нечітких дублікатів. Основним недоліком сигнатурних методів є нестійкість до невеликих змін в тексті.

Модифікований метод виявлення нечітких дублікатів текстових документів

За основу модифікованого методу було взято частотний метод Term Frequency, оскільки даний метод має один з найбільших показників F-міри та точності серед усіх існуючих аналогів. Нижче представлено принцип його роботи.

Кроки під порядковими номерами 1, 2, 3 та 5 є доповненнями до обраного методу. Крок 4 є запозиченням з методу Term Frequency, проте з наступним переліком модифікацій:

- при обчисленні коефіцієнтів «ваги» пропонується використовувати нормалізацію відношення кількості входження кожного слова до розміру документа;
- замість обчислення контрольних сум CRC32 пропонується виконувати обчислення хеш-значень коефіцієнтів «ваги».

Отже, запропонований модифікований метод виявлення нечітких дублікатів виглядає таким чином:

1. Попереднє оброблення кожного з документів (стемінг; переведення всіх слів до нижнього регістру; прибирання всіх

розділових знаків, окрім крапок). Попереднє оброблення дозволить позбутися «зайвої» інформації та розділових знаків, які можуть впливати на показник точності.

2. Розбиття кожного документу на речення, сортування їх за кількістю слів у порядку спадання. Цей крок допоможе в подальшому сформувати рівноцінні за розмірами блоки для порівняння.

3. Поділ кожного документу на n блоків за розмірами. Розміри блоків пропонується визначати пропорційно кількості слів в кожному документі.

4. Ітеративне виконання порівняння n -пар блоків, що співвідносяться за розмірами.

- обчислення коефіцієнтів входження кожного слова до загальної кількості слів в блоці з врахуванням нормалізації. Нормалізація вирішує проблему недооцінки коефіцієнтів входження кожного слова в документах більших розмірів;
- побудова частотного словника та сортування його за значенням коефіцієнту в порядку спадання;
- обчислення хеш-значень перших k -коефіцієнтів, «зчеплених» в алфавітному порядку алгоритмом MD5 (показник точності якого при вимірюванні F-міри склав 1,0);
- визначення чи є блоки дублікатами, шляхом порівняння хеш-значень. Перехід до наступної пари блоків.

5. На основі результатів порівнянь блоків визначення схожості документів між собою.

Таким чином, можна припустити, що кроки 3 і 4 допоможуть підвищити показник повноти шляхом більш детального порівняння фрагментів тексту визначеного розміру, а використання частотних характеристик базового методу дозволить підвищити точність. Автори даної статті планують перевірити справедливість всіх зазначених вище припущень в процесі продовження виконання дослідження на задану тематику.

Висновки

У статті наведено узагальнений опис методів виявлення нечітких дублікатів текстових документів, вказано їх переваги та недоліки. З'ясовано, що існуючі методи спрямовані на збільшення показників повноти і точності. Також методи з високими показниками точності, як

правило, мають низькі показники повноти, оскільки вони відкидають велику кількість фрагментів тексту, які є нечіткими дублікатами.

У даній роботі було поставлено задачу, висунуто гіпотезу та представлено модифікований метод виявлення нечітких дублікатів текстових документів, який має на меті покращення показників повноти з найменшими можливими втратами у точності.

Перспективним напрямком для продовження роботи над тематикою даної статті автори вважають дослідження ефективності запропонованої модифікації методу виявлення нечітких дублікатів текстових документів і пошук шляхів до підвищення швидкодії методу.

Література

1. Метод обнаружения нечетких дубликатов электронных текстовых документов/ Л.Э. Чалая, П.Ю. Попаденко // Бионика интеллекта: научн.-техн. журнал. – 2013. – с. 1-3.
2. The truth of the F-measure [Text] / Yutaka Sasaki. – Manchester, 2007. – 1 p.
3. Copydetection mechanisms for digital documents [Text] / S. Brin, J. Davis, and H. Garcia-Molina. – 1995. – 398-409p.
4. What is hashing? – [Електронний ресурс] – Режим доступу: <https://www.educative.io/answers/what-is-hashing>
5. How to Calculate Precision, Recall, and F-Measure for Imbalanced Classification. – [Електронний ресурс] – Режим доступу: <https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification/>

УДК 681.324

**К.т.н., доцент кафедри ПЗКС, Юрчишин В.Я.,
магістрант Санжаровський А.І.**

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

АЛГОРИТМІЧНО-ПРОГРАМНИЙ МЕТОД ДЛЯ ВИЯВЛЕННЯ ФЕЙКОВИХ НОВИН НА ОСНОВІ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ

Abstract

Vasyl Yurchyshyn, Anatolii Sanzharovskyi
*Algorithmic and Software method for detecting fake news based
on machine learning algorithms*

The article considers approach to solve the problems of effective detection and prevention of the spread of fake news on social networks using the Algorithmic and Software method for detecting fake news based on machine learning algorithms. This method consists of three main parts: classification of input data by the language of the text, clustering of input data by content, detection of the presence of fake information in the input data. The proposed algorithm was programmatically implemented, and experiments were performed on test data taken from datasets.

Вступ

Стрімке поширення інформації в соціальних медіа призводить до того, що фейкові новини стали однією з найбільших проблем у сучасному цифровому світі, зокрема вони використовуються як інструмент інформаційної війни. Найбільш поширеними є підробка чи фабрикація новин, яка полягає в маніпулятивному спотворенні фактів, свідомій дезінформації. Більшість цих фейків створюються і поширюються в соціальних медіа без перевірки на відповідність та розраховані на людей, які сприймають інформацію емоційно без належного аналізу на відповідність. Це зумовлено тим, що емоційне сприйняття, особливо негативного характеру, унеможлиблює критичний аналіз, а також сприяє подальшому поширенню фейкової інформації. В Україні ця проблема також є надзвичайно актуальна. Так, за даними дослідження USAID в 2019 році [1] майже 60% українців, віком від 18 до 65 років не схильні до критичного мислення. Лише 11% змогли відрізнити фейки. При цьому 65% стверджують, що вміють відрізнити 2неправдиву інформацію в медіа.

На думку дослідників, небезпека фейкових новин та їх поширення у соціальних мережах мають характеристики та ознаки, які певною мірою можна визначити як феноменальні, а саме: швидкість розповсюдження фейкових новин усереднено вшестеро вища, ніж швидкість поширення реальних новин; вірогідність репосту фейкової новини на 70 % вища від репосту реальної новини; певні фейкові новини поширювалися у десятеро швидше ніж реальні новини й досягали довжини ланцюжка у 19 перепостів (у реальних новин це значення майже ніколи не перевищує 10 репостів) [2].

Особливо це проявляється під час форс-мажорних обставин, зокрема в умовах війни. Саме тому зростає кількість досліджень, спрямованих на виявлення (ідентифікації) такого типу інформації та вироблення запобіжників щодо її подальшого поширення. Однак, переважна більшість таких досліджень зіштовхується з низкою проблем під час ідентифікації фейкових новин в соціальних мережах. До найбільш поширених з них можна віднести: складність збирання та визначання «вручну» належності до фейкової новини; неефективність блокування в месенджерах (Facebook, WhatsApp, Twitter та інших) можливості поширення фейкової інформації та скраг на неї; поширення дезінформації верифікованими новинними агентствами або друзями чи родичами.

Постановка задачі

Основна мета роботи полягає в підвищенні середньої точності процесу виявлення фейкових новин в соціальних медіа шляхом розробки та реалізації алгоритмічно-програмного методу виявлення фейкових новин на основі алгоритмів машинного навчання.

Термінологія

Машинне навчання – це дисципліна штучного інтелекту, яка надає машинам здатність автоматично навчатися на основі даних і минулого досвіду, щоб ідентифікувати закономірності та робити прогнози з мінімальним втручанням людини.

Нейронні мережі – це набір алгоритмів, які намагаються розпізнати складові закономірності в даних за допомогою процесу, який імітує те, як працює людський мозок.

Словник – це структура даних загального призначення для зберігання групи об'єктів. Словник містить набір ключів, і кожен ключ має одне пов'язане значення. Коли надано ключ, словник просто поверне відповідне значення.

Частота терміну – зворотна частота документа (*TF-IDF*) – це числова статистика, призначена для відображення важливості слова для документа в наборі.

Обробка природної мови (NLP) – це галузь комп'ютерної науки, а точніше галузь штучного інтелекту, яка займається наданням комп'ютерам здатності розуміти текст і вимовлені слова так само, як це можуть зробити люди.

«Стоп-слова» – це набір загальноживаних слів у мові. Якщо ми видалимо з тексту слова, які дуже часто використовуються в певній мові, ми зможемо зосередитися на важливих словах.

Стемінг – це техніка обробки природної мови, яка знижує флексію слів до їхніх кореневих форм, таким чином скорочуючи слова до основної форми та прибираючи зайву інформацію.

Флексія – це процес, за допомогою якого слово змінюється для передачі багатьох граматичних категорій.

Огляд пропонованого методу

Для вирішення проблеми поширення фейкових новин, нами було розроблено алгоритмічно-програмний метод виявлення фейкових новин на основі алгоритмів машинного навчання, який за вхідні дані приймає текст змісту новин, що потребують перевірки на наявність фейкової інформації. Загалом цей метод складається з трьох етапів:

1. Класифікація вхідних даних по мові тексту;
2. Кластеризація вхідних даних по вмісту;
3. Виявлення наявності фейкової інформації в вхідних даних.

На **першому етапі** методу після отримання вхідних даних відбувається визначення мови до якої належить текст у вхідних даних. При визначенні мови, до якої належить текст у вхідних даних використовується наступний алгоритм, який передбачений працювати з певним сталим наперед заданим набором мов. Доки використовується конкретна мова в цьому алгоритмі «літерами» визначаються лише символи її алфавіту.

Під час **навчання** алгоритму:

1. За вхідні дані використовується набір текстів, які написані різними мовами.
2. Для кожної мови відбуваються наступні дії:
 - 2.1. Створюється пустий словник з назвою «словник частот».
 - 2.2. Проходячи кожен текст для даної мови від початку до кінця для кожної послідовної пари літер в тексті:
 - 2.2.1. Якщо в «словнику частот» є запис, в якому ключ має таке ж значення як дана послідовна пара літер, тоді значення лічильника запису з даним ключем збільшується на 1;

- 2.2.2. Якщо в «словнику частот» немає запису, в якому ключ має таке ж значення як дана послідовна пара літер, тоді в «словник частот» додається запис з ключем рівним даній послідовній парі літер та обнуленим лічильником в якості значення.
3. Після побудови «словників частот» для мов, використовуючи всі вхідні дані, «словники частот» перетворюються в «словники ймовірностей» шляхом отримання суми всіх значень для кожного «словника частот» і ділення всіх значень елементів кожного «словника частот» на відповідну суму.
4. Для кожного отриманого «словника ймовірностей» береться натуральний логарифм від кожного значення ймовірності і присвоюється на місце того значення ймовірності, таким чином утворюючи «словники логарифмів ймовірностей».

Під час *використання* алгоритму:

1. Алгоритм отримує на вхід один текст;
2. Для кожного «словника логарифмів ймовірностей» виконуються наступні кроки:
 - 2.1. Створюються змінні «лічильник пар літер» і «лічильник логарифмів» та призначається їм значення 0;
 - 2.2. Для кожної пари послідовних літер в тексті, що присутня в даному «словнику логарифмів ймовірностей», збільшується значення «лічильника логарифмів» на відповідне значення запису в «словнику логарифмів ймовірностей» та збільшується значення «лічильника пар літер» на 1;
 - 2.3. Ділиться значення «лічильника логарифмів» на значення «лічильника пар літер» та зберігається результат як «остаточне значення» для даного «словника логарифмів ймовірностей».
3. Мова, словник якої має найбільше отримане значення, є передбаченою мовою тексту.

Після цього виконується **другий етап** методу, в якому відбувається визначення групи, до якої належить текст у вхідних даних по змісту. При визначенні групи, до якої належить текст у вхідних даних використовується наступний алгоритм, який використовує отримані результати першого етапу.

Перед виконанням даного алгоритму спочатку відбувається обробка та зміна тексту вхідних даних:

1. Видаляються з вхідних даних тексту всі символи, які не є літерами визначеної мови тексту та замінюються на пробіли;
2. Переносяться всі літери тексту до нижнього реєстру;
3. Текст розбивається пробілами на список слів, при цьому прибираючи пусті слова;

4. Вилучаються «стоп-слова» зі списку слів;
5. Проводиться Стеммінг кожного слова.

Під час **навчання** алгоритму:

1. При навчанні даний алгоритм приймає за вхідні дані лише тексти однією і тією ж мовою для конкретної моделі (до попередньої обробки тексту вхідних даних);
2. Створює список «унікальних слів» і робить його пустим;
3. Для кожного навчального тексту спочатку після попередньої обробки тексту відбувається додавання всіх слів зі списку слів даного тексту до списку «унікальних слів» яких там немає.
4. Після цього для кожного навчального тексту:
 - 4.1. Використовуючи список «унікальних слів» визначаються значення TF-IDF для кожного «унікального слова», використовуючи список слів даного тексту в якості документу;
 - 4.2. Створюється вектор зі значень TF-IDF[3] для даного тексту.
5. Використовуючи алгоритм k-means++ кластеризуються тексти за їхніми векторами.

Під час **використання** алгоритму:

1. Для кожного слова зі списку «унікальних слів» визначаються значення TF-IDF в даному списку слів та таким чином створюється вектор значень;
2. Кластеризується цей вектор, використовуючи алгоритм k-means++[4].

Насамкінець, виконується **третій етап** методу, в якому виявляється наявність або відсутність фейкової інформації у вхідних даних. При визначенні наявності фейкової інформації у вхідних даних використовується наступний алгоритм.

На цьому етапі використовується оброблений текст вхідних даних з другого етапу. Після виконання першого і другого етапів використовується алгоритм BERT (двосторонніх представлень кодувальників з перетворювачів) для класифікації вхідних даних на наявність фейкової інформації. Для кожної унікальної комбінації мови в першому етапі та кластеру в другому етапі створюється своя модель алгоритму BERT при навчанні. Після цього при використанні даного методу вибирається відповідна модель в залежності від результатів першого та другого етапів. Таким чином можна зробити моделі алгоритму BERT меншого розміру та покращити точність їх результатів.

Алгоритм представлення двонаправлених кодерів від трансформаторів (BERT)[5,6] – це алгоритм NLP, що призначений для попереднього навчання глибоких двонаправлених представлень на непозначеному тексті шляхом визначення залежностей та контексту між

всіма словами. Як наслідок, щоб точно налаштувати попередньо навчену модель BERT достатньо використати один додатковий вихідний рівнем для створення найсучасніших моделей для широкого спектру завдань. Цей алгоритм був створений за підтримки компанії Google в 2018 році. Його використання на третьому етапі алгоритмічно-програмного методу виявлення фейкових новин було вибрано через його можливість розрізняти контекст слів в тексті, що виокремлює його серед багатьох інших алгоритмів і зробило його одним з ключових алгоритмів в галузі NLP.

Наперед навчені моделі BERT можна знайти за посиланням [7]. В цьому алгоритмі застосовується модель «bert-large-uncased». Цей алгоритм має 4 види використання:

1. Задача класифікації пари речень;
2. Задача класифікації одного речення;
3. Задачі отримання відповіді на запитання;
4. Задача тегування одного речення.

В алгоритмічно-програмному методі виявлення фейкових новин алгоритм BERT застосовується таким чином:

1. Сточатку вхідні дані (оброблені на другому етапі) будуть перетворені таким чином:
 - 1.1. Обробляються всі слова у вхідних даних, використовуючи спеціальний вбудований токенизатор BERT;
 - 1.2. Якщо довжина списку слів більша ніж певне значення (128), то розбивається список слів на набір списків слів таким чином, щоб кожен список в наборі був менший за вище зазначене значення;
 - 1.3. Для кожного списку слів в наборі:
 - 1.3.1. Додається спеціальне слово «[CLS]» перед початком списку слів;
 - 1.3.2. Додається спеціальне слово «[SEP]» після кінця списку слів;
 - 1.3.3. Якщо даний список слів досі менший ніж вище вказане значення, то додається спеціальне слово «[PAD]», доки довжина даного списку не буде рівна вище вказаному значенню;
 - 1.3.4. Створюються «маски уваги» для всіх слів «[PAD]» в списку слів та додаються ці «маски уваги» до відповідного списку слів;
 - 1.3.5. Перетворюються всі слова в списку слів на відповідні індекси, використовуючи спеціальний вбудований токенизатор BERT.
2. Для кожного списку індексів в наборі:
 - 2.1. Використовуючи 2-й вид роботи алгоритму BERT та дану модель обробляється список індексів та відповідна «маска уваги» і отримується результат (значення від 0 до 1);

2.2.Результат свідчить про наявність фейкової інформації, якщо значення більше чи дорівнює 0.5, або про їх відсутність – якщо менше 0.5.

3. Фінальним результатом в попередньому кроці буде те значення, яке трапилось більше раз, або наявність фейкової інформації, якщо кількість обох значень є однаковою.

3.1.Якщо відбувається навчання моделі: використовуючи фінальний результат порівнюється його з очікуваним та при нерівності - оптимізується модель, використовуючи оптимізатор Adam.

Алгоритм адаптивної оцінки моменту (Adam) [8,9] – це алгоритм оптимізації, який можна використовувати замість класичного градієнтного спуску для ітераційного оновлення ваг мережі на основі навчальних даних. Його головною перевагою над подібними алгоритмами є якого можливість адаптивно змінювати «швидкість навчання» моделі під час його використання, а також швидкість виконання та мала ресурсоемкість.

Отже, під час першого етапу цього методу відбувається навчання на великих наборах текстів відповідними мовами. На другому та третьому – навчання відбувається на одному і тому самому наборі даних, що містять фейкові новини. Кожна мова має власний набір кластерів k-means++ після навчання. Кожен кластер має власну модель BERT після навчання. Через це набори кластерів та відповідні моделі BERT для різних мов використовують набори даних, що містять фейкові новини відповідною мовою. Навчання кожної частини відбувається по порядку.

Висновки

У даній статті запропоновано алгоритмічно-програмний підхід до виявлення фейкових новин на основі використання алгоритмів машинного навчання. Запропонований підхід дозволяє точніше виявляти фейкові новини за рахунок попереднього розподілу вхідних даних по мові та вмісту, а також зменшити складність обчислень. Попри це він повільніший, ніж інші методи глибинного навчання, хоча він досі достатньо швидкий, щоб використовувати його в режимі реального часу. Також він вимагає більшу кількість часу для навчання та зберігання.

В подальшому можна дослідити використання інших алгоритмів для попереднього розподілу вхідних даних для прискорення та уточнення загального алгоритму.

Література

1. Опитування USAID-INTERNEWS. Ставлення населення до ЗМІ та споживання різних типів медіа у 2019 р. URL: <https://ms.detector.media/mediadoslidzhennya/post/23680/2019-10-23-lyshe-11-ukraintsiv-zmogly-vidriznyty-feyky-doslidzhennya/>
2. Democracy under Lockdown. URL: <https://freedomhouse.org/reports>
3. TF-IDF. URL: <https://uk.wikipedia.org/wiki/TF-IDF>
4. David A., Sergei V. k-means++: the advantages of careful seeding. URL: <https://theory.stanford.edu/~sergei/papers/kMeansPP-soda.pdf>
5. Malakhov A., Patruno A. Bocconi S. Fake News Classification with BERT. URL: <http://ceur-ws.org/Vol-2882/paper38.pdf>
6. Jacob D., Ming-Wei C., Kenton L., Kristina T. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. URL: <https://arxiv.org/pdf/1810.04805.pdf>
7. About TensorFlow code and pre-trained models for BERT/ URL: <https://github.com/google-research/bert>
8. Diederik P., Jimmy L. Adam: A Method for Stochastic Optimization. URL: <https://arxiv.org/abs/1412.6980>
9. Gentle Introduction to the Adam Optimization Algorithm for Deep Learning. URL: <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>

К.т.н., асистент Погорелов В.В., магістрант Кравчук А.А.

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

МОДИФІКОВАНИЙ МЕТОД ВИЯВЛЕННЯ DDoS-АТАК ПРИКЛАДНОГО РІВНЯ НА РЕСУРСИ КОМП'ЮТЕРНИХ СИСТЕМ

Abstract

Volodymyr Pogorelov, assistant, PhD; Kravchuk Arkadii, student
*Combined method of detecting application layer DDoS attacks on computer systems
resources*

This paper concerns the task of fast detection of application-level DDoS attacks on webservers. Existing algorithms based on statistical analysis and machine learning are studied and discussed. The modified method with calculating informational entropy of HTTP request attributes is proposed. The comparative analysis of performance of both the SVM method and the modified algorithm is fulfilled. The ways for further research are proposed as well.

Вступ

Сьогодні тренд діджиталізації, тобто цифрової трансформації суспільства, активно поширюється. Але на заваді цьому можуть стати DDoS-атаки, відсутність захисту від яких спричиняє простій атакованої системи, що призводить до матеріальних збитків, втратити репутації та довіри клієнтів. З кожним роком кількість пристроїв, підключених до мережі Інтернет, постійно збільшується, як і кількість проведених DDoS-атак, за даними огляду експертів [1]. Проблема захисту ресурсів комп'ютерних систем від таких атак полягає саме в своєчасному виявленні факту проведення атаки і встановленню її джерел. Відповідно до наявних досліджень [2], для ідентифікації DDoS-атак прикладного рівня немає водночас ефективного та швидкодіючого методу, бо класичні методи, які добре підходять для мережевого рівня, не є дуже точними в даному випадку.

Таким чином, проблема виявлення розподілених атак на відмову в обслуговуванні ресурсів комп'ютерних систем є актуальною та потребує дослідження. Тому у даній роботі пропонується модифікований метод виявлення DDoS-атак прикладного рівня, який швидко виявлятиме джерела атак, разом з тим не втрачаючи точності ідентифікації.

Постановка задачі

Метою даного дослідження є підвищення ефективності аналізу даних Інтернет-трафіку для виявлення DDoS-атак прикладного рівня шляхом розробки та програмної реалізації методу, у якого час реагування на вторгнення буде менший, ніж в існуючих методів та алгоритмів. Відповідно до вказаної мети необхідно розв'язати такі задачі: оглянути наявні методи розпізнавання DDoS-атак прикладного рівня, запропонувати новий метод виявлення таких атак зі збільшеною швидкістю, експериментально перевірити ефективність розробленого методу.

Термінологія

LR-DDoS (скор. від англ. low-rate DDoS) – низькошвидкісні DDoS-атаки, що відзначаються низькою частотою запитів та націленістю на вичерпування ресурсів саме прикладних програм.

Middleware – шар ПЗ, який є посередником між компонентами певного фреймворку. Зазвичай підключається як додатковий програмний модуль, який має доступ до компонентів системи та може виконувати свій алгоритм.

Аналіз існуючих підходів та алгоритмів

DDoS-атаки можуть поділятися за темпом надсилання запитів на дві категорії: високошвидкісні та низькошвидкісні. Високошвидкісні атаки виснажують канали передачі даних, в свою чергу LR-DDoS – обчислювальні ресурси. Останні мають низьку швидкість і використовують ресурсоємні операції, як, наприклад, запис в базу даних великого об'єму інформації.

Високою швидкістю виявлення DDoS-атак вирізняється метод, який обчислює інформаційну ентропію [3], бо розраховується за достатньо простою формулою. Але в звичайній реалізації цей метод використовує атрибути мережевого рівня моделі OSI. Тому для виявлення LR-DDoS атак цим методом необхідно модифікувати процес збору параметрів, додавши атрибути з прикладного рівня, та вивести спосіб обчислення значень їх ентропії. Крім цього, методи машинного навчання, а саме: метод опорних векторів, штучні нейронні мережі – мають високу точність ідентифікації LR-DDoS атак, але водночас мають і велику обчислювальну складність [4].

Опис запропонованого методу

Одним із найпопулярніших прикладних протоколів в мережі Інтернет є протокол HTTP(S) для вебсторінок, оскільки зазвичай саме вебсайт є кінцевим інтерфейсом для отримання інформації чи надання послуг. Тому запропонований метод розроблявся для протоколу HTTP(S).

По-перше, було запропоновано модифікувати метод на основі інформаційної ентропії. В звичайній реалізації даного методу обчислюється ентропія кількості мережевих пакетів за наступною формулою:

$$H(x) = -p(x)\log_2 p(x) \quad (1)$$

де $p(x)$ – ймовірність появи символу x з певного алфавіту.

В формулі (1) для звичайної реалізації методу ймовірність появи пакетів $p(x)$ від джерела k обчислюється наступним чином:

$$p(x) = x_k / \sum_i x_i \quad (2)$$

де x_i – кількість пакетів від i -го джерела за встановлений проміжок часу.

Згідно з дослідженням [5] було визначено атрибути HTTP запитів, аналіз яких може вказати на факт проведення LR-DDoS атаки, а саме: URI запиту та його HTTP метод, клієнт (User-Agent) запиту, розмір запиту, час обробки запиту вебсервером. Для обчислення ентропії вищезазначених характеристик, необхідно адаптувати формулу (1) та вивести спосіб розрахунку ймовірностей появи цих атрибутів, який використовується в формулі (2). Нехай, кожен запит можна представити множиною значень $\{s, u\}$, де: s – IP адреса та порт джерела, u – URI та метод запиту. З набору множин всіх запитів для кожного унікального джерела s_i можна визначити унікальні URI і методи запитів u_j^i , а також кількість повторів відповідної унікальної пари – позначимо як $c(u_j^i)$. Тоді для джерела s_i ентропія такого параметру як URI та метод запиту H_u становитиме:

$$H_u(s_i) = -\sum_j p_u(u_j^i) \log_2 p_u(u_j^i); \quad p_u(u_j^i) = \frac{c(u_j^i)}{\sum_i \sum_j c(u_j^i)} \quad (3)$$

де $p_u(u_j^i)$ – ймовірність появи унікальної пари URI і методу запиту u_j^i серед всіх зафіксованих пар за встановлений проміжок часу.

Відповідно для інших атрибутів, а саме: клієнт запиту, розмір запиту – ентропія обчислюється аналогічно як в формулі (3). Тільки ймовірність для характеристики часу обробки запиту вебсервером буде розраховуватись іншим чином. Час обробки запиту можна вважати неперервною випадковою величиною. Для запитів з однаковим URI та методом час надання відповіді сервером теж буде приблизно однаковим, тому можна сказати, що розподіл цієї величини є нормальним. З цього випливає, що ймовірність $p_i(t_j^i)$ для характеристики часу обробки t_j^i певного запиту від i -го джерела для j -ої пари URI та методу запиту буде дорівнювати ймовірності попадання значення нормально розподіленої випадкової величини в заданий інтервал:

$$p_t(t_j^i) = \Phi\left(\frac{t_j^i + \delta - a}{\sigma}\right) - \Phi\left(\frac{t_j^i - \delta - a}{\sigma}\right) \quad (4)$$

де $\Phi(x)$ – значення функції Лапласа для змінної x ; a – середнє значення часу обробки запиту для відповідної пари URI та методу запиту; σ – середнє квадратичне відхилення значення часу обробки запиту для відповідної пари URI та методу запиту; δ – додатнє значення для допустимого часового інтервалу, яке близьке до 0, причому $\delta \ll t_j^i$.

Формулу (4) слід використати як значення ймовірності в формулі (3) для обчислення ентропії характеристики часу обробки запиту H_t .

Загальна архітектура запропонованого методу складається з наступних етапів: збір, обробка, класифікація даних та ідентифікація джерел атак. Етап збору даних має свої особливості, адже треба отримати інформацію з декількох рівнів моделі OSI. Вхідні пакети транспортного рівня можна отримати, якщо є доступ до маршрутизатора або мережевого інтерфейсу. Але так як один HTTP запит може бути розбитий на декілька пакетів, то необхідну інформацію прикладного рівня треба отримати за допомогою розробленого middleware для вебсерверу. Тому в даному дослідженні запропоновано програмний модуль для вебфреймворку (на прикладі Express.js), який вимірює час обробки кожного запиту та зберігає поля заголовків з необхідними атрибутами. Щоби зіставити потім ці дані з даними відповідних мережевих пакетів, middleware для кожного запиту визначає IP-адресу та порт відправника і отримувача. Зібрані дані з різних джерел відправляються до брокера повідомлень, наприклад, Kafka.

Обробка даних полягає у формуванні єдиного об'єкту параметрів з транспортного та мережевого рівнів: для цього серед зібраної інформації об'єднуються атрибути з однаковим часом захоплення, адресами відправника та отримувача. Для класифікації формується масив об'єктів з оброблених параметрів за певний часовий проміжок розміром Δt . Даний масив надається вищеписаному методу на основі інформаційної ентропії для отримання відповідних результатів. На заключному етапі ідентифікації джерел атак використовуються обчисленні показники ентропії визначених атрибутів кожного джерела, яке було зафіксовано в даному часовому проміжку, та порівнюються з пороговим значенням, внаслідок чого джерело може бути позначено як зловмисне. Порогове значення встановлюється емпірично та може коригуватись відповідно до зміни статистичних показників, наприклад, дисперсії величини одного з атрибутів.

Результати експериментальних досліджень

Запропонований метод було реалізовано на мові програмування Python. Для порівняння взято метод SVM з наявного дослідження [4] і також реалізовано на цій ж мові. Було використано датасет 2017 CIC DoS як вхідні дані. Для оброблення кожної тисячі значень з масиву вхідного набору даних в середньому витрачалось 2.46 секунд запропонованим методом, а методом SVM – 3.89 секунд на одному й тому самому комп'ютері. Отже, запропонований в даному дослідженні метод має дійсно більшу швидкодію.

Висновки

Отже, дана стаття була присвячена розробці модифікованого методу виявлення DDoS-атак прикладного рівня. Запропонований метод дозволяє підвищити швидкодію ідентифікації джерел DDoS-атак на вебсервери. Ідея дослідження полягала в тому, щоб використати метод на основі інформаційної ентропії, який має малу обчислювальну складність, та адаптувати його для прикладного рівня, шляхом виведення способу обчислення ентропії атрибутів HTTP запиту, серед яких є час його обробки.

Подальшого дослідження потребують наступні питання: алгоритм коригування порогового значення для ідентифікації атак; використання розподіленого обчислення для нових додаткових параметрів (наприклад, статистичних) на кластерах Nadoop екосистеми або Kubernetes.

Література

1. Cyberedge Group 2021 Cyberthreat Defense Report [Електронний ресурс] / CyberEdge. — Режим доступу: <https://cyber-edge.com/wp-content/uploads/2021/04/CyberEdge-2021-CDR-Report-v1.1-1.pdf>.
2. Kaur, P. A review of detection approaches for distributed denial of service attacks [Text] / P. Kaur, M. Kumar, A. Bhandari // Systems Science & Control Engineering. — 2017. — Vol. 5, № 1. — P. 301 – 320.
3. Bhuyan, M.H. E-LDAT: a lightweight system for DDoS flooding attack detection and IP traceback using extended entropy metric [Text] // Security and Communication Networks. — 2016. — Vol. 9, № 16. — P. 3251 – 3270.
4. Pérez-Díaz, J.A. A Flexible SDN-Based Architecture for Identifying and Mitigating Low-Rate DDoS Attacks Using Machine Learning [Text] // IEEE Access. — 2020. — Vol. 8. — P. 155859 – 155872.
5. Mohammed, A. Novel Protective Framework for Defeating HTTP-Based Denial of Service and Distributed Denial of Service Attacks [Text] // The Scientific World Journal. — 2015. — Vol. 2015, Article ID 238230.

УДК 004.021

Д.т.н., доцент Сулема Є. С., магістрант Песчанський Д. Ю.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

МЕТОД ВІЗУАЛЬНОГО МОДЕЛЮВАННЯ ДЛЯ ТЕХНОЛОГІЇ ЦИФРОВИХ ДВІЙНИКІВ

Abstract

Yevgeniya Sulema, DSc, assoc. prof.; Peschanskyi Danyil, student
Visual modeling method for digital twins technology

This work considers the task of creating a visual modeling method for a digital twin of an internal combustion engine. The method of developing a digital twin is described, the special characteristics of the method are highlighted and it is divided into components. The development of a behavioral model of a digital twin is described in detail with formulas and graphs.

Вступ

В наші часи людське суспільство збудовано на принципі споживання, саме ця тенденція послугували причиною створення різноманітних винаходів, що покращували людський побут. Кожна людина користується тими, чи іншими машинами, що були винайдені для спрощення життя, і з кожним роком, ці машини змінюються щоб забезпечити нові потреби людей.

З моменту появи воя Канью автомобілі також пройшли величезну еволюції, як єдиний механізм, так і кожна їх деталь окремо. Наразі існує більше сотні кампаній, які займаються розробкою та створенням автомобілів. Так як конкуренція притаманна галузі промисловості в цілому, вона має місце і в машинобудуванні. Основними критеріями успішного збуту продукту завжди є час його появи на ринку, та якість. Для поліпшення результатів обох критеріїв сучасні виробники використовують технологію цифрових двійників

Постановка задачі

Метою даної роботи є підвищення ефективності процесу моделювання та створення двигуна внутрішнього згоряння (ДВС) [1, 2] шляхом розробки та програмної реалізації методу візуального моделювання цифрового двійника двигуна. Відмінною рисою методу стане створення

бібліотеки прототипів, що значно спростять створення нових екземплярів ДВС на основі вже існуючих. На даний момент існує декілька ключових підходів побудови ДВС, для роботи було обрано двигун, принцип роботи якого оснований на наявності чотирьох послідовних тактів виконання, такі двигуни називаються чотиритактними. Вибір саме одного типу двигуна був зумовлений складністю механізму і значною різницею побудови різних видів двигунів. Відповідно до вказаної мети необхідно розв'язати такі задачі: створити метод візуального моделювання ДВС, дослідити клас заданих об'єктів з точки зору фізичних явищ та процесів, що супроводжують його функціонування, побудувати поведінкову модель вказаного об'єкту.

Термінологія

ДВС – двигун внутрішнього згоряння;

ВМТ – верхня мертва точка;

НМТ – нижня мертва точка;

Кут ПКВ – кут повороту колінчатого валу.

Метод візуального моделювання цифрового двійника двигуна внутрішнього згоряння

Розробка цифрового двійника складається з двох етапів: створення візуальної моделі об'єкту дослідження, створення поведінкової моделі об'єкту дослідження

Візуальне моделювання цифрового двійника відбувається за допомогою спеціально розробленого редактора, написаного за допомогою мови програмування С або С++. Метод передбачає наявність бібліотеки прототипів двигунів внутрішнього згоряння, прототип являє собою попередньо збережений цифровий двійник або стандартну модель, з вибору якої розпочинається створення нового екземпляра цифрового двійника. Модель поділяється на примітивні складові, геометричні фігури на кшталт гексаєдрів, тетраєдрів, тощо. Функція поділення моделі на менші складові є рекурсивним і може відбуватись окремо для різних частин моделі, це зроблено для збільшення деталізації та гнучкості редактора. Після зміни габаритних параметрів моделі редактор надає можливість обрати структуру для кожної складової моделі, також керуючись принципом рекурсивного поділення елементів моделі. Після зазначених етапів процес створення візуальної моделі цифрового двійника завершується.

Створення поведінкової моделі полягає у дослідженні фізичних процесів, що відбуваються під час роботи ДВС, і їх моделюванні за допомогою відомих технологій цифрового представлення і розрахунків

фізичних явищ. Як було зазначено в даній роботі розглядається чотиритактний двигун, отже необхідно розглянути фізичні процеси, які проходять під час кожного з тактів. Загальна робота двигуна такого типу складає собою роботу окремих чотирьох поршнів, кожен з яких перебуває у одному з станів, при чому, стани не повторюються, тобто у кожен момент часу двигун виробляє механічну роботу, яка забезпечена одним з поршнів. Принцип роботи ДВС – це перетворення теплової енергії на механічну роботу, яка певним чином впливає на провідну вісь автомобіля, я забезпечує його рух. Отже можна сказати, що побудова поведінкової моделі двигуна внутрішнього згоряння полягає у постійному вимірюванні параметрів об'єму (V), тиску (P) та температури (T) у кожній з камер згоряння.

Перший такт – впуск, під час цього такту суміш паливних речовин та повітря потрапляють у одну з камер згоряння через вхідний клапан, цей клапан відкривається лише на час впуску і закривається в кінці такту. Загальна зміна об'єму під час впуску пального у камеру описується наступним рівнянням:

$$dV = \partial_{ad}V + \partial_s V + \partial_k V, \quad (1)$$

В той час як відношення тиску до кута повороту колінчатого валу поршня описується графіком зображеним на рис. 1.

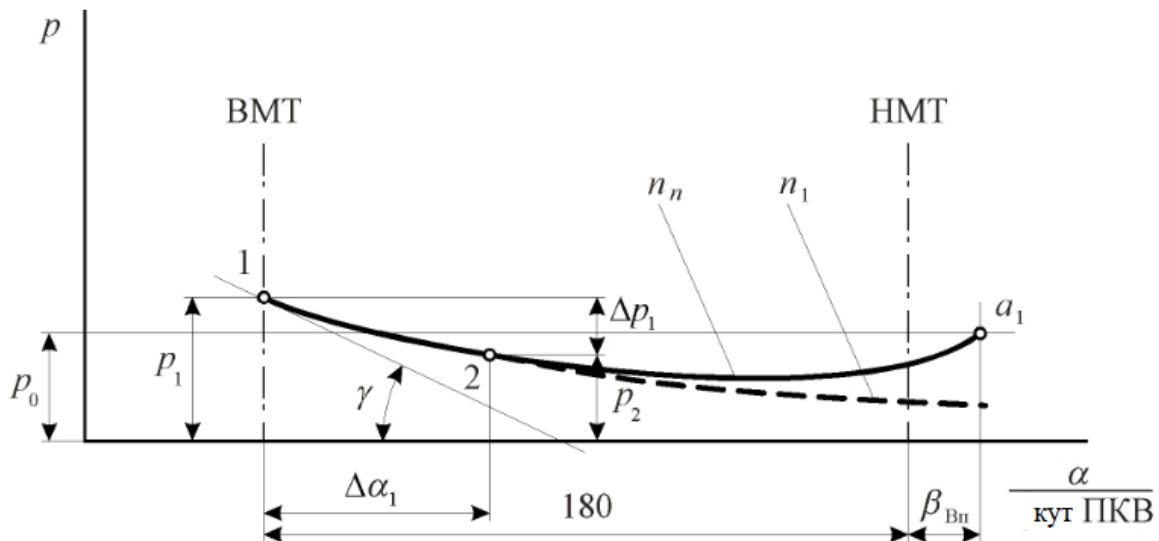


Рис 1. Залежність тиску від кута повороту колінчатого валу поршня (α) під час такту впуску

Другий такт – стиск, під час даного такту відбувається стиск речовини, що знаходиться у камері згоряння, під час цього процесу камера є герметичною. Стиск пальної речовини в замкнутому просторі призводить до підняття тиску та температури у камері, тобто накопичення теплової

енергії у камері. На рис. 2 зображено графіки залежностей, що описують цей такт.

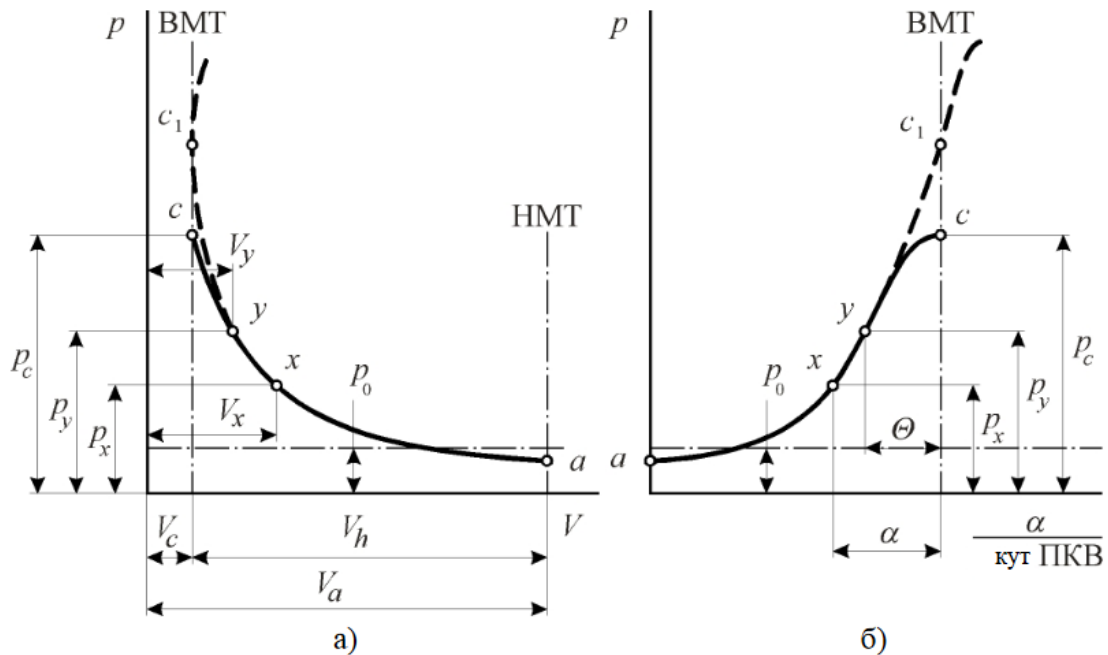


Рис 2. Схема індикаторної діаграми процесу стиснення в різних системах координат: а) "P - V" - діаграма; б) "P - α " - діаграма

Третій такт – розширення, можна вважати, що процес розширення робочого тіла в двигуні протікає за політропічним законом. У цьому випадку тиск і температура змінюються в процесі розширення залежно від обсягу відповідно до рівнянь:

$$pV^{n_2} = Const, \quad (2)$$

$$TV^{n_2-1} = Const, \quad (3)$$

де n_2 – показник політропи розширення.

Значення змінних може бути вираховано за допомогою графіка зображеного на рис. 3 для двигуна з ізохорним згорянням.

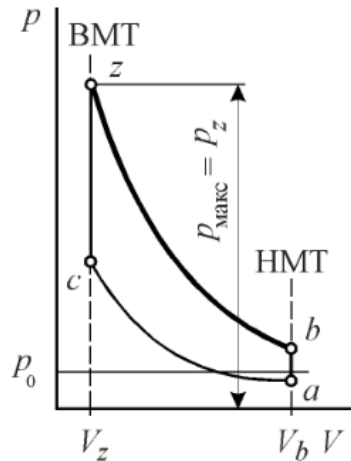


Рис 3. «P – V» діаграма двигуна для циклу із ізохорним згоранням

Четвертий такт – випуск. Після розширення в камері згорання залишаються відходи продукту горіння, під час даного такту ці відходи виходять з камери через вихідний клапан, що відкривається тальки на час цього такту. Температура тіла всередині комірчки при цьому залежить від способу утворення пального, а залежність об’єму від тиску зображена на рис. 4.

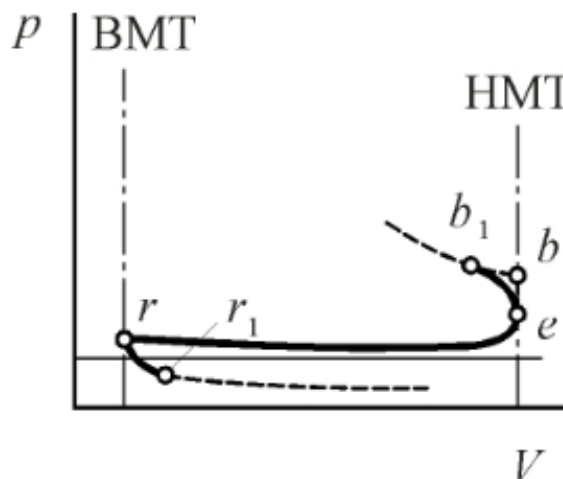


Рис 4. «P – V» діаграма двигуна під час такту випуску

Висновки

У цій статті запропоновано новий метод візуального моделювання цифрового двійника двигуна внутрішнього згорання, відмінною рисою методу є наявність бібліотеки прототипів, що значно спрощують створення нових цифрових двійників розглядаємого об’єкту. Зазначена актуальність даного дослідження в умовах сучасного світу, та сформовано поставлену задачу. Також були зроблені перші кроки реалізації даного методу, а саме було описано етап створення візуальної моделі цифрового двійника,

розглянуто кожен етап роботи чотиритактного ДВС і побудовано формули та графіки які описують поведінкову модель даного об'єкту, не всі з них було зазначено у роботі через їх великий об'єм. Зазначено які аспекти побудови цифрового двійника мають бути реалізовані ради досягнення оголошеної мети.

Література

1. Lino Guzzella, Christopher Onder. Introduction to Modeling and Control of Internal Combustion Engine Systems. Springer Science & Business Media, 2009. Technology & Engineering, 354 p.
2. Fiore M., Magi V., Viggiano A. Internal combustion engines powered by syngas: A review. Applied Energy, Vol 276, 2020, paper 115415.

УДК 519.688

К.т.н., доцент Люшенко Л.А., студент Перегуда Я.І.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

МОДИФІКОВАНИЙ СПОСІБ ПРОГНОЗУВАННЯ ДИНАМІКИ ВАЛЮТНОГО КУРСУ

Abstract

Lesya A. Lyushenko, assoc. prof.; Yaroslav Perehuda, student

Modified method of forecasting the dynamics of foreign exchange market prices

This paper concerns the task of forecasting the dynamics of foreign exchange market prices. The classical SARIMA mathematical model is studied and discussed. The modification of said model based on the linear trend model is proposed. The ways for further research are proposed as well.

Вступ

Задача прогнозування динаміки валютного курсу є економічною актуальною задачею, яка дозволяє передбачати виникнення ризиків дестабілізації економічних процесів як окремої країни, так і світу в цілому. Існує багато методів прогнозування валютних курсів, для кожного з яких потрібні свої окремі набори вхідних даних. Вони можуть включати в себе часові ряди валютних курсів, цін на золото, індекс споживчих цін, певні емпіричні коефіцієнти, визначені на основі спостережень або певного досвіду, тощо.

У даній статті пропонується для прогнозування динаміки валютного курсу модифікувати математичну модель SARIMA, яка уточнюється за рахунок лінійних трендів коливання курсу валют. Така модифікація дозволяє враховувати вплив різних чинників, а не тільки сезонні коливання курсу валют, які визначаються при аналізі динаміки часового ряду.

Постановка задачі

Задача полягає в побудові способу прогнозування валютних курсових різниць, використовуючи модифіковану математичну модель SARIMA, яка уточнюється за рахунок лінійних трендів коливання курсу валют.

Термінологія

Валютний курс – вираження ціни грошової одиниці однієї країни в грошових одиницях іншої.

Прогнозування валютних курсових різниць – передбачення значень валютного курсу за допомогою математичних чи економічних моделей, методу обліку паритету купівельної спроможності, методу обліку економічної стабільності тощо.

Математична модель SARIMA – один із видів класу математичних моделей ARMA, призначених для прогнозу значень часових рядів на основі власних лагів та відхилень. Модель SARIMA відрізняється від інших моделей тим, що дозволяє враховувати сезонність часових рядів.

Опис способу

Динаміка валютних курсів зазвичай подається як графік, у якому кількість одиниць однієї валюти, яку потрібно витратити щоб придбати одиницю іншої валюти співвідноситься до часу, що в свою чергу формує відповідний часовий ряд.

Існує досить багато способів для аналізу часових рядів та прогнозування його подальших значень. До них входять проста та зважена ковзна середня, експонентне згладжування, модель Брауна, авторегресійна ковзна середня тощо.

В цій статті увага приділятиметься модифікованому методу прогнозування часового ряду шляхом використання математичної моделі на базі ARMA – SARIMA та лінійного тренду.

Моделі ARMA (Autoregressive Moving Average Model) – це клас моделей авторегресійного ковзного середнього, одна з математичних моделей, що використовується для аналізу та прогнозування часових рядів у статистиці. Основу моделей даного класу складають дві основні суб моделі: авторегресійна модель AR, та модель ковзного середнього MA [1].

Варто звернути увагу, що ARMA застосовується тільки для стаціонарних часових рядів. Враховуючи, що часовий ряд валютних цін не є стаціонарним, дана модель для прогнозування валютного курсу не підходить. Для таких випадків існує варіація класичної моделі під назвою ARIMA. Вона дозволяє перед початком роботи привести часовий ряд до стаціонарного вигляду.

Хоча модель ARIMA вже можна використовувати для прогнозування динаміки курсу валют, у неї є свої недоліки. Вона не дозволяє враховувати сезонність значень у числовому ряді. Як приклад, якщо країна закуповує багато газу в зимовий період, це може чинити сезонний тиск на валюту. Ще одним прикладом може бути період збору зернових культур в період

серпня-січня, який в свою чергу збільшує валютну виручку яка надходить до країни [2].

Таких чинників є досить багато і кожен з них в деякій мірі впливає на економічний стан в країні. ARIMA не дозволяє взяти це до уваги, на відміну від математичної моделі SARIMA.

SARIMA (Seasonal ARIMA) дозволяє працювати з нестационарними часовими рядами які мають ознаки сезонності, тобто є ідеальним варіантом для прогнозу динаміки валютного курсу.

Сезонне диференціювання цієї моделі подібне до звичайного, але замість того, щоб віднімати доданки послідовно, потрібно віднімати доданки від попереднього сезону. Додаткові лаги компенсуються частотою сезонності [1].

Представлена модель SARIMA у вигляді даної формули:

$$\Delta^d X_t = c + \varepsilon_t + \sum_{i=1}^q \theta_i X_{t-i} + \sum_{j=1}^p \varphi_j \Delta^d X_{t-j} + \sum_{k=1}^Q \phi_k X_{t-k} + \sum_{n=1}^P \eta_n \Delta^d X_{t-n}$$

де φ , θ , ϕ , η – це параметри моделі, p – кількість попередніх значень (лагів) часового ряду, d – мінімальна кількість диференцій ряду, q – кількість попередніх значень, що сильно відхилялися від середнього значення, c – константа, Δ – оператор різниці часового ряду, ε – випадкова змінна із часового ряду (білий шум).

Проте, навіть така модель не є ідеальною. Вона не дозволяє враховувати нерегулярні тенденції, які не є сезонними, проте впливають на динаміку курсу. До таких тенденцій можуть належати не тільки економічні чинники (виплата зовнішнього займу, економічна криза тощо), а й політичні чинники (вибори президента, політична криза в країні, війна тощо)[3]. Так, міграція значної кількості людей за кордон негативно впливає на економічний стан, відповідно курс валют також змінюється [4].

SARIMA не дозволяє брати до уваги такі чинники. Тому є сенс покращити алгоритм прогнозування динаміки курсу валют шляхом уточнення результатів отриманих від моделі SARIMA за рахунок лінійних трендів коливання курсу валют.

Тренд – це загальна тенденція значень при різнонаправленому русі, визначена загальною спрямованістю змін показників часового ряду.

Лінійний тренд, у свою чергу, це тренд представлений у вигляді прямої, і визначається такою формулою:

$$X_t = \alpha + \beta t$$

де α – вільний член, β – коефіцієнт при змінній.

Визначити дані константи можна за допомогою даних формул:

$$\alpha = \frac{\sum_{t=1}^n y_t * \sum_{t=1}^n t^2 - \sum_{t=1}^n t y_t * \sum_{t=1}^n t}{n \sum_{t=1}^n t^2 - (\sum_{t=1}^n t)^2}$$

$$\beta = \frac{n \sum_{t=1}^n t y_t - \sum_{t=1}^n t * \sum_{t=1}^n y_t}{n \sum_{t=1}^n t^2 - (\sum_{t=1}^n t)^2}$$

де n – загальна кількість значень в часовому ряді.

В даному випадку нас цікавить лише константа, яка відповідає за кут нахилу лінійного тренду – β .

Дана константа дозволяє в цілому оцінити рух цін на валютному ринку, визначити чи валюта має тенденцію знецінюватись, чи навпаки, набувати ціну. При кризі, валютний курс буде знижуватись, тому кут нахилу лінійного тренду буде спрямований вниз, і навпаки, при узгодженні взаємовигідної співпраці між країнами валютний курс зазвичай зростає.

Такий показник можна використати для корекції значень отриманого прогнозу часового ряду курсу валют. Для більш точного результату лінійний тренд слід вираховувати лише за певний проміжок часу, в залежності від чинників, які впливають на поведінку тренду.

В результаті ми маємо таку формулу для прогнозування динаміки цін курсу валют:

$$\begin{cases} t < M, \Delta^d X_t = c + \varepsilon_t + \sum_{i=1}^q \theta_i \varepsilon_{t-i} + \sum_{j=1}^p \varphi_j \Delta^d X_{t-j} + \sum_{k=1}^Q \phi_k \varepsilon_{t-k} + \sum_{n=1}^P \eta_n \Delta^d X_{t-n} \\ t \geq M, \Delta^d X_t = (c + \varepsilon_t + \sum_{i=1}^q \theta_i \varepsilon_{t-i} + \sum_{j=1}^p \varphi_j \Delta^d X_{t-j} + \sum_{k=1}^Q \phi_k \varepsilon_{t-k} + \sum_{n=1}^P \eta_n \Delta^d X_{t-n}) * \beta \end{cases}$$

де M – теперішній момент часу.

Так, значення прогнозу курсу валют отримані після використання моделі SARIMA далі будуть скориговані впливом лінійного тренду, який існує на момент прогнозу.

Це дозволить в процесі прогнозу динаміки цін курсу валют враховувати загальний стан економіки в країні, що дуже тісно пов'язані між собою.

Висновки

Описане у даній статті дослідження присвячене розробці оптимізованого методу прогнозування динаміки курсу валют. Запропонована модифікація методу SARIMA дозволяє підвищити точність прогнозів за рахунок використання лінійного тренду курсу валют. Це забезпечується завдяки врахуванню в прогнозі загального стану економіки країни.

Перспективним напрямком для продовження роботи над тематикою даної статті автори вважають вирахування оптимального проміжку часу за який слід вираховувати коефіцієнт лінійного тренду, а також розглянути

можливість використання інших математичних моделей які використовуються прогнозу на основі часових рядів. Як приклад, модель SARIMAX, яка дозволяє враховувати екзогенні змінні або, іншими словами, використовує зовнішні дані в прогнозі. Такими зовнішніми даними можуть бути ціна на золото, нафту, електроенергію тощо.

Література

1. *Brendan Artley*. Time Series Forecasting with ARIMA , SARIMA and SARIMAX: [Електронний ресурс] / Brendan Artley // Сайт Towards Data Science. – Towards Data Science, [2022] – Режим доступу: <https://towardsdatascience.com/time-series-forecasting-with-arima-sarima-and-sarimax-ee61099e78f6> – Назва з екрану. – Дата звернення: 22.10.2022.
2. *Владислав Павлов*. Як сезонний фактор впливає на курс євро/долара: [Електронний ресурс] / Павлов Владислав // Сайт ТОВ «Файненс.юа». – ТОВ «Файненс.юа», [2021] – Режим доступу: <https://finance.ua/ua/saving/kak-sezonnyj-faktor-vlijaet-na-kurs-evro-dollar> – Назва з екрану. – Дата звернення: 23.10.2022.
3. *Слово і Діло*. Чи впливає на курс гривні наближення виборів?: [Електронний ресурс] // Сайт ТОВ «ІА Слово і Діло». – ТОВ «ІА Слово і Діло», [2021] – Режим доступу: <https://www.slovoidilo.ua/2019/02/13/infografika/polityka/chy-vplyvaye-kurs-hryvni-nablyzhennya-vyboriv> – Назва з екрану. – Дата звернення: 23.10.2022.
4. *Грошові перекази в Україну повернулися на довоєнний рівень: хто надсилає валюту до країни*: [Електронний ресурс] // Сайт ТОВ "МінфінМедіа". – ТОВ "МінфінМедіа", [2022] – Режим доступу до ресурсу: <https://minfin.com.ua/ua/currency/articles/denezhnye-perevody-v-ukrainu-vernulis-na-dovoennyu-uroven-kto-shlet-v-stranu-valyutu> – Назва з екрану. – Дата звернення: 22.10.2022.
5. *Бардас С.А.* Економетрики: Навч. посіб. / С.А. Бардас. – Т.: ТюмДУ, 2010. – С. 202-248.

ЗМІСТ

1.	Бай Ю. П., Городецький Д. С. Математичне та програмне забезпечення системи прогнозування ризиків виникнення діабету	4
2.	Бай Ю.П., Лисий П.О. Застосування методу word2vec при побудові контентно-орієнтованої рекомендаційної системи	10
3.	Вовк Л. Б., Нікітіна М. А. Алгоритм пошуку подібних зображень на основі глибокого автоенкодера	15
4.	Вовк Л. Б., Пастернак М.О. Система для пошуку рівноваги двох підприємств з урахуванням випадкового попиту за нешем та штакельбергом	19
5.	Жук І.С. Застосування конформних предикторів і степеревих мартингалів для виявлення підозрілих матчів футбольних турнірів	24
6.	Ліскін В.О., Кошопаленко І.О. Математичне та програмне забезпечення системи діагностики ока за зображенням очного дна	30
7.	Лось В.М., Борознюк Д.О. Математичне та програмне забезпечення системи розпізнавання COVID-19 за рентгенівськими знімками	35
8.	Лось В.М., Копичко С.М., Герасименко В.Р. Модифікований метод ітерацій для систем диференціальних рівнянь	42
9.	Маслянюк П.П., Мірко С.С. Математичне та програмне забезпечення системи пошуку прецедентів в судочинстві України	48
10.	Маслянюк П.П., Коломієць С.С. Математичне та програмне забезпечення системи прогнозування та аналізу показників ефективності імпорту та експорту зовнішньої торгівлі України	57
11.	Маслянюк П.П., Питайло І.Ю. Математичне та програмне забезпечення системи оцінювання та аналізу тривожності суспільства	72
12.	Маслянюк П.П., Савчук І. В. Математичне і програмне забезпечення devops-системи продукування систем типу question-answering	78

13.	Норкін Б.В., Пашенко К.М. Математичне та програмне забезпечення системи класифікації цифрових зображень текстових документів	86
14.	Олефір О.С., Буловацький М.С. Математичне та програмне забезпечення системи прийняття рішень у коаліційній грі з неповною інформацією	93
15.	Олефір О.С., Галюк Є.І., Ковальчук-Хімюк Л.О. Вплив величини популяції при використанні машинного навчання з підкріпленням на показники швидкості навчання	99
16.	Олефір О.С., Козак А.В., Ковальчук-Хімюк Л.О. Математичне та програмне забезпечення системи класифікації комбінацій карт та оцінки ймовірності виграшу	104
17.	Ориняк І.В., Жиляєв М.І. Моделювання деформування косоного стика трубопроводу як пружної оболонки під дією внутрішнього тиску	111
18.	Рудник Т. П. Математичний метод аналізу багатомовних повідомлень із соціальної мережі твіттер	123
19.	Сирота С. В., Агафонов Д. С. Порівняльний аналіз підходів та методів оцінювання емоційного забарвлення фінансової інформації про поточний стан компанії	129
20.	Сирота С. В., Копичко С.М., Бондаренко С.А. Математичне та програмне забезпечення підсистеми аналізу настроїв текстових даних за допомогою засобів обробки природної мови	140
21.	Соловійов С.О., Копичко С.М., Страшнова А. С. Математичне та програмне забезпечення ідентифікації власника голосу на основі архітектури сіамських нейронних мереж	149
22.	Третиник В.В., Бондарчук О. О. Метод вирішення проблеми холодного старту для рекомендаційної системи фільмів	158
23.	Третиник В.В, Гуро Д.А., Андрусенко О.М. Застосування методів машинного навчання в питаннях оцінки якості життя молоді	164
24.	Третиник В.В, Давиденко М.А. Попереднє покращення якості зображення для підвищення точності розпізнавання об'єктів	169
25.	Третиник В.В, Коваленко О.П. Автоматизація підбору гіперпараметрів LSTM для задачі розпізнавання емоцій за аудіозаписами	174

26.	Violeta V. Tretynyk,; Naser J. Hamad System of clasterization of arabic papers	180
27.	Чертов О. Р., Бірук С. В. Математичне та програмне забезпечення системи купівлі / продажу електроенергії потужностей відновлюваної енергетики	186
28.	Щьоголев М.О. Методи підтримки прийняття рішень та обчислювальної лінгвістики для визначення загальної оцінки рівня напруженості в суспільстві	192
29.	Петрашенко А.В., Бербєга В.О. Оптимізація алгоритму трасування променів шляхом використання спеціалізованого апаратного забезпечення	201
30.	Боярінова Ю.Є., Воронін М. Ефективний спосіб організації телекомунікацій в спеціалізованій комп'ютерній мережі рухомих об'єктів	209
31.	Боярінова Ю.Є., Пушняк Д.С. Комп'ютерна система об'єктно-реляційного відображення із автоматичною генерацією міграцій на основі змін в моделях даних	214
32.	Боярінова Ю.Є., Шимчук Т.В. Середовище для безпечного виконання розширень застосунків	218
33.	Боярінова Ю.Є., Юрчишен Ф.В. Запобігання брандмаузерам UDP	222
34.	Дробязко І.П., Брославський Р. М. Система моніторингу процесу зрошення сільськогосподарських культур	226
35.	Дробязко І. П., Туркін М. П. Аналіз фреймворків для роботи з великими даними при вирішенні задач NLP	231
36.	Єршихін С. В., Коляда К. В., Радченко К. О. Компоненти апаратного модуля для комплексних систем моніторингу якості повітря	236
37.	Зайцев В.Г., Стабровський В.К. Моделювання виконання задачі в системі реального часу	241
38.	Клятченко Я.М., Шевченко Д.С. Порівняльний аналіз версій WORLD WIDE WEB	245
39.	Клятченко Я.М., Шепель К.С. Модифікований алгоритм SB-рендерингу	249
40.	Клятченко Я.М., Шоломіцький І.В. Модифікований алгоритм розпізнавання об'єктів на зображеннях	253

41.	Коваленко О.П., Галицький Д. Розрахунок кількості процесорів у багатошинній відмовостійкій багатопроцесорній системі	257
42.	Марченко О.І., Єрмоленко Д.В. Спосіб порівняння абстрактних семантичних дерев програм написаних LISP-подібними мовами	261
43.	Марченко О.І., Колпаков П.С. Аналіз існуючих способів оптимізації передачі однотипних JSON-документів в комп'ютерних мережах	266
44.	Марченко О.І., Хомутник Д.Ю. Порівняльний аналіз способів опису ресурсів хмарної інфраструктури	271
45.	Молчанов О.А., Государська А.О. Аналіз і структурування контенту	277
46.	Морозов К.В., Шевченко А.О. Спосіб оцінки повноти тестових наборів на основі формального методу визначення стану процесорів багатопроцесорної системи	282
47.	Павловський В.І., Денисенко І.В. Алгоритм фільтрації джерел освітлення, що не потрапляють на тайл екрану	287
48.	Павловський В.І., Ковезюк А.О. Алгоритм символної процедурної генерації контенту за текстами природної мови	293
49.	Павловський В.І., Янечко А.С. Додаток з перетворення SDR контенту в HDR з використанням модифікованого алгоритму	298
50.	Петрашенко А.В., Жовнірський Д.М. Спосіб розпізнавання ключових елементів обличчя людини	304
51.	Петрашенко А. В., Топіха М. В. Метаморфне тестування нейронних мереж для розпізнавання об'єктів	310
52.	Потапова К.Р., Дадиверін В.В. Використання та модифікування HAAR CASCADE FACE DETECTOR	314
53.	Потапова К.Р., Станкевич В.В. Модифікований алгоритм Дейкстри для керування аварійною евакуацією в приміщенні в режимі реального часу	319
54.	Потапова К.Р., Фещенко І.О. Особливості апаратно-прискореного алгоритму рендерингу на основі трасування променів	324

55.	Потурай М.В., Коляда К.В. Розпізнавання техніки в реальному часі на відеоматеріалах, що зняті БПЛА	329
56.	Романкевич В.О., Дорош К.А. Захист інформації з використанням розподілених систем	334
57.	Романкевич О.М., Мережко І.П. Модифікація способу формування GL-моделей для складних систем	339
58.	Романкевич О.М., Горба Д.О. Алгоритм формування гамільтонових циклів для генераторів псевдовипадкових чисел з різною вагою	343
59.	Тарасенко-Клятченко О.В., Білоха А.К. Оптимізація алгоритму стиснення даних Гаффмана	347
60.	Тарасенко-Клятченко О.В., Качанов В.І. Програмний модуль надсилання коротких повідомлень стільниковою мережею	351
61.	Терейковський І.А., Бойко Д.Ю. Метод розробки нейромережевої системи аналізу голосового сигналу	354
62.	Терейковський І. А., Дідус А.В. Проблематика розпізнавання голосових сигналів в малоресурсних комп'ютерних системах	357
63.	І.А. Терейковський, О.С. Коровій Обґрунтування актуальності досліджень в області розпізнавання емоційної тональності фрагментів тексту в універсальних комп'ютерних системах	361
64.	Терейковський І. А., Самофалов А. В. Проблематика формування веб-орієнтованих баз даних емоційно забарвлених голосових сигналів	365
65.	Терейковський І.А., Юрченко В.В. Про систему автоматизованого аналізу безпеки API	369
66.	Тесленко О.К., Сенеджук А.Ю. Про автоматизацію тестування у порівнянні із функціональним тестуванням	372
67.	Яценко В.О., Журба А.В. Захист даних у хмарних технологіях	375
68.	Maria Y. Antusheva, Shkurat S. Oksana The modified neural network learning algorithm in the medical image classification system	379

69.	Вернік М. О., Олещенко Л. М. Програмний метод створення рекомендаційної системи з використанням метаевристичної оптимізації	386
70.	Мороз М.В., Сущук-Слюсаренко В.І. Метод моделювання приміщень з використанням технології доповненої реальності	393
71.	Онай М.В., Гришко Ю.І. Метод множення точки еліптичної кривої на скаляр у скінченному полі	398
72.	Volodymyr Pogorelov, Wenjiao Shi The method for DDOS attacks recognition based on artificial intelligence	405
73.	Кривда П. С., Сущук-Слюсаренко В.І. Метод пошуку оптимального варіанту тріангуляції	411
74.	Сулема Є. С., Печанський В.Ю. Алгоритм зчитування та аналізу даних медико-біологічних об'єктів у форматі PDF на основі оптичного розпізнавання символів	416
75.	Tang Jiaoyan, Suschuk-Sliusarenko V.I. Chinese text representation as BGW 2d barcode	421
76.	Фалілеєва Д.М., Хіцко Я.В. Створення персонального веб-сайту з використанням алгоритмів машинного навчання	424
77.	Панков Т.С., Потапова К.Р., Кучмій О.О. Сесійний алгоритм стискання зображень	432
78.	Гулько Д.Т., Заболотня Т.М. Метод автоматизованого генерування коду шаблонів програмних компонентів клієнтської частини вебзастосунків	439
79.	Демченко В.М., Заболотня Т.М. Методи статичного аналізу вихідного коду користувача засобами мови C#	445
80.	Легеза В.П., Атаманюк О.В. Алгоритмічний метод побудови цифрового двійника одного вібрзахисного процесу	452
81.	Легеза В.П., К.Здоровенко, Нецадим О.М. Розв'язання однієї задачі переслідування в замкненій формі	458
82.	Oleshchenko L. M., Chao Wang Machine learning methods analysis for human activity recognition	464

83.	Олещенко Л.М., Вовченко Д.С. Програмний метод підбору вакансій з використанням технологій машинного навчання	470
84.	Погорелов В.В., Труш А.О. Комбінований метод розпізнавання REDOS вразливостей в регулярних виразах	477
85.	Рибачок Н.А., Манохін А.В. Методи розпізнавання англомовних акцентів	482
86.	Рибачок Н.А., Тютюнник П.Б. Метод визначення можливих ризиків та затримок в AGILE проектах	488
87.	Чумак М.С., Заболотня Т.М. Метод виявлення нечітких дублікатів текстових документів	493
88.	Юрчишин В.Я., Санжаровський А.І. Алгоритмічно-програмний метод для виявлення фейкових новин на основі алгоритмів машинного навчання	499
89.	Погорелов В.В., Кравчук А.А. Модифікований метод виявлення DDOS-атак прикладного рівня на ресурси комп'ютерних систем	507
90.	Сулема Є. С., Песчанський Д. Ю. Метод візуального моделювання для технології цифрових двійників	512
91.	Перегида, Люшенко Л.А. Модифікований спосіб прогнозування динаміки валютного курсу	518

ДЛЯ НОТАТОК

ДЛЯ НОТАТОК

ДЛЯ НОТАТОК

Наукове видання

**ПРИКЛАДНА МАТЕМАТИКА ТА КОМП'ЮТИНГ
ПМК-2022**

Збірник тез доповідей

П'ятнадцятої наукової конференції магістрантів та аспірантів

Відповідальний за випуск Сирота С.В.
Комп'ютерна верстка Дрозденко Л.В.



"ПРОСВІТА"

КИЇВ

Промислово-торговельна фірма «Просвіта»
у формі товариства з обмеженою відповідальністю.
01032, Київ, бульвар Т. Шевченка, 46,
тел. (067) 440-29-96, E-mail: prosvita.kyiv@gmail.com
Свідоцтво ДК № 221 від 16.10.2000 р.

Підп. до друку 25.11.2022 . Формат 60x84¹/₁₆. Папір офс. Гарнітура Times.
Спосіб друку – різнограф. Ум .друк. арк. 30,5. Обл.-вид. арк. 27. Зам. № .
Наклад 80 пр.

«КПІ ім. Ігоря Сікорського» Видавництво «Політехніка»
03056, київ, вул. Політехнічна, 14, корпус 15
тел./факс (044) 204-81-78