

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

**ПРИКЛАДНА МАТЕМАТИКА  
ТА КОМП'ЮТИНГ  
ПМК' 2021**



**Чотирнадцята наукова конференція магістрантів та  
аспірантів**

**Київ, 17 – 19 листопада 2021 р.  
Збірник тез доповідей  
Рекомендовано вченою радою  
факультету прикладної математики**

Просвіта  
Київ  
2021

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

**Рекомендовано вченою радою факультету прикладної математики**  
(протокол № 4 від 29.11.2021)

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., 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. Орлова М.М., д.т.н., доц. Сулема Є.С., 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.

**Прикладна** математика та комп'ютинг. ПМК, 2021 : чотирнадцята наук.  
П75 конф. магістрантів та аспірантів, Київ, 17—19 лист. 2021 р. : зб. тез доп. / [редкол.: Дичка І. А. та ін.]. — К. : Просвіта, 2021. — 368 с.

ISBN 978-617-7010-18-9

У збірнику опубліковано тези доповідей на конференції з таких напрямів: прикладна математика та прикладні рішення; прикладна математика в ІТ-технологіях; системна інженерія проектів інформатизації організаційних систем; моделювання складних систем; апаратно-програмні засоби комп'ютеризованих і комп'ютерних систем та мереж; компоненти та пристрої обчислювальної техніки і систем керування; комп'ютеризовані технологічні процеси та інформаційно-вимірювальні системи; паралельні та розподілені обчислювальні системи і мережі; технічна діагностика, тестування, надійність і відмовостійкість у комп'ютерних технологіях; кодування та ущільнення даних; захист інформації в інформаційно-комунікаційних системах; фізико-технологічні, алгоритмічні, логічні та мовно-програмні основи проектування технічних та програмних засобів; бази даних та знань, інтелектуальний аналіз та обробка інформації; інформаційні сховища та інформаційні колектори; системний аналіз та системне програмування, технології проектування програмного забезпечення комп'ютерних систем; компоненти штучного інтелекту в технічних системах; мультимедійні комп'ютерні засоби; веб-технології, комп'ютерна графіка.

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

Використання матеріалів збірника можливе за умови обов'язкового посилання.

ISBN 978-617-7010-18-9

© Факультет прикладної  
математики КПІ ім. Ігоря Сікорського, 2021

## **Передмова**

Підготовка магістрів (master) та докторів філософії (PhD) є одним з пріоритетних завдань дослідницького університету.

Особливістю підготовки магістрів є те, що поряд з традиційним навчальним процесом має здійснюватися активне залучення магістрантів до науково-дослідної діяльності.

Головним результатом навчання магістранта у вищому навчальному закладі є захист магістерської дисертації, яка має бути науковим або науково-інноваційним дослідженням.

Метою конференції «Прикладна математика та комп'ютинг» є сприяння магістрантам (аспірантам) в оприлюдненні та обговоренні результатів наукових досліджень, виконаних ними протягом навчання в магістратурі (аспірантурі). Для магістранта випускного курсу виступ на конференції є попереднім захистом наукових положень його магістерської дисертації.

Виступати на конференції мають можливість також студенти молодших курсів, які активно займаються науковою діяльністю.

У роботі конференції взяли участь понад сто учасників, 86 з них виступили з науковими доповідями.

До збірника включено тези доповідей тих учасників конференції, які виконали формальні вимоги.

На конференції працювали три секції – «Прикладна математика», «Комп'ютерні науки, комп'ютерна інженерія» та «Інформаційні технології, інженерія програмного забезпечення». Тези доповідей у межах кожної секції розміщено в алфавітному порядку за прізвищем першого автора.

**Програмний комітет ПМК – 2021**

УДК 519.688

Магістрант Брусенцов Ю.О., д.т.н., доцент Сулема Є.С.

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

## АЛГОРИТМІЧНО-ПРОГРАМНИЙ МЕТОД ПРОЦЕДУРНОЇ ГЕНЕРАЦІЇ ВОКСЕЛЬНОГО ЛАНДШАФТУ ІЗ 3D-ТАЙЛІВ

Yurii Brusentsov, student; Yevgeniya Sulema, DSc, Assoc. Prof.  
*Algorithmic-Software Method of Procedural Generation of Voxel Landscape  
from 3D-Tiles*

*The paper is devoted to solving the problem of software generation of the landscape. A method for generating a voxel landscape using 3D-tiles is proposed. The analysis of the efficiency of voxel landscape generation using the WFC algorithm is performed. The paper also suggests areas for further research.*

### Вступ

Створення та візуалізація віртуальних пейзажів є актуальним завданням у різних сферах, таких, як розроблення навчальних та моделюючих середовищ, відеоігор, створення матеріалів для фільмів та анімації, а також середовища для моделювання різних природних процесів.

Процедурна генерація – це потужний інструмент для генерації контенту, проте процедурний вміст інколи важко контролювати. У цій статті пропонується метод для генерації нескінченного воксельного ландшафту, який базується на підстановці попередньо створених користувачем 3D-тайлів.

### Постановка задачі

Основною метою дослідження є розроблення методу генерації воксельного ландшафту.

Задача полягає в тому, щоб створити алгоритм генерації нескінченного воксельного ландшафту із 3D-тайлів таким чином, щоб згенерована мапа була цілісною, та швидкість генерації була б якнайвищою.

## Термінологія

*Тайл* (від англ. *tile* – плитка) – невеликі зображення однакових розмірів, які служать фрагментами великої картини.

*Тайлсет* – набір тайлів які використовуються для створення контенту.

*Воксель* (від англ. *volume* та англ. *pixel*) – одиничний елемент простору що позначає значення певної величини в клітинках рівномірної просторової ґратки [1].

*WFC (Wave Function Collapse)* – алгоритм, який генерує бітові зображення, локально подібні до вхідного бітового зображення [2].

## Опис алгоритму

В переважній більшості випадків 3D-ландшафт генерується за допомогою карт висот [3]. Проте, у цій статті пропонується інший підхід – виконання генерації ландшафту із заздалегідь створених елементів. Для цього необхідно створити 3D-тайлсет, який буде використовуватись у розробленому алгоритмі. Для подальшої демонстрації алгоритму було створено тайлсет із 6 тайлів (рис. 1).

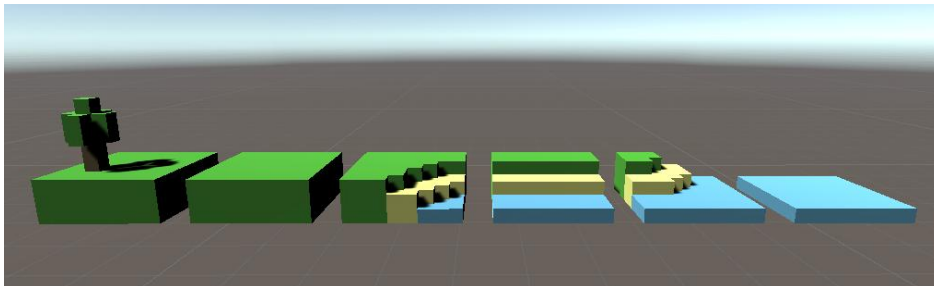


Рис. 1. Приклад воксельного тайлсету

Базовий метод полягає в тому, щоб зовнішні сторони розміщених воксельних тайлів, які стоять поруч, мали однакові кольори. Також слід зауважити, що при порівнянні тайлів у методі враховується не лише колір, але й наявність самого вокселя в конкретній позиції тайлу (рис. 2).

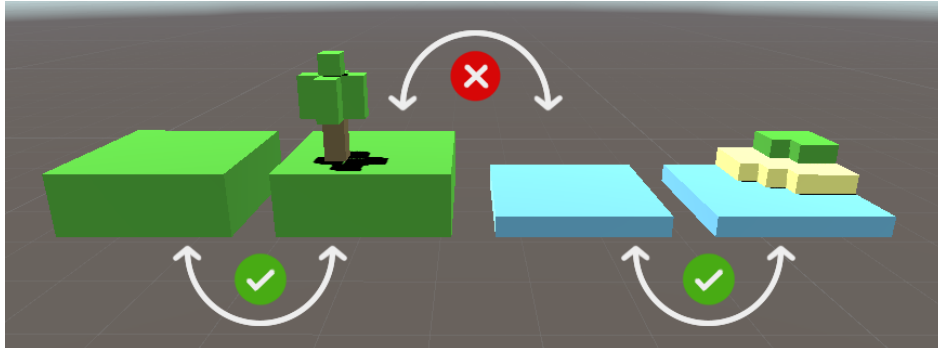


Рис. 2. Приклад розстановки тайлів

До етапів методу також входить поворот тайлів: під час оброблення необхідно розглядати чотири сторони тайлу для коректної побудови мапи.

Якщо тайли ставляться послідовно, тобто для їх постановки не використовується жоден спеціальний алгоритм, то застосування цього методу може призводити до створення мапи із «дірками». Виникає це через неможливість встановлення жодного тайлу для вже поставлених сусідніх тайлів.

Для генерації ландшафтів без «дірок» пропонується алгоритм на основі WFC, який працюватиме не з пікселями зображень, а з наданими 3D-тайлами. Алгоритм виконує обхід матриці можливих тайлів для генерації мапи попередньо заданого розміру.

Таким чином, запропонований алгоритмічно-програмний метод процедурної генерації воксельного ландшафту із 3D-тайлів передбачає наступні етапи.

1. Кожен елемент матриці заповнюється всіма можливими тайлами із заданого тайлсету.
2. Заповнюються передумовлені елементи матриці, наприклад поля сусіднього для генерації чанку згенерованої мапи.
3. Обирається перший тайл, він обирається випадково або для його вибору використовується спеціальний алгоритм.
4. Ітераційний процес розстановки тайлів у матрицю (рис. 3):
  - 4.1. Генеруються можливі варіанти сусідніх тайлів із глибиною  $N$ .
  - 4.2. Обирається елемент матриці із найменшою кількістю можливих для встановлення тайлів.
    - 4.2.1. Якщо сусідні тайли не дозволяють встановити жоден тайл у поточний елемент матриці, тобто кількість можливих тайлів у поточному елементі матриці дорівнює нулю, то сусідні елементи матриці перераховуються.
  - 4.3. Якщо можливих для підстановки тайлів у обраному елементі матриці більше ніж один, то фінальний тайл обирається

випадково, або для його вибору використовується спеціальний алгоритм.

- 4.4. Із сусідів обраного елемента матриці видаляються всі неможливі для постановки тайли.
- 4.5. Якщо у всіх елементах матриці знаходиться по одному елементу, ітераційний процес завершується та матриця вважається згенерованою.

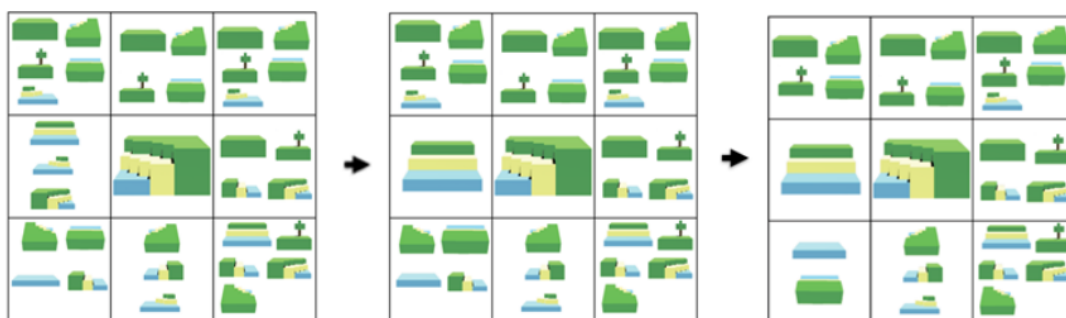


Рис. 3. Приклад генерації сусідніх тайлів

Ітераційний процес продовжується, доки у всіх елементах матриці не залишиться по одному тайлу (рис. 4).

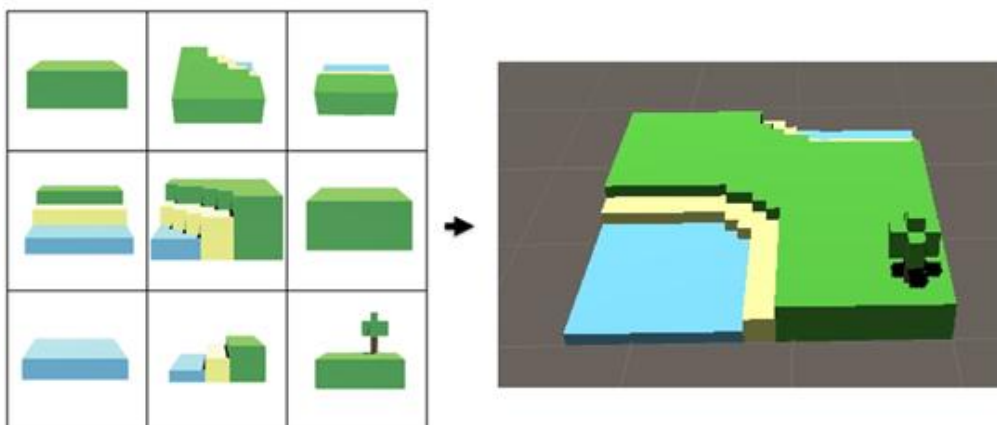


Рис. 4. Результат роботи алгоритму

Використовуючи запропонований метод, можна генерувати нескінченний воксельний ландшафт, якщо при генерації поточного чанку додати до границь матриці можливих тайлів границі сусідніх чанків.

Також слід зауважити, що генерація у запропонованому методі можлива не лише по горизонтальній площині. Запропонований метод дозволяє порівнювати всі шість сторін 3D-тайлу для генерації мапи.

## Висновки

Після імплементації алгоритму WFC для генерації воксельного ландшафту вірогідність некоректної генерації мапи істотно знизилась. Тестування алгоритмів проводилось із використанням попереднього тайлсету (рис. 1), розмір генерованих чанків дорівнював 8, та генерація відбувалася із врахуванням попередньо згенерованих чанків, тобто при генерації наступного чанку враховувались границі сусідній чанків.

Внаслідок застосування WFC середній час генерації воксельних ландшафтів зростає (табл. 1) у порівнянні з базовим методом генерації (без імплементації WFC-алгоритму). Натомість, запропонований метод забезпечує меншу вірогідність того, що мапа згенерується невірно.

Таблиця 1

Оцінка часової ефективності методу

Назва методу	Базовий метод (без застосування WFC)	Запропонований метод із застосуванням WFC
Середній час генерації чанку, мс	3.922	24.364

У подальшому доцільно провести тестування із використанням різних варіантів вибору першого елемента матриці можливих тайлів та провести тестування алгоритму із глибиною WFC-алгоритму, більшою за одиницю, на різних тайлсетах, щоб отримати результати швидкодії запропонованого методу на різних розмірах генерованої мапи та тайлсетах. Також у подальшому доцільно додати до запропонованого методу генерації етап оброблення карт висот.

## Література

1. Voxel (3D computer graphics): [Електронний ресурс]. Режим доступу: <https://en.wikipedia.org/wiki/Voxel>
2. Wave Function Collapse : [Електронний ресурс]. Режим доступу: <https://github.com/mxgmn/WaveFunctionCollapse>
3. Realtime Procedural Terrain Generation / J. Olsen // Department of Mathematics And Computer Science (IMADA) University of Southern Denmark – 2004: [Електронний ресурс]. Режим доступу: <http://web.mit.edu/cesium/Public/terrain.pdf>



**К.т.н., доцент Заболотня Т.М., магістрантка Корунська А.М.**

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

## **МЕТОД АВТОМАТИЗОВАНОЇ ВЕРИФІКАЦІЇ ПРОГРАМНОГО КОДУ СМАРТ-КОНТРАКТІВ**

### **Abstract**

**Tetiana Zabolotnia, assoc. prof., PhD, Anna Korunska, student**  
*Smart contracts software code automatized verification method*

*This paper concerns the task of analyzing software code of the smart contracts. As smart-contracts become often used and widespread, the task of their analysis becomes a bigger challenge, since there is no way to change smart contracts logic after deployment, which can lead to hacks and massive money loss. This article focuses on using Slither, an open-source static analyzer as a basis for implementing a modified method allowing verifying formal requirements. Steps of modified method are described and bounded model checking approach is justified. The ways of method evaluation and further research are proposed as well.*

### **Вступ**

Технологія блокчейн стала широко розповсюдженою та привернула до себе увагу в контексті криптовалют, проте, згодом вона отримала подальше застосування в інших галузях. Підвищена увага до даної технології пов'язана з перевагами підходу до організації даних, який в ній використовується. Згідно нього, кінцевий стан бази даних певної системи або мережі є криптографічно захищеним від несанкціонованих змін. Таким чином, в системах з великою кількістю незалежних сторін блокчейн дозволив організувати безпечну синхронізацію даних та, з деякими допущеннями, їхню незмінність.

Безумовно, однією з найбільш широко використовуваних можливостей блокчейн-систем є підтримка ними функціонування смарт-контрактів. Після публікації в блокчейн-мережі смарт-контракт стає самостійною комп'ютерною програмою, яка автоматично виконується за настання певних умов. У блокчейн-системі смарт-контракти дозволяють програмному коду працювати без будь-якої цензури, шахрайства або втручання третіх сторін. За допомогою використання смарт-контрактів

доцільно організовувати обмін цифровими цінностями, наприклад, даними про гроші, рахунки, акції, права власності тощо.

Така функціональність є доступною через максимальне унеможливлення внесення змін у записи бази даних після їх публікації; потім усунення проблем займає багато часу, є дорогим, а часто – дійсно, повністю неможливим.

Перевага незмінюваності коду смарт-контрактів має і зворотній бік, адже у разі припущення розробниками помилки в коді, її неможна виправити після публікації, а, отже, зловмисники можуть скористатися цією вразливістю і призвести до непрацездатності смарт-контракту або ж викрасти з нього певну кількість активів, що там зберігалась. Так, досить показовою ситуацією атаки на смарт-контракт є атака на The DAO, проведена в 2016 році, під час якої було втрачено близько 60 млн. доларів. Тоді смарт-контракт The DAO багато разів перевірявся експертами блокчейн-галузі на безпечність та відповідність вимогам, але, на жаль, цього виявилось недостатньо.

Отже, як бачимо, перевірка програмного коду смарт-контрактів вручну людьми не гарантує коректності. Тому, безсумнівно, має сенс приділення уваги питанню розроблення автоматизованих інструментів контролю якості і відповідності програмного коду контрактів бізнес-вимогам та вимогам безпеки.

## **Постановка задачі**

Метою даного дослідження є розроблення методу автоматизованої верифікації відповідності програмного коду смарт-контракту переліку формалізованих вимог, сформованих розробником, що дозволить пришвидшити процес розроблення та тестування смарт-контрактів та підвищити надійність та відмовостійкість їхнього програмного коду.

## **Термінологія**

Блокчейн – розподілена база даних, що зберігає впорядкований ланцюжок записів (так званих блоків), що постійно довшає; дані захищено від підробки та спотворення, а кожен блок містить часову позначку, хеш попереднього блока та дані транзакцій, подані як хеш-дерево [1].

Ethereum – відкрита, загальнодоступна розподілена обчислювальна платформа на основі технології блокчейн з підтримкою роботи смарт-контрактів [2].

Смарт-контракти – згода щодо перерозподілу цінностей між контрагентами, яка передбачає суворе і однозначне задання умов,

автоматизацію процесів виконання і мінімізацію включення довірених сторін [3].

Статичний аналіз – аналіз програмного коду, який здійснюють без реального виконання досліджуваних програм.

Динамічний аналіз – аналіз програмного коду, який здійснюють під час виконання досліджуваних програм.

Символьна верифікація – вид автономної верифікації, який перевіряє ПЗ шляхом перевірки його стану в різних точках виконання.

Формальна верифікація – доказ відповідності або невідповідності системи певній формальній специфікації або характеристиці, що використовує формальні математичні методи. Дана верифікація може довести, що система, алгоритм або програма не містить помилок і дефектів та має певні властивості.

### **Методи аналізу програмного коду смарт-контрактів**

Зазначимо, що поточне дослідження фокусується на смарт-контрактах для платформи Ethereum, реалізованих мовою Solidity, оскільки саме вони є найбільш розповсюдженими для формування блокчейн-частини розподілених додатків (dapps).

Існує багато методів аналізу програмного коду, котрі варіюються від динамічного до статичного і від ручного до автоматичного.

Серед великої кількості відомих інструментів аналізу смарт-контрактів, що реалізують різні підходи, авторами було виділено інструмент Slither, який використовує методи статичного аналізу. Інструмент дозволяє проводити перевірки за допомогою вже реалізованих модулів аналізу, які можуть надати інформацію про розповсюджені помилки або можливі проблемні місця коду, що можуть містити вразливості.

Slither приймає на вхід абстрактне дерево синтаксису (AST) смарт-контракту, сформоване компілятором Solidity з вихідного коду контракту. Робота інструменту складається з трьох основних етапів [5].

На першому етапі Slither відновлює зі скомпільованих файлів важливу для аналізу інформацію, таку як граф наслідування контракту, граф потоку керування (CFG) та повний перелік команд.

На другому етапі весь код смарт-контракту конвертується у SlithIR, внутрішню мову представлення. SlithIR використовує єдину статичну оцінку (SSA) для полегшення обчислення різноманітних метрик коду.

Під час третього етапу, фактичного аналізу коду, Slither проводить набір визначених перевірок, які виконуються заздалегідь описаними модулями аналізу. Вони перевіряють такі сценарії небажаної поведінки, як

переприсвоєння константних значень, потенційно небезпечні рекурсивні виклики тощо.

Запропонована у статті модифікація полягає у заміні третього кроку роботи інструменту статичного аналізу Slither на метод, що дозволить проведення формального аналізу програмного коду досліджуваних смарт-контрактів.

### **Модифікований метод аналізу програмного коду смарт-контрактів на основі використання проміжного представлення SlithIR**

Гіпотеза даної модифікації полягає в тому, що мову проміжного представлення SlithIR, а також граф виконання смарт-контракту, що будується на його основі, можна використати для верифікації формально описаних користувачем вимог за допомогою методу обмеженої перевірки моделі (bounded model checking, BMC).

Метод обмеженої перевірки моделі – це підвид методів символної верифікації, основна ідея якого полягає в обході графа станів програми за  $k$  кроків таким чином, щоб виявити неприпустимий стан, який вказував би на порушення виконання формальної вимоги, заданої користувачем [4].

На основі отриманих результатів дослідження різноманітних підходів, інструментів та методів аналізу програмного коду смарт-контрактів авторами сформульовано узагальнений метод аналізу програмного коду, що дозволяє проводити перевірку формальних тверджень, заданих користувачем. Метод передбачає такі кроки:

1. Переведення бізнес-вимог до програмного коду контракту у формальне подання. На даному кроці розробнику необхідно сформулювати одну чи декілька вимог до смарт-контракту формальною математичною мовою.
2. Конвертація програмного коду смарт-контракту у проміжне представлення SlithIR. На вхід даного кроку подається результат компіляції коду смарт-контрактів, після чого на основі графа виконання кожен вираз мови Solidity переводиться в набір інструкцій у спеціальному форматі подання, зручному для аналізу.
3. Аналіз графу потоку виконання на можливість переходу у стани, що не відповідають заданим формальним вимогам, отриманим після п.1.
4. У разі виявлення недопустимих станів – генерування звіту про послідовність кроків, яка призвела до порушення вимоги. Такий звіт міститиме дані про порушену формальну вимогу, а також послідовність кроків, яка призвела до порушення.

Запропонована модифікація дозволить розробникам на ранніх етапах розробки сформулювати формальні вимоги до програмного коду і бути

впевненими у відповідності коду смарт-контракту цим вимогам при будь-яких сценаріях виконання.

Недоліками даного методу є те, що метод обмеженої перевірки покладається на кількість циклів  $k$ , що задається користувачем, а, отже, перевірка може бути неповною при низьких значеннях  $k$ . Крім того, задача формальної верифікації в цілому має невирішену проблему експоненційного зростання кількості станів системи (представленого у методі графом виконання) відомого як *State explosion problem*, тому може не підійти для ефективної верифікації великих систем.

## Висновки

У статті запропоновано модифікований метод статичного аналізу смарт-контрактів на основі проміжного представлення *SlithIR*, що дозволяє виконувати перевірку формально заданих розробником умов. Такий підхід дозволяє напряму верифікувати відповідність коду смарт-контракту бізнес-логіці і мінімізувати ризики публікації вразливого програмного коду.

Даний метод дозволить розробникам вказувати певні бізнес-вимоги як формальні твердження і перевіряти, чи відповідає смарт-контракт поставленим вимогам. Це дозволить на ранніх етапах розроблення виявляти проблеми і уникати публікації у блокчейн смарт-контрактів, які є несправними, вразливими до атак зловмисників тощо.

У подальшій роботі у даному напрямку є доцільним проведення порівняльного аналізу ефективності модифікованого методу, а саме: дослідження точності верифікації шляхом перевірки кількості невірно оцінених формальних тверджень.

## Література

1. Andreas M. Antonopoulos. 2014. *Mastering Bitcoin: Unlocking Digital Crypto-Currencies* (1st. ed.). O'Reilly Media, Inc.
2. Antonopoulos, A.M. and Wood, G. and Wood, G. 2018, *Mastering Ethereum: Building Smart Contracts and DApps*. O'Reilly Media, Inc.
3. Кравченко П. Блокчейн і децентралізовані системи : навч. посібник у 3 ч. Ч. 1 / П. Кравченко, Б. Скрябін, О. Дубініна. – Харків : ПРОМАРТ, 2019. – 452 с.
4. Clarke, E.; Biere, A.; Raimi, R.; Zhu, Y. (2001). "Bounded Model Checking Using Satisfiability Solving". *Formal Methods in System Design*. 19: 7–34. doi:10.1023/A:1011276507260.
5. Alex Groce, Josselin Feist, and Gustavo Greico. "Slither: A Static Analysis Framework For Smart Contracts," 2019. doi:10.1109/wetseb.2019.00008.

УДК 004.023

К.т.н., доцент Заболотня Т.М., студент Лук'янець М.О

Національний технічний університет України  
«Київський політехнічний інститут ім. Ігоря Сікорського»

## **АНАЛІЗ МЕТОДІВ ПОПЕРЕДНЬОГО ОБРОБЛЕННЯ ТЕКСТОВИХ ДАНИХ ЩОДО ЇХ ВИКОРИСТАННЯ У ВИЗНАЧЕННІ ТОНАЛЬНОСТІ УКРАЇНОМОВНИХ ТЕКСТІВ**

### **Abstract**

**Tetiana Zabolotnia, assoc. prof.; Mykhailo Lukianets, student**  
**Analysis of methods of preprocessing of text data regarding their use in**  
**determination of tonality of ukrainian language texts**

*This work is devoted to the text preprocessing for the future use in the sentiment analysis of ukrainian text. The main goal is to make a method of preprocessing that would allow to achieve the best speed and accuracy in the sentiment analysis. The methods of preprocessing that were analyzed in this article are lemmatization, stemming and tokenization.*

### **Вступ**

Постійно зростаюча кількість текстів, складених українською мовою, які з'являються щодня в мережі Інтернет, зумовлює необхідність вдосконалення існуючого та розроблення нового програмного інструментарію для підтримки автоматизованого аналізу тональності україномовних текстових даних. Такого роду програмне забезпечення може бути використане для відстеження настроїв покупців щодо певних продуктів, швидкого реагування на інциденти тощо.

Оскільки кількість текстів є надзвичайно великою, то вкрай важливою є швидкість їх оброблення. Разом з тим, необхідно не втратити точність отримуваних результатів. Відповідно, постає питання, які поєднання методів оброблення тексту можуть бути достатньо ефективними як з боку швидкості, так і з боку точності. Одним із можливих напрямків пошуку розв'язку даного питання є дослідження шляхів оптимізації етапу попереднього оброблення тексту, адже практично всі задачі щодо аналізу текстових даних починаються саме з їх препроцесингу.

Для попереднього оброблення текстів, зазвичай, використовуються

такі методи як стемінг, лематизація та токенізація тексту. Існує чимало різних способів їх реалізації, і тим паче, комбінацій цих реалізацій. В ході даного дослідження не було виявлено у вільному доступі реалізованих методів препроцесингу текстових даних для української мови.

Тому вивчення можливих шляхів реалізації методів попереднього оброблення текстових даних, складених українською мовою, з точки зору їх подальшого використання для вирішення питань аналізу тональності тексту є актуальною задачею.

### **Постановка задачі**

Метою даного дослідження є оптимізація процесу аналізу тональності україномовних текстів за критеріями швидкості та точності шляхом знаходження ефективної комбінації методів попереднього оброблення текстових даних.

Було поставлено такі задачі для досягнення вищенаведеної мети:

- Дослідити існуючі методи попереднього оброблення текстових даних та особливості їх використання для україномовного тексту.
- Обґрунтовано вибрати метод для подальшої оптимізації.
- Сформулювати метод попереднього оброблення текстових даних як початкового етапу подальшого аналізу їхньої тональності.
- Програмно реалізувати метод аналізу тональності тексту, складеного українською мовою.

### **Особливості будови словника української мови**

За генеалогічною класифікацією українська мова належить до східнослов'янської підгрупи слов'янської групи індоєвропейської мовної сім'ї, що зумовлює певні її особливості [1].

З точки зору оцінювання тональності найбільш важливими її характеристиками є флективність та синтетичність. Велика кількість слів в українській мові утворюється завдяки додаванню афіксів. Афіксами є префікси, суфікси та закінчення. Дані афікси можуть доповнювати забарвлення слова або ж змінювати його на протилежне.

Українська мова утворювалася довгий час під впливом великої кількості факторів та продовжує змінюватися. Через це існує велика кількість як самих слів (близько 250 тисяч), так і складників слів, таких як вище наведені афікси.

Таким чином, можна зробити висновок, що через широке

використання афіксів в українській мові, варіативність слів зі схожим значенням, але різними зміненими частинами є досить великою. Відповідно для оцінки тональності слів може знадобитися їх передоброблення для подальшого аналізу, проте варто прийняти до уваги, що не всіма афіксами можна знехтувати, оскільки деякі з них можуть мати значний вплив на тональне забарвлення слів.

### **Аналіз методів передоброблення текстових даних**

#### *Токенізація*

Токенізація — це процес розмежування та, опціонально, класифікації частин рядка вхідних символів. Процес можна вважати одним з підзавдань задачі розбору вхідних даних [2]. Оскільки українська мова не значно відрізняється від, наприклад, англійської, у відношенні до створення слів (слова складаються із букв, розділові знаки не відрізняються від аналогічних в англійській та більшості європейських мов), токенізація не має певних значних особливостей, притаманних її застосуванню до текстів, складених українською.

#### *Лематизація*

Це процес групування складених форм слів з метою їх подальшого аналізу як єдиного елемента, ідентифікованого за лемою слова або словниковою формою. На відміну від стемінгу, лематизація залежить від правильного визначення частини мови та значення слова в реченні. Також, можливий аналіз у більш широкому контексті, тобто з урахуванням сусідніх речень або навіть цілого документу. Як наслідок, розроблення ефективних алгоритмів лематизації є відкритим напрямком досліджень [3].

Лематизація тісно пов'язана зі стемінгом. Різниця полягає в тому, що стемер оперує одним словом, не знаючи контексту, і тому не може розрізнити слова, які мають різні значення залежно від контексту використання слова. Однак, стемери легше реалізувати і вони працюють швидше. Знижена «точність» може не мати значення для деяких додатків. Насправді, в системах пошуку інформації стемінг покращує точність результату запиту або справжню швидкість у порівнянні з лематизацією.

Тривіальний спосіб лематизації - простий пошук у словнику. Він добре працює для простих форм, але для інших випадків, наприклад, у мовах з довгими складними словами, знадобиться система, заснована на правилах. Такі правила можна створювати вручну або вивчати автоматично з анотованого корпусу текстів [3].

Через те, що наявні набори слів української мови, промарковані за їх тональністю, є відносно невеликими, знаходження кореню слова може



бути важливим для оптимізації роботи методу аналізу тональності тексту.

Водночас, через ту ж саме причину, а саме через проблему з наявністю словників української мови, зокрема словників для лематизації у даному конкретному випадку, лематизація може не дати значного виграшу у швидкості та точності аналізу тональності тексту, що викликає певні складнощі у використанні лематизації для україномовних текстів.

### *Стемінг*

Стемінг — це процес зведення слів до їхніх кореневих форм, наприклад, обрізання групи слів зі спільною основою до цієї основи, навіть якщо вона сама не є справжнім словом у мові.

Основа не обов'язково має бути ідентичною морфологічному кореню слова; зазвичай достатньо, щоб споріднені слова відображалися в одній основі, навіть якщо ця основа сама по собі не є дійсним коренем.

Стеми створюються шляхом видалення суфіксів або префіксів, які використовуються зі словом [4].

Особливістю використання стемінгу для україномовних текстів є те, що прибирання префіксів та суфіксів може неточно або й протилежно змінити оцінку слова. Можливим рішенням може бути створення словника афіксів, що мають вплив на тональне забарвлення слова зі значенням їх впливу. Після оцінки тональності слова без афіксів можливо буде обрахувати тональність зважаючи на те, які афікси були видалені зі слова. Наприклад, префікс “анти-“ змінює значення тональності на протилежне. Разом з тим, варто звернути увагу що не всі афікси є значущими для оцінки слова, тому задача створення словників значущих афіксів є можливою.

## **Запропонований метод автоматизованого передоброблення україномовних текстових даних**

В результаті проведення вищенаведеного аналізу авторами було зроблено висновок щодо доцільності використання токенизації та стемінгу для попереднього оброблення україномовного тексту в рамках задачі визначення тональності текстових даних. Перевагу було віддано стемінгу через відсутність лематизаторів саме для української мови. На основі наведених вище відомостей можна запропонувати такий метод попереднього оброблення україномовних текстових даних як початкового етапу подальшого оцінювання їхньої тональності:

- Приведення тексту до нижнього регістру.
- Прибирання стоп-слів, у тому числі знаків пунктуації та емоджі.

- Токенізація тексту для подальшого окремого оброблення токенів.
- Стемінг токенів із запам'ятовуванням впливу прибраних афіксів на тональність токена.
- Аналіз тональності токенів, що залишилися.
- Підсумовування оцінки для загального тексту.

Використання спеціалізованих корпусів та словників дозволить реалізувати даний метод з урахуванням особливостей, притаманних українській мові.

## Висновки

Таким чином, у даному дослідженні розглянуто методи попереднього оброблення текстових даних, способи їх використання для оброблення україномовних текстів та запропоновано метод попереднього оброблення тексту, характерною особливістю якого є етап стемінгу із запам'ятовуванням видаленого афіксу та його впливу на оцінку тональності.

Аналіз особливостей української мови показав, що можна очікувати підвищення ефективності аналізу тональності текстових даних при використанні запропонованого методу попереднього оброблення за критеріями швидкодії та точності аналізу, тому що наявні в українській мові афікси мають вплив на тональне забарвлення слів та їх врахування є важливим для точної оцінки тональності.

Подальша робота над запропонованим методом має включати в себе дослідження поєднань різних методів передоброблення тексту, включно з їх різними реалізаціями для з'ясування швидкодії та точності при застосуванні як вже існуючих методів, так і запропонованого.

## Література

1. *Українська мова*— [Електронний ресурс] – Режим доступу: [https://uk.wikipedia.org/wiki/Українська\\_мова](https://uk.wikipedia.org/wiki/Українська_мова)
2. Yang, W.; Tsay, Chey-Woei; Chan, Jien-Tsai . "On the applicability of the longest-match rule in lexical analysis". *Computer Languages, Systems & Structures*. — 2002 — P. 273–288.
3. *Stemming and Lemmatization in Python*— [Електронний ресурс] – Режим доступу: [www.datacamp.com/community/tutorials/stemming-lemmatization-python](http://www.datacamp.com/community/tutorials/stemming-lemmatization-python)
4. Rajaraman, A.; Ullman, J. D. (2011). "Data Mining" (PDF). *Mining of Massive Datasets*. pp. 1–17. doi:10.1017/CBO9781139058452.002. ISBN 9781139058452.

**К.т.н, доцент Заболотня Т.М., магістрант Мединський М.С.**

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

## **ПІДСИСТЕМА ІНТЕЛЕКТУАЛЬНОГО РЕДАКТОРА ПРИРОДНОМОВНИХ ТЕКСТІВ**

### **Abstract**

**Tetiana Zabolotnia, assoc. prof., PhD; Medynski Mykola, student**  
***Intelligent editor subsystem for natural language texts processing***

*This paper is about developing software for processing natural texts in modern editor using TypeScript programming language. The problem of assisting with the creation of natural-language texts is mostly confined to spelling and syntax, and clues rarely go beyond adding a prefix to a lemma or the most frequent word-form, which is why the topic of work is relevant, the original product of the work will increase the efficiency and quality of writing natural language texts. In the course of the work, the development of language support for writing texts in natural language was further developed through the creation of affix parser, perfection of smart supplement providers, creation of the language server, and improvement of the language server protocol. Creating affixed parsers provides more accurate computation of the token's relevance by context. Improving Reasonable Provisioning Providers reduces the time spent writing a token group. Creating a dedicated language server reduces the load on the language client, upgrades the language providers without the need to update the language client, and improves the speed of language support. The language server protocol has been improved by choosing a lower level protocol as the transport protocol, which reduces the volume of the message.*

### **Вступ**

Така функція редактора текстів як інтелектуальне асистування (доповнення, підказки, діагностика) допомагає значно скоротити час написання, рецензування та редагування текстів. На даний момент майже для кожної штучної мови існує велика кількість редакторів чи повнофункціональних середовищ розроблення, оскільки для таких мов заздалегідь визначені граматики, алфавіт та словник, і вони є повністю формальними мовами (мають скінченну послідовність символів, які описуються правилами певного виду). Разом з тим, проблема асистування створенню природномовних текстів здебільшого зводиться лише до перевірки — орфографії, синтаксису та, можливо, певних стилістичних показників тексту, а підказки рідко виходять за межі доповнення префіксу до леми чи найчастотнішої словоформи. Необхідність збільшення кількості та різноманітності перевірок і під-

казок для підвищення ефективності роботи користувача, який створює чи редагує природномовний текст, і визначає актуальність обраної теми.

### **Постановка задачі**

Метою даного дослідження є вдосконалення (збільшення швидкості надання мовної підтримки, зменшення витраченого часу при написанні групи лексем, зменшення обсягу повідомлення при взаємодії з мовним сервером) інтелектуального асистування при написанні природномовних текстів через побудову підсистеми інтелектуального редактора природномовних текстів та створення мовного сервера. Для досягнення поставленої мети сформульовано наступні задачі дослідження: розглянути особливості автоматизованого оброблення природномовних текстів, існуюче інструментальне програмне забезпечення та способи його використання; розробити та реалізувати архітектуру розширюваного середовища для написання природномовних текстів; розробити архітектуру мовного сервера для написання природномовних текстів на основі LSP протоколу.

### **Термінологія**

VS Code (Visual Studio Code) — програма, редактор програмного коду.

LSP (Language server protocol) – протокол мовного сервера.

TCP (Transmission Control Protocol) – протокол передачі даних.

ASN (Abstract Syntax Notation) – мова для опису абстрактного синтаксису даних.

API (Application Programming Interface) – набір чітко визначених методів для взаємодії різних компонентів.

### **Архітектура підсистеми інтелектуального редактора природномовних текстів**

В рамках розроблення підсистеми для підтримки створення природномовних текстів було вирішено розробити:

- мовний клієнт;
- мовний сервер;
- протокол мовного сервера.

Мовний клієнт повинен використовувати протокол мовного сервера для зв'язку з мовним сервером.

Запропоновано таку внутрішню структуру мовного клієнта (рис. 1):

- модуль доповнення тексту;
- модуль діагностики тексту;
- модуль генерування підказок.

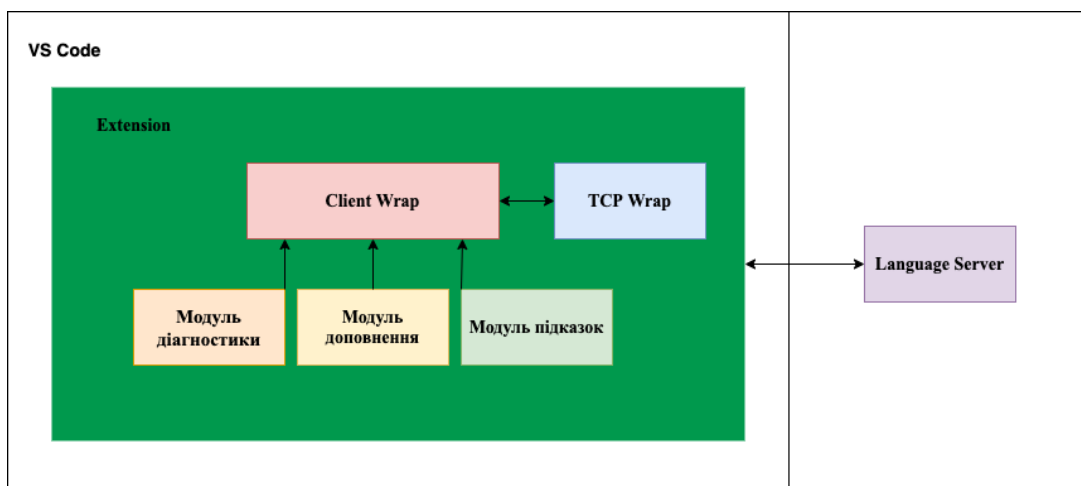


Рис 1. Архітектура мовного клієнта

Модулі діагностики тексту, доповнення тексту та генерування підказок взаємодіють з відповідним процесором документу через посилання на `vscode.TextDocument`. Процесори документів зберігаються в обгортці для роботи з клієнтом (`Client wrap`). Це необхідно для забезпечення одночасної роботи з декількома відкритими документами у Visual Studio Code.

Модуль діагностики повинен надавати користувачу інформацію про помилки вводу тексту запиту, наприклад, невірний символ. Вхідними даними для цього модуля є природномовний текст. Вихідними даними є текст повідомлення про помилку, індекси рядків та колонок початку та кінця помилки, можливі автоматичні виправлення.

Модуль доповнення тексту повинен надавати список можливих варіантів продовження тексту, який був введеним користувачем, що забезпечить прискорення написання тексту. Вхідними даними тут є природномовний текст. Вихідними даними є список лексем.

Модуль генерування підказок повинен надавати опис заданої лексеми. Вхідними даними є природномовний текст, індекс лексеми, на яку наведено курсор. Вихідними даними є опис лексеми.

Мовний сервер (рис. 2) забезпечує функціональну підтримку підсистеми оброблення текстів. Завдяки запропонованій архітектурі зменшується навантаження на мовні клієнти та створюється окремий модуль, що надає однакову функціональність різним мовним клієнтам.

Мовний сервер складається з трьох збірок:

- `DocumentAnalyzer.dll` – бібліотека для підтримки аналізу та оброблення природномовних текстів;
- `Server.exe` – консольний додаток, який обмінюється повідомленнями з клієнтом та використовує `DocumentAnalyzer` для аналізу та оброблення отриманих повідомлень;
- `Base.dll` – бібліотека, яка надає базові типи для `Server.exe` та

DocumentAnalyzer. Було вирішено розробити такі базові модулі для сервера у DocumentAnalyzer.dll:

- лексичний аналізатор;
- синтаксичний аналізатор;
- адаптер для роботи з словником;
- модуль забезпечення мовної підтримки (підказки, доповнення).

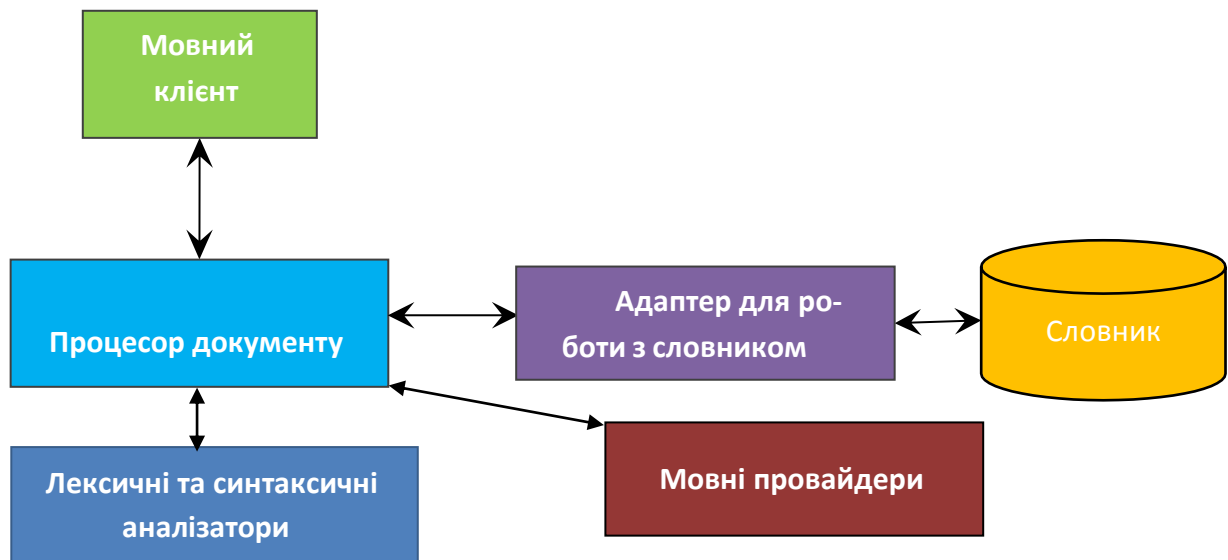


Рис 2. Архітектура мовного сервера

Server.exe складається з ID генератору для операцій, TCP Listener, ClientConnector (оброблення TCP повідомлень), ASN кодерів (ASN Encoder, ASN Decoder), eClient (API для роботи з клієнтом), TCP Message Wrap та допоміжних модулів. Протокол мовного сервера – LSP (Language Server Protocol), являє собою контракт, набір правил, який підтримується мовним сервером і використовується клієнтами для користування послугами сервера.

Єдиний контракт забезпечує стандартизацію повідомлень (рис. 3) між множиною мовних клієнтів та множиною мовних серверів. Мовним клієнтом, який є розширенням для VS Code виконується в окремому процесі спільного хосту, використовує особливості мови TypeScript, використовує події VS Code.



Рис 3. Перевага використання LSP

## Висновки

У даній статті запропоновано архітектуру програмного забезпечення для вдосконалення інтелектуального асистування при написанні природномовних текстів. Вдосконалення досягається через побудову підсистеми інтелектуального редактора природномовних текстів та створення мовного сервера, що в свою чергу надає такі переваги як збільшення швидкості надання мовної підтримки, зменшення витраченого часу при написанні групи лексем, зменшення обсягу повідомлення при взаємодії з мовним сервером при використанні визначеного LSP протоколу.

Були розглянуті особливості оброблення природномовних текстів, що вплинули на вибір протокола передачі даних та на архітектуру взагалі. Також було розглянуто існуюче інструментальне програмне забезпечення для визначення мовного клієнта VS Code.

В якості клієнта для написання природномовних текстів, що підтримує всі можливості запропонованого мовного сервера, було обрано редактор Visual Studio Code, розширення для якого має всі основні модулі для оброблення природномовних текстів.

## Література

1. Extending Visual Studio Code [Електронний ресурс], 2021— Режим доступу: <https://code.visualstudio.com/docs/extensions/overview>.
2. Yo Code — Extension Generator [Електронний ресурс], 2018— Режим доступу: <https://code.visualstudio.com/docs/extensions/yocode>.

УДК 004.912

**К.т.н., доцент Заболотня Т.М., магістрант Свинарчук М.В.**

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

## **АЛГОРИТМІЧНО-ПРОГРАМНИЙ МЕТОД ПОШУКУ ЗАПОЗИЧЕНЬ У ПРОГРАМНОМУ КОДІ**

### **Abstract**

**Tetiana Zabolotnia, assoc. prof., PhD; Maksym Svnarchuk, student**  
*Algorithmic and programmatic method of finding plagiarism in the source code*

*The article describes the existing approaches of finding plagiarism in the source code. The combined method of finding plagiarism in the source code is proposed, and its key stages are considered. The resulting method combines the advantages of each of the described approaches, retains the ability to work with a database of source codes. At the same time, the main part of the method has very good parallelism, which reduces the execution time in the presence of computational power.*

### **Вступ**

Плагіат програмного коду – це несанкціоноване повторне використання програмної структури та її реалізації певною мовою програмування [1]. Об'єктивно визначити запозичення програмного коду вкрай складно. Тому при перевірці необхідно застосовувати зовсім інші інструменти – відмінні від тих, які використовуються для виявлення текстового плагіату.

Існуючі рішення для виявлення плагіату програмного коду не здатні гарантувати виявлення наявності або відсутності запозичень, можуть лише вказати загальний відсоток подібного коду та виокремити його для перевірки користувачем. Реалізовані методи для аналізу подібності програмного коду у своїй основі намагаються виділити одну конкретну характеристику коду (кількість атрибутів, структурне представлення програми, представлення у вигляді графу тощо) та використовувати її для аналізу. На практиці це дає непогані результати, але налаштованість лише на одну характеристику коду дозволяє знайти спосіб приховування плагіату для кожного з методів окремо. Це створює проблему, коли використання одного підходу для аналізу може давати як добрі, так і незадовільні результати.



### **Постановка задачі**

Мета дослідження – покращити точність пошуку запозичень у програмному коді шляхом розроблення комбінованого методу пошуку плагіату.

Відповідно до даної мети в роботі поставлено та розв’язано такі задачі: аналіз існуючих підходів для виявлення запозичень програмного коду; вибір найбільш ефективних за критеріями часу та точності підходів та алгоритмів виявлення плагіату в програмному коді; розроблення комбінованого методу пошуку запозичень у програмному коді; забезпечення можливості роботи методу з базою даних програмних кодів.

### **Класифікація існуючих підходів**

Методи для виявлення запозичень в програмному коді базуються на різних підходах, які прийнято класифікувати наступним чином [2]:

- **Виявлення запозичень методом текстового порівняння** – використовується для виявлення звичайного текстового плагіату. Документи порівнюються дослівно з можливим перекриттям тексту. Головний недолік – неефективність у виявленні запозичень програмного коду, адже мінімальна модифікація коду, що не впливає на результат роботи програми, забезпечує її оригінальність.
- **Виявлення запозичень на основі атрибутів коду** – схожість коду визначається на основі порівняння атрибутів. Вважається, що подібні коди програм мають аналогічні показники. Головний недолік – підхід ефективний тоді, коли використовуються близькі копії програм та вразливий до додавання/видалення великих частин коду.
- **Виявлення запозичень на основі структури коду** – схожість коду визначається на основі аналізу його структури. Код програми обробляється з виділенням токенів та ігноруванням тих властивостей, що легко модифікуються (порожні символи, коментарі, назви ідентифікаторів). Головним недоліком даного підходу є залежність процесу токенизації від конкретної мови програмування.

### **Комбінований метод пошуку запозичень у програмному коді**

Метод, що пропонується, передбачає комплексний аналіз програмного коду та базується на поєднанні трьох описаних вище підходів. Комбінований метод складається з наступних етапів (рис. 1):

#### **1. Етап 1. Нормалізація програмного коду.**

Буквені символи програмного коду переводяться в нижній регістр, символи порожнього простору (включаючи пробіл, табуляцію, прогін

сторінки, символ нового рядка) замінюються одиничним пробільним символом.

На етапі нормалізації програмний код розбивається на дві частини:

- Коментарі в нормалізованому форматі, які використовуються на етапі аналізу коментарів.
- Програмний код в нормалізованому форматі з видаленими коментарями, який далі буде використовуватися на етапі токенизації.

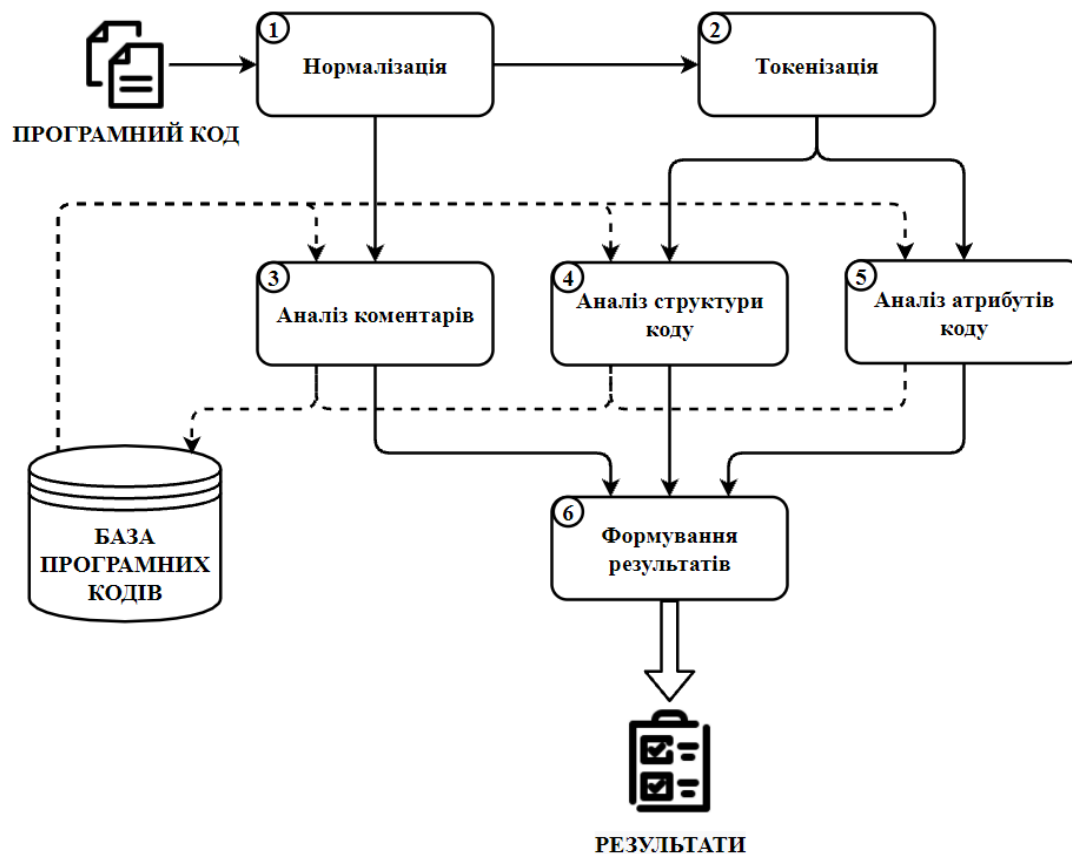


Рис. 1. Схема комбінованого методу пошуку запозичень у програмному кодї

## 2. Етап 2. Токенізація програмного коду.

Ґрунтується на регулярних виразах для виділення та заміни наступних видів токенів: ідентифікатори (назви змінних), ключові слова, літерали (константи), оператори (знаки операцій), роздільники (пунктуаційні знаки) – знаки пунктуації та парні роздільники (наприклад, фігурні дужки).

На вхід етапу токенизації надходить нормалізований програмний код без коментарів. Токенізований формат вхідних даних використовується на етапі аналізу структури коду. Також під час процесу токенизації, а саме заміни операторів, окремо збираються статистика їх використання в програмному кодї, яка потім використовується на етапі аналізу атрибутів коду.

### **3. Етап 3. Аналіз коментарів.**

В основі даного етапу лежить підхід виявлення запозичень методом текстового порівняння. Для виділення коментарів (як простих, так і багаторядкових) з вихідного коду програми застосовується функціональність регулярних виразів на етапі нормалізації. Для пошуку запозичень серед коментарів використовується алгоритм просіювання [3].

Відсутність збігів не є гарантом унікальності програм, оскільки коментарі піддаються легкій модифікації або повного видалення. Однак, досить високий відсоток їх схожості практично точно свідчить про наявність плагіату в програмах.

### **4. Етап 4. Аналіз структури коду.**

Даний етап являє собою реалізацію підходу виявлення запозичень на основі аналізу структури коду. Оброблюється програмний код в токенизованому форматі. Для забезпечення можливості роботи методу з базою даних програмних кодів використовується алгоритм просіювання [3].

### **5. Етап 5. Аналіз атрибутів коду.**

Етап реалізує підхід пошуку запозичень на основі аналізу атрибутів коду.

Кожна програма має певну структуру даних, яка може бути використана в якості однієї з характеристик програмного коду. Для доведення факту запозичення ця характеристика повинна слабко змінюватися у випадку модифікації програмного коду або включення частин однієї програми в іншу. Для аналізу атрибутів коду пропонується використати наступні характеристики:

- Частота появи операторів – кількість використань конкретного оператора в коді, поділена на кількість всіх операторів. При незначній зміні коду можуть бути видалені або додані нові оператори, але загальні зміни будуть незначні. Високий відсоток збігів двох різних програм не завжди є фактом запозичення програмного коду, але служить стабільним індикатором для подальшої перевірки.
- Взаємна кореляція двох послідовностей операторів (рис. 1). Перший оператор однієї послідовності порівнюється з першим оператором іншої, другий – з другим, третій – з третім і так до тих пір, поки не будуть перебрані всі оператори. Якщо довжини послідовностей різні, порівняння відбувається за довжиною коротшої послідовності. Якщо виявлено збіг – лічильник збільшується на одиницю. Потім зберігається значення лічильника і коротша послідовність зсувається так, щоб її перший оператор порівнювався з другим іншої послідовності, другий – з третім і так далі. І знову збільшуємо значення лічильника, якщо знайдено збіг.

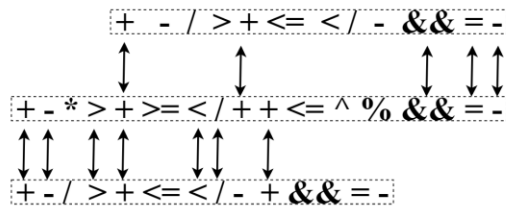


Рис. 2. Схема аналізу взаємної кореляції

Модифікація даних характеристик вимагає глибокого розуміння логіки функціонування програми та є дуже трудомістким процесом. Для забезпечення можливості роботи методу з базою даних програмних кодів на даному етапі пропонується зберігати описані вище характеристики в базі даних.

## 6. Етап 6. Формування результатів програмного коду.

Формування результатів виявлення запозичень програмного коду передбачає:

- Отримання результатів аналізу запозичень по кожній з характеристик окремо (коментарі, атрибути коду, структурний аналіз).
- Можливість задання порогових значень процентного збігу вихідних текстів.
- Виділення ділянок програмного коду, підозрілого на збіг.

Варто зазначити, що кожний з етапів аналізу (етапи 3, 4 та 5) застосовується не до всього програмного коду, а лише до виділених на етапах попереднього оброблення його частин. Це дозволяє частково позбутися недоліків кожного з підходів до аналізу запозичень та зосередитися на їхніх перевагах.

Оскільки запропонований метод включає три етапи аналізу різних частин програмного коду, то виявлення запозичень займе більше часу, ніж окреме виконання одного типу аналізу. При цьому етапи аналізу не залежать один від одного та можуть виконуватися паралельно, що дозволить оптимізувати роботу методу та, за наявності обчислювальних потужностей, зменшити час виконання методу до часу виконання найдовшого з трьох етапів аналізу і звести вплив такого об'єднання до мінімуму. Дана оптимізація методу є одним з напрямків роботи щодо його реалізації. Також комбінований метод вимагає більше ресурсних затрат для роботи з базою даних, адже необхідно окремо зберігати характеристики програмного коду для кожного з етапів аналізу.

## Висновки

В даній статті виконаний аналіз існуючих підходів до виявлення запозичень у програмному коді, а саме виявлення запозичень методом текстового порівняння, на основі аналізу атрибутів коду та на основі аналізу структури коду. Запропоновано комбінований метод, який базується на всіх трьох розглянутих підходах і метою якого є підвищення точності виявлення запозичень в програмному коді та забезпечення можливості роботи з базою даних програмних кодів.

Подальша робота над даною тематикою може бути зосереджена у двох напрямках: дослідження альтернативних способів аналізу атрибутів коду для підвищення точності роботи 5 етапу комбінованого методу; дослідження конкурентних алгоритмів до алгоритму просіювання, які підвищують точність виявлення запозичень на 3 та 4 етапах та одночасно дозволять ефективно працювати з базою даних програмних кодів.

## Література

1. Victor Ciesielski, Nelson Wu, Seyed M. M. Tahaghoghi Evolving similarity functions for code plagiarism detection. GECCO 2008: 1453-1460.
2. Tresnawati, D. Plagiarism Detection System Design for Programming Assignment in Virtual Classroom Based on Moodle [Text] / Dewi Tresnawati, Arief Syaichu Ra, Kuspriyanto // Procedia - Social and Behavioral Sciences. – 2012. – Vol. 67 – P. 114-122 – ISSN: 1877-0428.
3. Schleimer, S. Winnowing: local algorithms for document fingerprinting [Text] / Saul David Schleimer, Daniel S. Wilkerson, Alex Aiken // Proceedings of the ACM SIGMOD international conference on Management of data. – 2003. – P. 76-85 – ISBN: 978-1-58113-634-0.
4. Chen, X. Shared information and program plagiarism detection [Text] / Xin Chen, Brent Francica, Ming Li, Brian McKinnon, Amit Seker. // IEEE Transactions on Information Theory. – 2004. – Vol. 50, № 7. – P. 1545-1551 – ISSN: 0018-9448.

Магістрант Згуровський Я.Ю., д.т.н., доцент Сулема Є.С.

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

## ПРОГРАМНА БІБЛІОТЕКА ДЛЯ РЕАЛІЗАЦІЇ ПРОЦЕДУРОБРОБЛЕННЯ ТЕМПОРАЛЬНИХ МУЛЬТИМОДАЛЬНИХ ДАНИХ

### Abstract

Yaroslav Zghurovskyi, Master student; Yevgeniya Sulema, DSc, Assoc. Prof.  
*Software library for temporal multimodal data processing procedures  
implementation*

*This paper concerns the task of implementation of procedures for processing temporal multimodal data. The basic operations of the Algebraic System of Aggregates for which the library is implemented were defined. The results of the software library implementation were presented. The steps and direction of further research were proposed in the paper.*

### Вступ

На сьогоднішній день цифровий двійник є ключовою концепцією для впровадження кіберфізичних систем у промислове виробництво, медицину, енергетику та інші важливі галузі життєдіяльності людини. Він являє собою віртуальний аналог фізичної сутності і може використовуватися для моделювання різних сценаріїв на основі моделей, вхідних даних та інформації, що отримується за допомогою давачів.

У віртуальному просторі аналіз даних дозволяє робити прогнози та оптимізацію моделі [1]. В багатьох випадках під час впровадження цифрових двійників є потреба у зборі даних, що були отримані з різних за своєю природою джерел та в різні часові проміжки. Наступним етапом після збору даних є синхронізація наборів даних, так званих мультимодальних даних. Даний етап є частиною процесу аналізу даних, для отримання сукупного стану об'єкта. Одним зі способів для вирішення завдання аналізу мультимодальних даних є представлення наборів даних, що описують одині той самий об'єкт за допомогою структури даних, що має назву "агрегат". Ця структура даних була запропонована в алгебраїчній системі агрегатів [2]. Ця стаття присвячена особливостям програмної реалізації операцій в алгебраїчній системі агрегатів.

## **Постановка задачі**

Задача полягає у розробленні програмного забезпечення для оброблення темпоральних мультимодальних даних на основі логічних операцій та операцій впорядкування над агрегатами, визначених в алгебраїчній системі агрегатів [2-4].

## **Термінологія**

Алгебраїчна система (алгебраїчна структура) — це об'єкт, що складається з множин ( $M, F, R$ ), що задовольняють деякій системі аксіом, де  $M$  — непорожня множина (носій), елементи якої є елементами системи;  $F$  — сукупність операцій;  $R$  — сукупність відношень.

Алгебраїчна система агрегатів (АСА) — алгебраїчна система, носієм якої є довільна множина специфічних структур — агрегатів.

Агрегат — це впорядкована скінченна сукупність елементів, яка визначається кортежем множин та кортежем кортежів елементів.

Кортеж — це впорядкований набір елементів.

## **Операції над агрегатами в АСА**

Операції над агрегатами включають операції наступних категорій: логічні, операції впорядкування та арифметичні. Програмне забезпечення було розроблене для логічних операцій та операцій впорядкування. Далі описано основні принципи реалізованих операцій:

Логічні операції [3] включають в себе:

1. Об'єднання: результатом об'єднання двох агрегатів є агрегат, який містить елементи кортежів, які належать обома агрегатам.
2. Переріз: результатом перерізу двох агрегатів є агрегат, який містить компоненти, які є спільними для цих обох агрегатів.
3. Різниця: результатом різниці агрегатів є агрегат, який містить компоненти першого агрегата, яких немає у другому агрегаті.
4. Симетрична різниця: результатом симетричної різниці агрегатів є агрегат, який містить компоненти першого агрегата, яких немає у другому агрегаті, та компоненти другого агрегата, яких немає у першому агрегаті.
5. Виключний переріз: результатом виключним перерізом двох агрегатів є агрегат, який містить лише компоненти першого агрегата, які є спільними для цих агрегатів.

До операцій впорядкування [4] належать:

1. Розміщення: дана операція перевпорядковує кортеж множин та кортеж кортежів агрегата відповідно до порядку слідування кортежу множин та кортежу кортежів шаблонного агрегата. За шаблонний агрегат може бути взятий невизначений, пустий або

будь-який довільний агрегат.

2. Сортування: (сортування за зростанням та сортування за спаданням) виконання цієї операції дозволяє отримати відсортований порядок слідування елементів (за зростанням та за спаданням відповідно) певного кортежу, який називається первісним кортежем. Результатом застосування операції сортування до агрегата є агрегат, у всіх кортежах якого елементи перевпорядковані відповідно до нового порядку слідування індексів елементів відсортованого первісного кортежу. Різновидом операцій сортування є операції сортування з доповненням (сортування за зростанням з доповненням та сортування за спаданням з доповненням), які дозволяють збільшити довжину коротших кортежів відповідно до довжини первісного кортежу шляхом додавання значення певного значення або у кінець, або у початок коротших кортежів.
3. Проріджування: дозволяє вилучити з первісного кортежу агрегата однакові значення, що розташовані поруч; одночасно вилучаються елементи з тими самими індексами у всіх інших кортежах.
4. Видалення: застосовується для вилучення певний елемент з агрегата. Різновидом операції видалення є операція умовного видалення, яка використовується для вилучення елемента в разі виконання певної умови.
5. Вставлення: застосовується для додавання певного елемента в агрегат. Різновидом операції вставлення є операція умовного вставлення, яка використовується для додавання елемента в разі виконання певної умови.

### **Бібліотека функцій для оброблення наборів темпоральних мультимодальних даних**

Бібліотека була реалізована мовою програмування Python 3.10.0.

Для програмного представлення агрегату було створено клас `Aggregate`:

```
class Aggregate:
    def __init__(self, values: dict = {}) ->
        None: self.__values = dict(values)
```

```
A1 = Aggregate({'M1': [9, 5, 2, 7, 9, 3, 6], 'M2': [2, 4, 6]})
```

Він містить у собі структуру даних *dict*, яка зберігає інформацію про дані агрегату. Словник зберігає дані у структурах ключ-значення. Ключем у даному випадку є назва певної множини агрегату. А значенням пари для конкретного ключа є структура даних *list*, у якій власне і зберігається кортеж елементів певної множини.



Обрані структури даних за своїми характеристиками дозволяють швидко отримувати доступ до кортежу елементів по назві множини та мати лише унікальні назви множин, що є перевагами для їхнього використання в контексті реалізації операцій для агрегатів.

Для демонстрації результатів реалізації бібліотеки у Лістингу 1 наведено фрагмент програмного коду мовою Python, що реалізує операцію сортування.

### Лістинг 1. Операція сортування агрегату

#### Aggregate.py

```
def sorting(self, primary_tuple_index, reverse=False):
    result = Aggregate(self.__values)
    aggregate_length = len(self.__values)

    if primary_tuple_index <= 0 or primary_tuple_index > aggregate_length:
        return result

    primary_tuple = list(self.__values.values())[primary_tuple_index - 1]
    sorted_indexes = [i[0] for i in sorted(enumerate(primary_tuple),
key=lambda x: x[1], reverse=reverse)]

    for key, value in self.__values.items():
        origin_list = list(value)
        origin_length = len(origin_list)
        updated_length = len(sorted_indexes)

        if origin_length < updated_length:
            result.__values[key] = [None for _ in range(0, updated_length)]

        for index, new_index in enumerate(sorted_indexes):
            if new_index < origin_length:
                result._assign_value(key, index, origin_list[new_index])

    return result
```

У Лістингу 2 продемонстровано результат роботи методу “sorting” на прикладі агрегату A1, що містить наступні кортежі з цілочисельними значеннями: M1, M2, M3.

## Лістинг 2. Результат виконання операції сортування агрегатів

```
demo.py
A1 = Aggregate(
    {'M1': [9, 3, 2, 7, 8, 3, 3, 3, 6],
     'M2': [2, 4, 6],
     'M3': [4, 2, 7, 6, -5]})
B = A1.sorting(3)
print(B)
M1: [8, 3, 9, 7, 2, 3, 3, 3, 6]
M2: [None, 4, 2, None, 6]
# print_results
M3: [-5, 2, 4, 6, 7]
```

У табл. 1 наведено результати вимірювання швидкодії роботи реалізованих операцій, крім вставлення, видалення та розміщення. Оскільки операції вставлення та видалення за рахунок використання словника відбуваються майже моментально, а операція розміщення потребує лише перестановки порядку кортежів, без роботи з елементами Виміри було проведено на агрегатах, що містять 4 множини цілочисельних значень, з рівними довжинами кортежів (N), де N=1000000. Для операцій з двома агрегатами використовувались сумісні агрегати.

Таблиця 1.

Результати вимірювання швидкодії реалізованих операцій\*

Операція/№ спроби	1	2	3	4	5	Середнє арифметичне
Об'єднання	0,031	0,328	0,156	0,187	0,14	0.1684
Переріз	2,938	3,012	3,137	3,205	3,296	3,1176
Різниця	1,5	1,703	1,687	1,672	1,875	1,6874
Симетрична різниця	3,469	3,282	3,235	3,375	3,203	3,3128
Виключний переріз	1,36	1,563	1,531	1,516	1,578	1,5096
Сортування	10,593	10,797	10,344	10,922	11,013	10,7338
Проріджування	2,437	2,844	2,812	2,828	2,797	2,7436

\* Результати вимірювань наведені у секундах

## Висновки

В даній статті запропоновано реалізацію програмного забезпечення для оброблення темпоральних мультимодальних даних з використанням математичного апарату АСА. Використання створеної бібліотеки дозволить обробляти дані, що були отримані з кількох джерел, та в подальшому синхронізувати їх між собою.

Подальша робота буде спрямована на вдосконалення методу створення мультиобrazу на основі АСА та його реалізація з використанням розробленої бібліотеки.

## Література

1. Rasheed A., San O., Kvamsdal T. Digital twin: Values, challenges and enablers from a modeling perspective //IEEE Access. – 2020. – Т. 8. – С.21980-22012.
2. Mathematical Methods in Interdisciplinary Sciences / Sulema Ye., Kerre E., et al.; editor Chakraverty S. Wiley, USA, 2020. 63-72 p. ISBN : 9781119585640. DOI : 10.1002/9781119585640.
3. Dychka I.A., Sulema Ye.S. Logical Operations in Algebraic System of Aggregates for Multimodal Data Representation and Processing. Наукові вісті КПІ, 2018. № 6, С. 44–52.
4. Dychka I.A., Sulema Ye.S. Ordering Operations in Algebraic System of Aggregates for Multi-Image Data Processing. Наукові вісті КПІ, 2019. № 1, С. 15–23.

УДК 004.023

Магістрант Ільїн М.О., к.т.н., доцент Заболотня Т.М.

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

## МЕТОД ВИДІЛЕННЯ ВІДПОВІДІ ІЗ КОРИСТУВАЦЬКОЇ БАЗИ ЗНАНЬ ПИТАЛЬНО- ВІДПОВІДНОЇ СИСТЕМИ ВІДДІЛУ ТЕХНІЧНОЇ ПІДТРИМКИ

### Abstract

**Ilin Maksym, student; Tetiana Zabolotnia, assoc. prof., PhD**

*The method of answer- extraction from user's knowledge base of a question-answering models of technical customer support*

*This article is devoted to BERT question answering model, metrics that are used for comparison of QA-models and method modification aimed at ensuring that chosen way of solving the problem of constructing a QA-system of technical customer support based on the basis of company's knowledge base is the most efficient. Also, in this article described ways of preprocessing company's knowledge base data, listed the criteria that records in the knowledge base must meet to achieve the most accurate question answering result.*

### Вступ

Постійне зростання обсягів розроблюваного програмного забезпечення та кількості користувачів програмних продуктів сприяє закономірному збільшенню кількості звернень до служби технічної підтримки. Забезпечення високої якості підтримки є пріоритетним завданням для технічних компаній, адже якість технічної підтримки напряму пов'язана з показником утримання користувача [1].

На сьогоднішній день клієнти мають безліч можливостей встановити зв'язок із постачальниками послуг, використовуючи телефон, електронну пошту, текстовий чат тощо. Серед згаданих шляхів взаємодії функціональні можливості саме текстового чату забезпечують найвищу ступінь оперативності, що сприяє швидкому вирішенню проблеми і не вимагає безперервності, як телефонна розмова [1]. Більше того, текстовий чат – це сам запис розмови, тому його можна легко реєструвати, індексувати та використовувати для аналізу користувацької поведінки [2]. Можливими є два варіанти реалізації текстового чату: чат із працівником служби підтримки та чат-бот, побудований без використання людського

агента. До других можна віднести чат-боти із використанням наперед продуманих сценаріїв та чат-боти з обробленням запитів за допомогою систем штучного інтелекту. Реалізація служби технічної підтримки у вигляді чат-бота є значно вигіднішою для продуктових компаній: згідно з дослідженням Стівена Макдональда, вартість людської технічної підтримки в 100 разів вища, ніж при використанні чат-ботів [3]. З цим пов'язаний активний перехід компаній від служб підтримки, де працюють реальні люди, до систем з природномовним інтерфейсом (чат-бот асистентів) [4].

Існуючі програмні продукти спрямовані на вирішення проблеми побудови питально-відповідної системи на базі таких наборів даних, як Stanford Question Answering Dataset (SQuAD) [5], Natural Questions [6] тощо.

Через орієнтованість на конкретні набори даних запропоновані рішення не є універсальними та не можуть забезпечувати достатньо високу точність генерації відповіді на запит користувача.

Отже, наразі актуальним є розроблення узагальненого методу побудови питально-відповідних систем у вигляді чат-бот асистента на основі власної інформаційної бази знань.

### **Постановка задачі**

Метою даного дослідження є вдосконалення методу побудови питально-відповідних систем на основі розпізнавання природномовних текстових даних задля забезпечення високої точності надання правильної відповіді на запитання, шляхом виділення її з користувацької бази знань.

Відповідно до даної мети в роботі поставлено такі задачі: аналіз існуючих методів побудови питально-відповідних систем; дослідження особливостей необхідних вхідних даних та роботи щодо їх збору та попереднього оброблення; удосконалення методу виділення відповіді із бази знань та визначення шляхів його подальшого розвитку.

### **Термінологія**

*ПВС (питально-відповідна система, QA-система)* – це особливий тип інформаційних систем, які є гібридом пошукових та інтелектуальних систем, наділених природномовним інтерфейсом.

*BERT (англ. Bidirectional Encoder Representations from Transformers, двоспрямовані кодувальні представлення з Трансформерів)* – це методика машинного навчання, що ґрунтується на Трансформері, для попереднього тренування моделі оброблення природної мови (ОПМ), розроблена Google.

*Оброблення природної мови (англ. Natural language processing, NLP)* – загальний напрям інформатики, штучного інтелекту та математичної

лінгвістики, що вивчає проблеми комп'ютерного аналізу та синтезу природної мови.

### **Вимоги до бази знань та попереднє оброблення даних**

Першим кроком у побудові проекту з оброблення природної мови є попередня підготовка бази знань, що буде використовуватися для навчання моделі. До даного етапу належить формування бази знань, заповнення пропущених значень, стандартизація даних шляхом їх токенізації.

З метою досягнення більш точного результату прогнозування сформульовано наступні кроки попередньої підготовки даних:

- привести усі текстові дані до нижнього регістру;
- видалити усі розділові знаки та пробільні символи;
- видалити стоп-слова.

Сам набір даних повинен бути маркованим за принципом питання-відповідь.

### **Метрики оцінювання якості моделі**

В даному дослідженні оцінювання якості моделі проводиться за допомогою F-міри. Дана метрика обчислюється через влучність та повноту тексту відповіді. У даному контексті влучність є часткою релевантних зразків серед знайдених, а повнота є часткою загального числа позитивних зразків, яку було дійсно знайдено. Оцінка F1 визначається за формулою:

$$2 * \frac{P * R}{P + R}$$

де P – показник влучності тексту, а R – показник повноти тексту.

### **Техніка оброблення природномовних текстових даних з метою виділення відповіді на запитання із корпусу тексту**

Для вирішення задачі пошуку відповіді на питання, сформульоване природною мовою, пропонується використовувати методику двоспрямованих кодувальних представлень з Трансформерів (BERT). Результати роботи моделі BERT показують, що мовна модель, яка навчається двоспрямовано, може більш глибоко аналізувати мовний контекст, ніж односпрямовані мовні моделі. F-міра такої моделі становить 80.22% [7].

BERT використовує Трансформер – механізм уваги, який вивчає контекстні відносини між словами у тексті. У своїй звичайній формі Трансформер включає два окремих модулі – кодер, який зчитує введений текст, і декодер, який відповідає за генерацію відповіді на запитання.

На відміну від односпрямованих моделей, які зчитують введений текст послідовно (зліва направо або справа наліво), кодер Трансформера зчитує всю послідовність слів одночасно, що дозволяє моделі визначати контекст слова на основі всього його оточення (ліворуч і праворуч від слова).

BERT використовує дві стратегії навчання:

- мовне моделювання за допомогою маски;
- передбачення наступного речення.

Принцип мовного моделювання за допомогою маски полягає у тому, що перед подачею послідовностей слів у BERT 15% слів у кожній послідовності замінюються маркером [MASK]. Потім модель намагається передбачити вихідне значення замаскованих слів на основі контексту, наданого іншими, незамаскованими словами в послідовності. З технічної точки зору, для передбачення вихідних слів необхідними є наступні кроки:

- додавання шару класифікації над рівнем вихідних даних кодера;
- множення вихідних векторів на матрицю вбудовування, перетворення їх у словниковий вимір;
- розрахунок ймовірності кожного слова у словнику за допомогою softmax-рівня.

Функція втрат BERT враховує лише передбачення замаскованих значень і ігнорує передбачення немаскованих слів.

Принцип передбачення наступного речення полягає в тому, що у процесі навчання модель отримує пари речень як вхідні дані та вчиться передбачати, чи є друге речення в парі наступним реченням у вихідному документі. Під час навчання 50% даних, що подаються на вхід, є парою, в якій друге речення є наступним реченням в оригінальному корпусі тексту, тоді як в інших 50% – друге речення вибирається з корпусу тексту випадково.

Щоб передбачити, чи справді друге речення пов'язане з першим, виконуються такі дії:

- вся вхідна послідовність проходить через модель Трансформера;
- вихід токена [CLS] перетворюється у вектор у формі  $2 \times 1$ , використовуючи простий шар класифікації;
- відбувається обчислення ймовірності IsNextSequence з softmax-шару.

Під час навчання система мовного моделювання за допомогою маски та система передбачення наступного речення навчаються разом з метою мінімізації комбінованої функції втрат двох стратегій. Метою мінімізації функції втрат є отримання найменшої можливої помилки та підвищення точності предиктивної моделі.

У завданнях побудови питально-відповідної системи програмне забезпечення отримує запит на текстову послідовність, що є відповіддю на поставлене питання. Використовуючи BERT, модель питально-відповідної системи формується навчанням двох додаткових векторів, які позначають початок і кінець відповіді у блоці тексту бази знань.

### **Модифікація техніки ОПМ**

Як було вказано на початку статті, модель Трансформерів використовує операцію самоуваги, щоб забезпечити значущі результати у розумінні мовного контексту. Зі збільшенням довжини послідовності обчислення різко масштабуються для цього механізму.

При використанні стандартної моделі BERT буде спостерігатися помилка, яка вказує на те, що довга послідовність введів не може бути оброблена.

Проблему довших послідовностей пропонується вирішити використанням Лонгформера [8] замість стандартних моделей на основі Трансформерів. Лонгформер перевершує більшість інших моделей у задачах з довгими документами, і забезпечує значне скорочення у використанні пам'яті та часового ресурсу.

Стандартну модель Трансформера можна переобладнати в його «довгу» версію. Для переобладнання Трансформера у Лонгформер розширимо позицію вбудовування з 512 до 4096, а також замінімо стандартні шари самоуваги на шар уваги Лонгформера.

### **Висновки**

В даній статті розглянуто основні механізми методу двоспрямованих кодувальних представлень з Трансформерів. Запропоновано модифікацію даного методу, що спрямована на збільшення довжини оброблюваних послідовностей тексту. Також описано вимоги до вхідних даних та кроки оброблення вхідних даних з метою досягнення більш точного результату прогнозування.

Подальша робота над запропонованим методом має включати в себе дослідження особливостей побудови україномовних питально-відповідних систем.

### **Література**

1. *Mercer Smith*. “Customer Service Statistics and Facts You Shouldn't Ignore” [Електронний ресурс]. Режим доступу: <https://www.helpscout.com/75-customer-service-facts-quotes-statistics/>



2. *Namee Jani*. “Chatbot Data Analytic – The Next Big Thing for Business Optimization” [Электронный ресурс]. Режим доступа: <https://segment.com/recipes/analyze-chatbot-data-for-customer-insights/>
3. *Steven MacDonald*, “Why tech support is crucial to the overall user experience” [Электронный ресурс]. Режим доступа: <https://www.superoffice.com/blog/tech-support/>
4. *Nicolas Fayon*. “How chatbots are killing jobs (and creating new ones)” [Электронный ресурс]. Режим доступа: <https://bit.ly/3EE3Cj4>
5. *Vihan Lakshman*. "Question Answering on the SQuAD Dataset". [Электронный ресурс]. Режим доступа: <https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2761899.pdf>
6. *Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield*. “Natural Questions: a Benchmark for Question Answering Research” [Электронный ресурс]. Режим доступа: <https://research.google/pubs/pub47761/>
7. “Question Answering Squad2.0 Bert Large Uncased Model Card”, [Электронный ресурс]. Режим доступа: [https://catalog.ngc.nvidia.com/orgs/nvidia/teams/tlt-riva/models/questionanswering\\_squad\\_english\\_bertlarge](https://catalog.ngc.nvidia.com/orgs/nvidia/teams/tlt-riva/models/questionanswering_squad_english_bertlarge)
8. *Iz Beltagy, Matthew E. Peters, Arman Cohan*, “Longformer: The Long-Document Transformer”, [Электронный ресурс]. Режим доступа: <https://arxiv.org/abs/2004.05150>

Магістрант Казимиров Д.М., к.т.н., доцент Заболотня Т.М.

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

## УЗАГАЛЬНЕНИЙ МЕТОД АВТОМАТИЗОВАНОГО АНОТУВАННЯ ТЕКСТОВИХ ДАНИХ З ВИКОРИСТАННЯМ НЕЙРОННОЇ МЕРЕЖІ BERT

### Abstract

**Danylo Kazymyrov, student; Tetiana Zabolotnia, assoc. prof., PhD**  
*The method of automated abstractive text summarization based on  
BERT neural network*

*This article reviews the method of automatic annotation of textual information based on the BERT neural network. The considered method is proposed to be used for solving the problem of textual data reduction to form an overview description of the content. The article presents a generalized algorithm of the method and describes the advantages of using BERT over other annotation methods.*

### Вступ

Щорічно обсяги текстових даних, доступних для використання, зростають дедалі швидше. Це призводить до того, що час, який може виділити людина на оброблення кожного джерела інформації, зменшується. Результатом цього є актуалізація проблеми автоматичного аналізу та попереднього оброблення інформації, представлення її в скороченому вигляді для економії часу користувачів.

Одним із підходів до попереднього оброблення текстової інформації є її автоматичне стиснення без втрачання початкового змісту. Розрізняють анотацію та реферат тексту. Реферат складається з найбільш значущих конструкцій оригінального тексту, а анотація може містити елементи, які перефразовано викладають зміст оригіналу, тож не присутні в ньому явно.

Як відомо, перед публікаціями, зокрема перед науковими роботами, прийнято надавати анотацію, в якій автор коротко описує зміст того, про що далі в роботі піде мова. Завдяки цьому, при пошуку джерел інформації, дослідник може зробити висновок про доцільність ознайомлення із повним текстом статті до того, як почне його читати. Однак, не всі текстові документи мають анотацію від автора або рецензентів. Однією з причин цього є очевидна необхідність ознайомлення із повним текстом публікації

перед тим, як коротко описати її зміст. З огляду на все вищезазначене, автоматизація процесу генерації анотації є актуальною задачею.

Сучасні методи створення анотацій базуються на статистичних та математичних обчисленнях або використовують нейронні мережі для оброблення природномовних даних [1]. Перевагою другого підходу є точність кінцевого результату, оскільки в такому разі комп'ютер аналізує текст так, як його б аналізувала людина: знаходить зв'язки між одиницями інформації в тексті, шукає контекст для кожної з них та на основі цього будує модель. Недоліком другого підходу є складність створення нейронної мережі та потреба в її попередньому навчанні. На сьогоднішній день однією з перспективних нейронних мереж для оброблення природної мови є мережа BERT [2], розроблена компанією Google. Після проведення аналізу існуючих рішень для анотування текстів та зважаючи на вищенаведені факти, автори вважають, що розроблення методу генерування анотації тексту на основі нейронної мережі BERT є актуальною задачею.

### **Постанова задачі**

Метою даного дослідження є покращення якості автоматично згенерованої анотації за критерієм ROUGE за рахунок створення методу анотування текстової інформації з використанням нейронної мережі BERT, який в подальшому може бути застосований для анотування текстів, створених мовами із східнослов'янської мовної групи, зокрема українською.

### **Термінологія**

Анотування текстових даних — піддомен оброблення природної мови, який займається вилученням стислого змісту із більших за обсягом текстів [3].

BERT — це методика машинного навчання для оброблення природної мови, що ґрунтується на архітектурі Трансформері, розроблена та попередньо натренована Google.

Трансформер — архітектура глибоких нейронних мереж, яка використовує механізм уваги для оцінювання та порівняння значущості кожної частини вхідних даних, переважно використовується для оброблення природномовних даних [2].

### **Узагальнений метод генерування анотації текстових даних на основі використання нейронної мережі BERT**

Загальну послідовність дій щодо генерування анотації до тексту можна розбити на наступні етапи: нормалізація вхідних даних, позбуття

від стоп-слів, побудова моделі-представлення вхідного тексту (граф, синтаксичне дерево, матриця), ранжування лексичних одиниць інформації (речень, словосполучень, лексем) на основі відношень між ними, вибірка найбільш цінних з них, оброблення результату, формування анотації [4].

Використання нейронної мережі, побудованої на основі архітектури Трансформера, дозволяє оптимізувати етап побудови моделі представлення тексту та ранжування його лексичних одиниць. Авторами пропонується покласти відповідальність за оцінювання та розбір змісту вхідної інформації на нейромережу, яка має бути попередньо натренована для вирішення цієї задачі. Слід зауважити, що сама мережа BERT вже є натренованою та має кінцеві ваги зв'язків між нейронами в своїх схованих шарах, що дозволяє вирішити проблему аналізу та розбору природної мови. Проте для вирішення задачі анотування текстів слід довчити модель (англ. *fine-tuning* — тонке налаштування) на задачах, для яких вона буде використовуватися. Для цього до вже існуючих шарів нейронів додаються нові, незалежні. Ваги зв'язків нейронів в нових шарах мають бути скорельовані таким чином, щоб нейронна мережа могла вирішувати вже прикладну задачу анотування текстових даних.

Авторами пропонується розглядати задачу анотування текстової інформації як пошук рекомендацій: на основі вхідного запиту слід сформулювати список речень, які є релевантними до нього. Запит тут — це вхідний текст, а показник релевантності — ймовірність того, що дане речення передає зміст тексту та має потрапити до анотації тексту.

Запропонований метод полягає у виконанні таких дій для кожного вхідного тексту:

1. Розбити його на речення та нормалізувати їх за допомогою пакету Spacy [5]. Цей пакет добре зарекомендував себе для вирішення проблеми первинної нормалізації тексту та підготування даних для оброблення їх нейронною мережею.
2. З отриманих речень сформулювати новий текст, назовемо його нормалізованим. З нього мають бути вилучені стоп-слова, сполучники та інші неважливі мовні конструкції.
3. Для кожного речення сформулювати кортеж, який складається із речення та нормалізованого тексту. Ці кортежі передати на вхід нейронній мережі. Таким чином, для текста із  $N$  речень слід виконати оцінку важливості для кожного речення, тобто  $N$  разів використати модель. Як результат, буде отримано оцінку кожного речення, яку можна додати до кортежу третім елементом.
4. Отримані кортежі відсортувати за оцінкою нейронної мережі, з них вибрати  $T$  перших, де  $T$  задається людиною.

5. Отриманий список кортежів відсортувати за положенням речення в початковому тексті. Критерій сортування можна змінювати, для знаходження результату з більшою оцінкою за критерієм ROUGE.

Оцінювання отриманих результатів роботи розглянутого методу, пропонується провести за допомогою метрики ROUGE [6]. Для цього для тренування нейронної мережі слід підготувати тестові дані, які містять повний текст та анотацію до нього, створену людиною, або сформовану іншим інструментом, результат роботи якого має високий показник оцінки за критерієм ROUGE.

## **Висновок**

В статті описано проблему оброблення великого обсягу текстової інформації та запропоновано метод автоматичного генерування анотації тексту із використанням нейронної мережі BERT. Цей метод дозволяє отримувати більш точну та повну анотацію у порівнянні з існуючими методами, його ключова перевага полягає в тому, що він може бути застосований до текстів, створених мовами більш складними, ніж англійська, зокрема українською. Було наведено основні етапи методу.

Напрямами подальшого дослідження даної тематики є оптимізація запропонованого методу для його застосування для текстів, складених українською мовою, та формування навчального масиву даних для різних жанрів текстів, що дозволить підвищити якість згерерованих анотацій текстів цих жанрів.

## **Література**

1. Pradeepika Verma. A Review on Text Summarization Techniques/ Pradeepika Verma, Anshul Verma. — Journal of Scientific Research, 2020. — С. 351-357.
2. Jacob Devlin. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding / Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova — North American Chapter of the Association for Computational Linguistics, 2019.
3. Juan-Manuel Torres-Moreno. Automatic Text Summarization — Wiley-ISTE, 2014 — 376 с. — ISBN: 978-1-848-21668-6.
4. Zakia Jalil. Extractive Multi-Document Summarization: A Review of Progress in the Last Decade / Zakia Jalil, Jamal Abdul Nasir, Muhammad Nasir — IEEE Access, Department of Computer Science and Software

Engineering, International Islamic University, Islamabad, Pakistan — 2021 — 19 с.

5. spaCy: Industrial-Strength Natural Language Processing — [Электронный ресурс] — Режим доступа: <https://spacy.io/>.
6. Chin-Yew Lin. ROUGE: A Package for Automatic Evaluation of Summaries // Text Summarization Branches Out. — Association for Computational Linguistics, Barcelona, Spain — 2004. — Vol. Association for Computational Linguistics. — С. 74-81.

**Магістрант Казимиrowa О.О., к.т.н., доцент Заболотня Т.М.**

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

## **ГІБРИДНИЙ ПІДХІД ДО АВТОМАТИЗОВАНОГО РЕФЕРУВАННЯ ТЕКСТОВИХ ДАНИХ**

### **Abstract**

**Olena Kazymyrowa, student; Tetiana Zabolotnia, assoc. prof., PhD**  
***Hybrid approach to automated text summarization***

*This paper describes the hybrid method of text extractive summarization. The method under consideration is proposed to be used to solve the problem of extracting meaning from text information with a given level of compression to form a text abstract. The method is based on using the results of abstracting existing abstracting methods and combining their results into a single text.*

### **Вступ**

Обсяг текстових даних, які продукує людство, в тому числі в мережі Інтернет, з кожним роком зростає з експоненціальною швидкістю. Такі масиви інформації вже майже неможливо обробляти вручну, тому знаходження потрібних та актуальних даних займає все більше часу.

Однією з проблем, які виникають при пошуку інформації, є невідповідність назви публікації її вмісту, а також нерелевантність результатів пошуку. На жаль, аби зрозуміти, що знайдений текст не задовольняє інтересам шукача, потрібно прочитати статтю та витратити на це дорогоцінний час. Ознайомлення з коротким витягом вмісту публікації допомогло б шукачу зробити висновки щодо актуальності матеріалу ще до початку читання цілого документу. Але на заваді цьому стає розповсюджена ситуація, коли скорочений зміст текстового документа відсутній.

Для вирішення цієї проблеми застосовуються засоби для автоматичного генерування змістовного витягу з документа. Розрізняють два види підходів до формування витягу: генерування анотації та генерування реферату. Суть першого підходу полягає в створенні змістовного витягу з перефразуванням або уточненням початкового тексту. В такому разі результат буде містити крім важливих речень із оригінального тексту ще й згенеровані самою програмою. Мета

реферування полягає в оцінюванні важливості мовних конструкцій в оригінальному тексті та створенні нового тексту з найбільш вагомих з них. Очевидно, що такий текст буде складатися з речень або словосполучень, які присутні в оригінальному тексті. В даному дослідженні будуть вивчатися саме методи генерування реферату.

Існуючі методи базуються на алгоритмічному підході або на використанні штучної нейронної мережі. Найбільш відомі з алгоритмічних - це TextRank, LSA, Fuzzy Logic. До основних мінусів алгоритмічного підходу відноситься висока залежність від структури тексту та його ліричності. В свою чергу, основним мінусом використання штучної нейронної мережі виступає висока залежність від тренувальних даних та висока складність розроблення нейронної мережі.

Рішенням даних проблем може бути розроблення гібридного методу автоматизованого реферування текстових даних, що використовує сильні сторони кожного підходу.

### **Постановка задачі**

Метою даного дослідження є вдосконалення якості вихідного реферату за критеріями семантичної зв'язності та повноти передачі змісту вхідних текстових даних шляхом розроблення гібридного методу реферування.

Для досягнення визначеної мети поставлені наступні завдання:

- Провести аналіз сучасних підходів, методів та алгоритмів, що застосовуються для вирішення задач автоматичного реферування текстових даних.
- Обґрунтовано обрати методи автоматичного реферування текстів та застосувати їх при розробленні гібридного методу.
- Провести аналіз існуючих способів оцінки якості вихідного реферату.
- Розробити гібридний метод автоматизованого реферування текстових даних і провести оцінку ефективності розробленого методу.
- Розробити алгоритм злиття рефератів утворених обраними методами автоматичного реферування текстів.

### **Термінологія**

Реферування тексту — піддомен ОПМ, який займається вилученням змісту із текстового документа шляхом відбору з нього важливих фрагментів для стиснення інформації [1].

ОПМ — оброблення природної мови, напрям інформатики, математичної лінгвістики та штучного інтелекту, вивчає проблему аналізу



та синтезу природної (людської) мови [1].

### **Гібридний метод автоматизованого реферування текстових даних**

Гібридний метод автоматизованого реферування текстових даних, що пропонується, представляє собою поєднання трьох методів реферування TextRank[2], LSA[3] та Fuzzy Logic[4] за допомогою алгоритму злиття, який об'єднує всі набори даних. Результати порівняльного аналізу цих методів подано у таблиці 1.

Таблиця 1

#### Порівняння методів реферування

Метод	TexRank	LSA	Fuzzy Logic
Принцип роботи	Оцінка схожості двох речень	Представлення речень як векторів в N-мірному просторі	Представлення речень як векторів значень функцій. До цих функцій відносяться: заголовок, довжина речення, вага терму, позиція речення та тематичні слова. Кожній функції надається значення між «0» та «1».
Як формується результат	Обирають речення, з найбільшою кількістю зв'язків, оминаючи дублювання	Обираються речення, в який присутні словосполучення, які часто зустрічаються в тексті	Обираються речення, які отримали найбільшу оцінку, отриману із функцій цього речення
Структура даних в основі	Зважений граф	Вектори в N-мірному просторі	Вектори функцій
Слабка сторона	Нелінійна складність обчислень для	Речення, які потрапляють в реферат можуть не	Для формування правил та функцій приналежності

	великих текстів та текстів з ліричними відступами	передавати зміст тексту, а лише містити словосполучення, яких є багато в тексті	ключових слів до теми статті, необхідно залучати експертів конкретної предметної області
Сильна сторона	В реферат потрапляють речення, на яких в реченні акцентується увага автором	Швидкість реферування	Найбільша схожість реферату з рефератом, складеним людиною

З вищезазначеного видно, що у кожного з цих методів є свої плюси та мінуси. Однією із головних характеристик цих методів є їхня здатність давати якісний результат на різних за обсягом даних. Ідея гібридного методу, що пропонується авторами, полягає в тому, що в кінцеву анотацію потрапляють слова та речення, які отримали високу середню оцінку важливості. Цю оцінку пропонується вираховувати як середнє арифметичне оцінок значущості ключового слова, які йому дав кожний з використаних методів. Проте, як зазначалося раніше, зважаючи на розмір тексту, методи можуть давати поганий результат. Для того, аби мінімізувати цей негативний вплив на результат роботи гібридного методу, пропонується застосовувати нормуючий коефіцієнт до доданка, що відповідає оцінці методом, негативний вплив якого планується мінімізувати. Таким чином, очікується, що використані три методи реферування разом дадуть якісніший та більш семантично зв'язний результуючий реферат. Розглянемо гібридний метод детальніше.

На першому етапі гібридного методу маємо отримати список ключових слів вхідного тексту. Для цього оригінальний текст подається на вхід кожному із зазначених методів реферування окремо. Отримані ключові слова передаються далі.

На другому етапі пропонується об'єднати результати з попереднього етапу в єдиний список ключових слів. Для цього використовується алгоритм злиття, описаний вище. Він відповідає за визначення загальної ваги слова на основі його оцінки кожним з методів.

На наступному етапі ключові слова упорядковуються за спаданням значень їх ваг. Із упорядкованого списку обираються  $N$  перших слів. Виокремлення ключових слів має відбуватися у відповідності до заданої ступені стиснення документа. Зазвичай 2% від усіх ключових слів

достатньо для того, щоб висловити головну ідею документа.

Заключний етап полягає у формуванні реферату. Для цього речення об'єднуються в порядку, в якому вони зустрічалися в початковому тексті. Таким чином, результат буде складатися із фрагментів оригіналу в логічному порядку.

Перевагою запропонованого гібридного методу є й те, що результат його роботи допомагає проаналізувати якість роботи використаних методів реферування. Для цього достатньо порівняти оцінку, яку отримало слово після злиття, з оцінкою, яку йому дав метод. Якість отриманого реферату можна оцінити за допомогою критерію ROUGE [5].

До недоліків запропонованого методу можна віднести високу обчислювальну та часову складність через використання обраних методів реферування разом у порівнянні із застосуванням їх по одному. Мінімізувати цей недолік можна за рахунок розподілення обчислень між кластером комп'ютерів, які будуть відповідати за запуск лише одного з методів. Таким чином досягається зменшення часу роботи гібридного методу за рахунок паралельних обчислень.

Одним із шляхів оптимізації запропонованого методу можуть слугувати введення штучних обмежень на вхідні дані, для досягнення стабільності роботи системи. Наприклад, можна заборонити використовувати тексти в жанрах, які погано піддаються реферуванню, наприклад вірші чи романи, через їх складні мовні конструкції та ліричні відступи і метафори. Іншим обмеженням може слугувати ліміт на розмір вхідного тексту. Чим більший текст, тим більше обчислень проводиться для його аналізу. Для того, аби обійти це обмеження, можна запропонувати користувачам методу дозволити реферування тексту частинами, через автоматичне розділення тексту по розділам або абзацам.

## **Висновки**

У даній статті проведено аналіз сучасних підходів до автоматизованого реферування текстових даних. На основі результатів даного аналізу запропоновано гібридний метод автоматизованого реферування, який в собі використовує злиття рефератів, згенерованих за допомогою існуючих методів автоматичного реферування. Проаналізовано та обрано методи оцінювання якості результуючого реферату. За цю оцінку приймаємо критерій ROUGE. Можемо очікувати, що запропонований гібридний метод дозволить підвищити цей показник вихідного реферату.

Подальша робота над методом, на думку авторів, полягає в оптимізації роботи методу, а саме пошуку шляхів покращення швидкості

алгоритму злиття та пошуку формули для розрахунку значень коефіцієнтів, які використовує алгоритм злиття при розрахунку середньої оцінки ключових слів.

## Література

1. Juan-Manuel Torres-Moreno. Automatic Text Summarization — Wiley-ISTE, 2014 — 376 с. — ISBN: 978-1-848-21668-6.
2. Prateek Joshi. An Introduction to Text Summarization using the TextRank Algorithm (with Python implementation) — [Електронний ресурс] — Режим доступу: <https://www.analyticsvidhya.com/blog/2018/11/introduction-text-summarization-textrank-python/>
3. Srinivas Chakravarthy. Document Summarization Using Latent Semantic Indexing — [Електронний ресурс] — Режим доступу: <https://towardsdatascience.com/document-summarization-using-latent-semantic-indexing-b747ef2d2af6>
4. Anukarsh Singh. Introduction to Fuzzy Logic and its Application to Text Summarization — [Електронний ресурс] — Режим доступу: <https://becominghuman.ai/introduction-to-fuzzy-logic-and-its-application-to-text-summarization-dee71da67bbe>
5. Chin-Yew Lin. ROUGE: A Package for Automatic Evaluation of Summaries // Text Summarization Branches Out. — Association for Computational Linguistics, Barcelona, Spain — 2004. — Vol. Association for Computational Linguistics. — С. 74-81.

УДК 519.688

Д.т.н., професор Легеза В.П., студент Атаманюк О.В.,  
к.ф.-м.н, доцент Нещадим О.М.

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

## АМПЛІТУДНО-ЧАСТОТНА ХАРАКТЕРИСТИКА ВІБРОЗАХИСНОЇ СИСТЕМИ ТА МЕТОД ВИЗНАЧЕННЯ ОПТИМАЛЬНИХ ПАРАМЕТРІВ ЇЇ КУЛЬОВОГО ГАСНИКА

### Abstract

Viktor P. Legeza, full prof., D.Sc.; Oleksii V. Atamaniuk, student  
Oleksandr M. Neshchadym, assoc. prof., PhD

*Amplitude-frequency characteristic of a vibro-protective system and method of  
determination of optimal settings for ball vibration absorber*

*This paper concerns the task of determination of equation for amplitude-frequency characteristic of a vibro-protective system. The method of determination of optimal settings for ball vibration absorber (BVA) is proposed as well. A comparative analysis of the dynamics of two systems is carried out: with and without a BVA. 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

При експлуатації висотних гнучких споруд (телевеж, радіощогл, металевих вентиляційних труб, димарів тощо) часто виникають вимушені коливання, боротьба з якими перетворюється на велику технічну проблему. Вимушені коливання висотних об'єктів виникають при їх взаємодії із повітряним потоком і здійснюються як у площині вектора повітряного потоку, так і в ортогональній напрямку вектора повітряного потоку площині. До останнього часу для розв'язання цієї проблеми використовувались динамічні гасники маятникового типу на підвісі [1]. Ситуація, що склалася, і обумовила необхідність розробки гасителів такої конструкції, яка б була компактною, надійною та безпечною і яка б дозволила зберегти маятниковий характер функціонування робочого тіла гасителя. В цих умовах найбільш ефективним є новий метод віброзахисту низькочастотних висотних споруд із використанням гасителів коткового типу [1]. Проте відсутність математично обґрунтованого методу

оптимального налаштування основних параметрів кульових гасників цього класу на власну частоту несучого тіла та відповідного програмного забезпечення, яке б дозволило його реалізувати, 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.

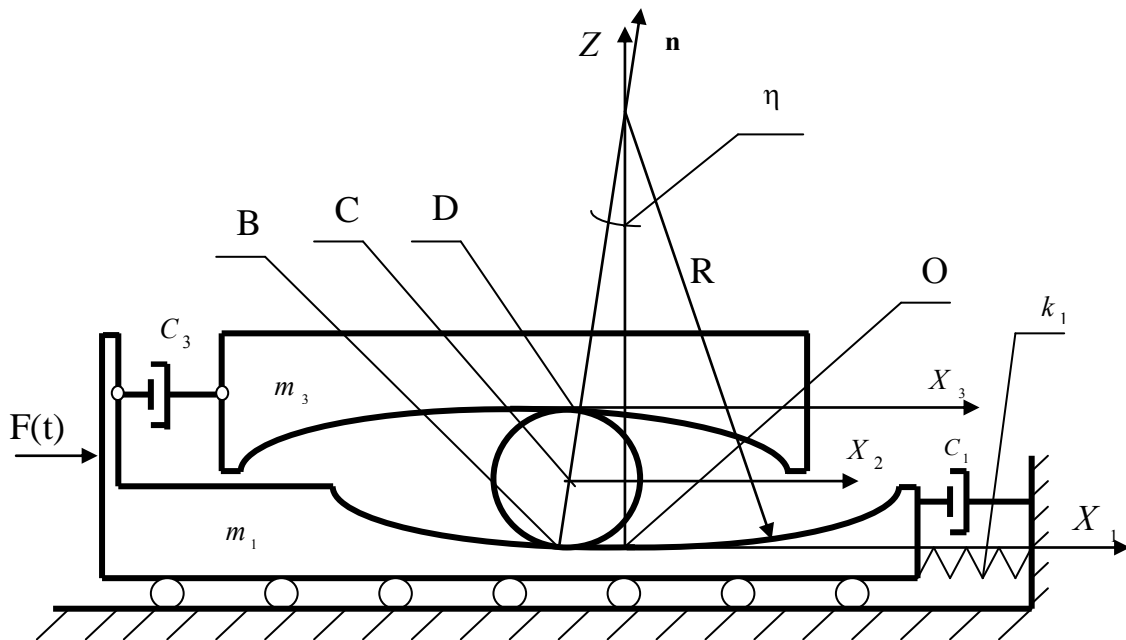


Рис. 1. Схема динамічної системи "несуче тіло - котковий гасник"

Вимушені коливання віброзахисної системи відбуваються під дією зовнішнього гармонічного збурення  $F(t) = F_0 \sin(\omega t)$ , що діє на несуче тіло масою  $m_1$ . Збурення моделює пульсацію вітрового потоку, а несуче тіло – висотну гнучку споруду, яка коливається за своєю основною (першою) формою. Ефект придушення вимушених коливань несучого тіла (споруди) в таких системах досягається за рахунок «чистого» кочення кулі з радіусом  $r$ , затиснутої між відповідними сферичними виїмками з радіусом  $R$ , симетрично розташованими на несучому тілі масою  $m_1$  та на робочому тілі масою  $m_3$ . Частина вхідної енергії безповоротно поглинає в'язкий демпфер (або система демпферів)  $c_3$ , який з'єднує робоче і несуче тіла.

Поточна робота є продовженням наукового дослідження [1], у якому було створено математичну модель процесу віброзахисту гнучких

низькочастотних об'єктів у вигляді наступної системи нелінійних диференціальних рівнянь:

$$(1 + \nu) \ddot{x}_1 + 2n_x \dot{x}_1 + \omega_0^2 x_1 - 2\nu \bar{R} \cos \eta \cdot \ddot{\eta} + 2\nu \bar{R} \sin \eta \cdot \dot{\eta}^2 = \bar{F}_0 \sin(\omega t) \quad (1)$$

$$\ddot{x}_1 \cos \eta = 2\bar{R} \ddot{\eta} + 4n_\eta R \dot{\eta} + g \cdot \sin \eta, \quad (2)$$

де  $\nu = \frac{m_3}{m_1}$ ;  $\bar{F}_0 = \frac{F_0}{m_1}$ ;  $\omega_0^2 = \frac{k_1}{m_1}$ ;  $2n_x = \frac{C_1}{m_1}$ ;  $2n_\eta = \frac{C_3}{m_3}$ ;  $m_1$  - маса несучого тіла;  $m_3$  -

маса робочого тіла;  $g$  - прискорення вільного падіння;  $F_0$  - максимальна амплітуда гармонічної збуджуючої сили;  $k_1$  - коефіцієнт пружності несучого тіла;  $\bar{R} = R - r$  - різниця радіусів сферичних виїмок і кулі, що перекочується між ними. Припускається, що маса кулі є набагато меншою порівняно із масами несучого і робочого тіл, тобто  $\frac{m_2}{m_1} \rightarrow 0$ ;  $\frac{m_2}{m_3} \rightarrow 0$ .

В роботі розглядаються малі усталені коливання системи. Вважаючи кут  $\eta$  та кутову швидкість  $\dot{\eta}$  малими (через що можна покласти  $\sin \eta \approx \eta$ ,  $\cos \eta \approx 1$ ,  $\sin \eta \cdot \dot{\eta}^2 \rightarrow 0$ ) проведемо лінеаризацію системи рівнянь (1) – (2). В результаті отримаємо систему лінійних диференціальних рівнянь відносно двох невідомих функцій  $x_1(t)$ ,  $\eta(t)$ :

$$(1 + \nu) \ddot{x}_1 + 2n_x \dot{x}_1 + \omega_0^2 x_1 - 2\nu \bar{R} \ddot{\eta} = \bar{F}_0 \sin(\omega t), \quad (3)$$

$$\ddot{x}_1 = 2\bar{R} \ddot{\eta} + 4n_\eta \bar{R} \dot{\eta} + g \eta. \quad (4)$$

Для оптимального регулювання процесу віброзахисту та мінімізації амплітуди несучого тіла потрібно спочатку встановити амплітудно-частотну характеристику (АЧХ) досліджуваної системи, а вже після цього розробити відповідну програмну процедуру налаштування (tuning) оптимальних параметрів гасника. Виходячи з цих міркувань сформулюємо мету поточного дослідження.

**Мета** дослідження – побудувати рівняння АЧХ віброзахисної системи у **лінійній постановці** та розробити спеціальний програмний комплекс, який забезпечить можливість визначення оптимальних регулюючих параметрів кульового гасника.

### Побудова рівняння АЧХ віброзахисної системи

Оскільки розглядаються усталені коливання системи, то розв'язок системи диференціальних рівнянь (3) – (4) шукатимемо у вигляді:

$$x_1 = A(\omega) \sin(\omega t - \alpha) \quad (5)$$

$$\eta = B(\omega) \sin(\omega t - \beta) \quad (6)$$

Підставимо вирази (5) та (6) у рівняння (3) та (4). Після цього отримані рівняння спочатку помножимо на  $\sin(\omega t)$ , потім на  $\cos(\omega t)$  та проінтегруємо їх в межах від 0 до  $2\pi$ . В результаті отримаємо чотири

алгебраїчно-тригонометричних рівняння відносно чотирьох невідомих  $A, B, \alpha, \beta$ :

$$[\omega_0^2 - \omega^2(1 + \nu)] A \cos \alpha + 2n_x A \omega \sin \alpha + 2\nu \bar{R} B \omega^2 \cos \beta = \bar{F}_0, \quad (7)$$

$$\omega^2 A \cos \alpha + (g - 2\bar{R}\omega^2) B \cos \beta + 4n_\eta \bar{R} \omega B \sin \beta = 0, \quad (8)$$

$$[\omega_0^2 - \omega^2(1 + \nu)] A \sin \alpha - 2n_x \omega A \cos \alpha + 2\nu \bar{R} \omega^2 B \sin \beta = 0, \quad (9)$$

$$\omega^2 A \sin \alpha + (g - 2\bar{R}\omega^2) B \sin \beta - 4n_\eta \bar{R} \omega B \cos \beta = 0, \quad (10)$$

Далі було проведено виключення зайвих змінних  $B, \alpha, \beta$ . Після певних перетворень (проміжні викладки опущено) отримано рівняння АЧХ  $A(\omega)$  системи “несуче тіло – кульовий гасник”:

$$A(\omega) = \frac{\bar{F}_0 \sqrt{F_6(\omega)}}{\sqrt{F_6(\omega) F_7(\omega) + F_9(\omega) + F_{10}(\omega)}}, \quad (11)$$

де  $F_1(\omega) = \omega_0^2 - \omega^2(1 + \nu)$ ;  $F_2(\omega) = 2n_x \omega$ ;  $F_3(\omega) = g - 2\bar{R}\omega^2$ ;  $F_4(\omega) = 4n_\eta \bar{R} \omega$ ;

$F_5(\omega) = 2\nu \bar{R} \omega^4$ ;  $F_6(\omega) = (F_3(\omega))^2 + (F_4(\omega))^2$ ;  $F_7(\omega) = (F_1(\omega))^2 + (F_2(\omega))^2$ ;

$F_8(\omega) = F_2(\omega) F_4(\omega) - F_1(\omega) F_3(\omega)$ ;  $F_9(\omega) = 2F_5(\omega) F_8(\omega)$ ;  $F_{10}(\omega) = (F_5(\omega))^2$ .

### Аналіз результатів числового експерименту

Для числових експериментів була обрана віброзахисна система з такими параметрами:  $\nu = 0,05$ ,  $n_x = 0,03$ ,  $\bar{F}_0 = 0,03g$ ,  $\omega_0 = 1,2$  рад/с. Оптимальні параметри налаштування кульового гасника були встановлені для задачі в лінійній постановці за допомогою спеціально створених аналітично-числового методу та програмного забезпечення.

На рис.2 наведено графік АЧХ  $A(\omega)$  системи з оптимально налаштованими параметрами гасника. Відомо [2,3], що характерною особливістю оптимально налаштованої віброзахисної системи є те, що її АЧХ має вигляд, симетричний відносно частоти  $\omega = \frac{\omega_1 + \omega_2}{2}$  з двома

рівними за амплітудою максимумами на відповідних частотах  $\omega_1$  та  $\omega_2$ . З фізичної точки зору симетричний вигляд АЧХ вказує на рівний розподіл вхідної енергії зовнішнього збурення між робочим тілом гасника і несучим тілом. Частоти  $\omega_1$  і  $\omega_2$  було знайдено числовими методами [4,5], після

розв’язання нелінійного алгебраїчного рівняння  $\frac{dA(\omega)}{d\omega} = 0$ . На основі створеного аналітично-числового методу та програмного комплексу було знайдено оптимальні параметри налаштування кульового гасника.

Для віброзахисної системи з оптимальними параметрами гасника ( $\bar{R} = 3,822$  М,  $n_\eta = 0,16$ ) величина амплітуд вимушених коливань несучого тіла



на частотах  $\omega_1$  і  $\omega_2$  дорівнює  $A(\omega_1) = A(\omega_2) = 1,032$  м. Для порівняння на рис.3 наведено графік  $A(\omega)$  АЧХ системи, яка функціонує без гасника.

Можна побачити, що максимальне значення амплітуди ( $A(\omega) = 4,09$  м) несучого тіла при такому ж збуренні, як і в попередньому випадку, приблизно в чотири рази більше, ніж з гасником.

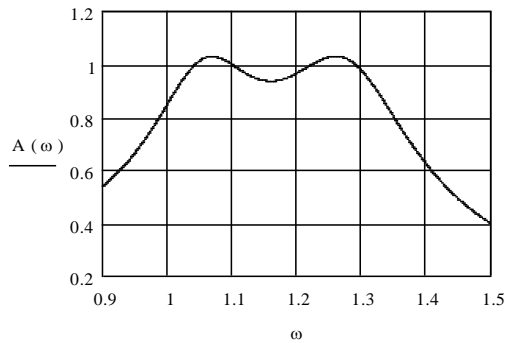


Рис. 2

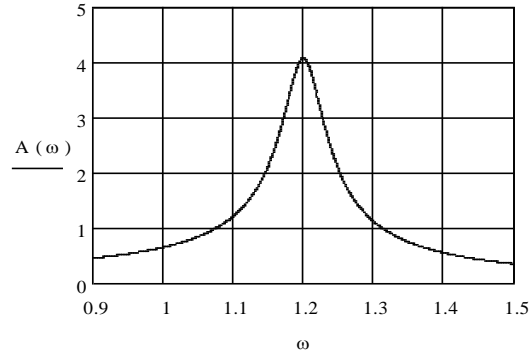


Рис. 3

## Висновки

В роботі зроблено перший крок зі створення оптимального цифрового двійника, що відображає процес придушення небезпечних вимушених коливань висотних гнучких споруд на основі використання кульових гасників. При умові малих вимушених коливань отримано рівняння АЧХ лінійної віброзахисної системи. На основі числового аналізу встановлено, що форма кривої АЧХ суттєво залежить від спеціального налаштування регулюючих параметрів кульового гасника. При його оптимальних параметрах крива АЧХ віброзахисної системи набуває симетричного вигляду з двома рівними за амплітудою максимумами на відповідних частотах  $\omega_1$  та  $\omega_2$ , що вказує на рівний розподіл вхідної енергії зовнішнього збурення між робочим тілом гасника і несучим тілом. Порівняльний числовий аналіз показав високу ефективність функціонування запропонованої віброзахисної системи з оптимально налаштованими параметрами кульового гасника.

## Література

1. Легеза В.П. Теория виброзащиты систем с применением изохронных катковых гасителей: модели, методы, динамический анализ, технические решения. Lambert Academic Publishing, Saarbrucken, Deutschland. 2013. – 108 с.
2. Den Hartog, J.P. Mechanical Vibrations. Courier Corporation, 2013. – 464 p.

3. W.Weaver Jr., S.P.Timoshenko, D.H.Young. Vibration Problems in Engineering, 5th Edition, Wiley, 1991. - 624 p.
4. Пакеты математической обработки данных: [Электронный ресурс]. — Режим доступа: <https://studfile.net/preview/6218974/>.
5. MathCad [Электронный ресурс]. — Режим доступа: <https://mathcad.com>.

**Oleshchenko L.M., assoc. prof., PhD; Burchak Pavlo, student**

**National Technical University of Ukraine  
“Igor Sikorsky Kyiv Polytechnic Institute”**

## **ANALYSIS AND OPTIMIZATION OF METHODS FOR STORING AND PROCESSING DATA OF WEB APPLICATIONS**

### **Анотація**

**К.т.н., доцент Олещенко Л. М., магістрант Бурчак Павло**

*Аналіз та оптимізація методів зберігання та обробки даних вебзастосунків*

*У роботі проведено дослідження з використання програмних засобів для зберігання та управління станом вебзастосунків. Запропоновано програмний оптимізований метод для зберігання стану різного роду застосунків при різних вхідних даних. Описані переваги та недоліки використовуваних аналогів такого роду бібліотек та методів.*

### **Introduction**

With the huge increase of demand on software development, more and more new technologies are coming to this sphere. Especially web application development is in its better states. But, for instance when there comes to long-term application support, here comes the question. How to efficiently manage the state and data storage of the application? There are many different methods and libraries that provide modern tools to accomplish such a task, but when comes to large scaling or developers convenience, some issues could happen. Our task is to figure out which methods and libraries are better choice under certain circumstances. The result of this research would be newly created optimized method that will both be great scalable solution and be good to work with.

### **Terminology**

*Hypertext Markup Language (HTML)* is the basis of any website development process, without which there is no web page.

*Cascading Style Sheets (CSS)* is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.

*JavaScript* is an imperative event-based programming language (as opposed to the declarative HTML model), which is used to convert a static HTML page into a dynamic interface.

## **Problem statement**

State management refers to the state management of one or more web application user interface controls, such as text boxes, buttons, switches, etc. in the graphical user interface (UI). In this UI programming technique, the state of one interface control depends on the state of the other interface controls. For example, a state-controlled user interface control, such as a button, will be on when input fields have valid input values, and a button will be off when input fields are empty or have invalid values. As applications grow, this can eventually become one of the most difficult challenges in developing a user interface.

This is especially true when the state of any particular message or form on a page depends on factors that are not included in the current page or are available on multiple pages. For example, consider a user who is logged in and sees a greeting message the first time they visit any page, but not the next time they visit. Does each page control the status of the user it belongs to? This would create too much copying and duplicating code. Instead, we can use the status management template to process messages (this may also include processing error messages and informational messages, along with the welcome message described), and then call it to receive the message when it becomes available.

Examples of state management libraries. Vuex is a state management library for the Vue.js JavaScript framework. The Angular structure includes its own Observables library, called RxJS, and Redux is a common state management library that can be used with any of the above frameworks or other view libraries, but is very often used with the React library. As noted in the Redux documentation, many of these public management libraries are lightweight and can be replaced.

## **Sphere analysis**

A web application is a distributed application in which the client is a browser and the server is a web server. The browser can be an implementation of so-called thin clients – the logic of the application focuses on the server, and the function of the browser is mainly to display information downloaded by the network from the server, and transfer back user data. One of the advantages of this approach is the fact that clients do not depend on the specific operating system of the user, so web applications are cross-platform services.

Due to this versatility and relative simplicity of web application development, they became widely popular in the late 1990s and early 2000s.

A significant advantage of building web applications to support standard browser functions is that the functions must be performed independently of the

client operating system. Instead of writing different versions for Microsoft Windows, Mac OS X, GNU / Linux and other operating systems, the application is created once for a randomly selected platform and deployed on it. However, different implementations of HTML, CSS, DOM, and other specifications in browsers can cause problems in web application development and subsequent support. In addition, the user's ability to customize many browser settings (such as font size, colors, disabling scripting support) can prevent the application from working properly [1-3].

Hypertext means that the text has built-in links, which are called hyperlinks. When a user clicks on a word or phrase that has a hyperlink, he opens another web page. Markup language means that text can be converted to images, tables, links, and other images. It is the HTML code that provides the overall structure of what the site will look like. HTML was developed by Tim Berners-Lee. The latest version of HTML is called HTML5 and was published on October 28, 2014 on the recommendation of W3. This version contains new and efficient ways to handle items such as video and audio files.

It does this by supporting style sheets that sit on top of other style rules and run based on other inputs, such as device screen size and resolution.

JavaScript code can use an HTML object model (DOM) to manipulate a web page in response to events such as user input.

Using AJAX technique, JavaScript code can also actively retrieve content from the Internet (regardless of the original HTML page search), as well as respond to server-side events, adding a truly dynamic nature to the web page. Web application framework or web framework is a software framework designed to create web applications, services or resources. It simplifies development, in part through automation, and eliminates the need to write routine code. Most frameworks simplify access to databases and also reduce code duplication. Most web application frameworks implement the Model-View-Controller (MVC) design template.

The architecture of almost all frameworks is based on the decomposition of several individual words (applications, modules, etc.), which means that the developer can expand the functionality, depending on their needs, and use the produced version with the framework code or use third-party applications. This flexibility is another key advantage of frameworks .

There are many open source communities and commercial organizations that create applications or extensions for popular frameworks, such as the Django REST Framework, ng-bootstrap, and more.

### **The proposed method**

The proposed method is a combination of the best software solutions selected for library analysis. Our research is conducted for each of the method

libraries. Features have been written using JavaScript and have been minimized for faster downloading. If the application makes rendering of long rows, we use a method of window's accessing. This method makes rendering only a small amount of a subset of rows during one timeframe and allows reducing time which is needed for repeating rendering of components and numbers of creating DOM knots [4]. React components of application have the structure in a shape of tree and the root element, which includes children components of hierarchical structure. If the main component is updated, children components are updated too. It loads the system, as not all components have to be updated.

The current component does not have to be updated. For optimization of components it is possible to apply the automation of the comparing props by the method of prohibition or permissions of updating the component with using of the components, which are imported from the React library.

Optimizing the developed system with front-end technologies does not affect the functionality and integrity of the system. System must fulfill its tasks, use design patterns and have the same routes as before optimization, optimization provides reduce server load.

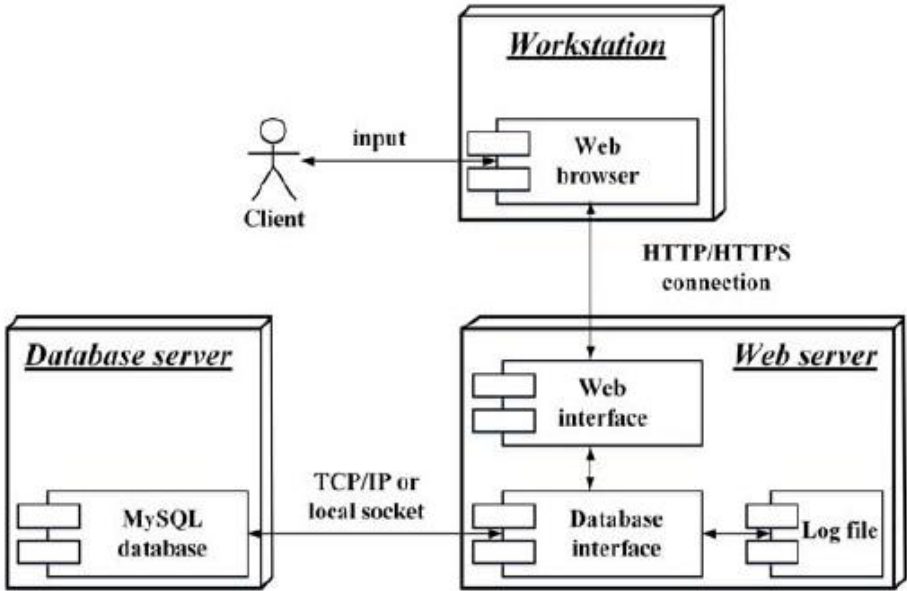


Fig. 1. Client-server interaction [5].

A web application with a user interface is developed to be used as a test template for libraries. That is, libraries will serve as data providers and operators. The research includes measurements of the difference in program execution time, the speed of rendering the interface when changing data.

The research also includes an analysis of the convenience of writing code for the developer. An attention should be paid to such parameters as code complexity, code smell, difficulty in understanding the code by the developer. The research stages of the proposed method are the measurement of page load

time, external resources in a non-optimized system; measurement of time of script execution; configuring load balancing between server clusters using JavaScript code; setting the sequence of loading of resources and basic JavaScript of the system; setting up parallel load of resources; providing interactive download; minimizing js, css, html files; setting of data caching; measurement of page load time, comparing optimized and non-optimized systems with slower internet connections.

### **Conclusions**

After the all proposed technologies research, our conclusions are made about each of the libraries. Based on this, the selected methods will be optimized and improved, which will result in the development of their own methods for managing the status of the web application. After development, a comparison of the developed combined library will be performed to solve the applied problems with existing analogies, on the basis of which a conclusion will be made about the feasibility of such an approach.

### **References**

1. Alternatives to React Redux in 2020 [<https://blog.bitsrc.io/redux-react-alternatives-c1733793a339>].
2. Tajima, M., Goto, K., Toyama, M.: Non-procedural generation of web pages with nested infinite-scrolls in superSQL. In: Proceedings of the 19th International Conference on Information Integration and Web-based Applications & Services, pp. 289-295 (2017). doi:10.1145/3151759.3151806.
3. Ahmed, T.M., Bezemer, C.P., Chen, T.H., Hassan, A.E., Shang, W. Studying the effectiveness of application performance management (APM) tools for detecting performance regressions for web applications: an experience report. In: IEEE/ACM 13th International Conference on Mining Software Repositories (MSR), pp. 1-12. IEEE (2016)
4. Atterer, R., Schmidt, A. Tracking the interaction of users with AJAX applications for usability testing. In: Proceedings of the SIGCHI conference on Human factors in computing systems, 1347-1350 (2007)
5. Implementation of Web System Optimization Method [<http://ceur-ws.org/Vol-2608/paper16.pdf>]

UDC 004.415.2

Oleshchenko L.M., assoc. prof., PhD; Sadrytskyi Serhii, student

National Technical University of Ukraine  
“Igor Sikorsky Kyiv Polytechnic Institute”

## MATHEMATICAL METHODS FOR CLUSTERING SOFTWARE USERS BY GIVEN ATTRIBUTES

### Анотація

К.т.н., доцент Олещенко Л. М., магістрант Садрицький Сергій

*Математичні методи кластеризації користувачів програмного забезпечення за даними атрибутами*

*У роботі проведений огляд і порівняльний аналіз математичних методів кластеризації даних. Дослідження проводиться з метою визначення оптимального математичного методу кластеризації персональних даних користувача програмного забезпечення за заданими атрибутами для використання у відповідному програмному методі. Наведено властивості математичних методів, взятих до розгляду та загальна структура модулів програмного забезпечення.*

### Introduction

The data around us is being generated in such huge volumes it was never possible before. Innovations in sensing and storage technologies, as well as rapid expansion in applications such as Web search, digital imaging, and video monitoring, have resulted in the creation of a large number of high-volume, multidimensional datasets. Aside from the increase in data volume, the variety of data (text, image, and video) has expanded as well. Because the majority of data is kept digitally, there is a significant opportunity for the advancement of automatic data analysis, classification, and retrieval technologies. The aim of cluster analysis is to organize data of complex forms into meaningful groupings. It has recently emerged as one of the most fundamental mechanisms of comprehension and learning. In this paper, we compare the most popular clustering methods for differentiating and analysis of personal text data, with the goal of incorporating the best features of them into a newly developed software method with a wide variety of uses.

### Terminology

*Cluster analysis* – the process of arranging a collection of objects so that those in the same group are more similar than those in other groups. It is a common technique for statistical data analysis used in many domains, including



pattern recognition, image analysis, information retrieval and others, and is a main goal of exploratory data analysis.

*Data Mining* – the process of searching through big data sets in order to find patterns and relationships that may be used to address business problems. Enterprises can use data mining techniques and technologies to forecast future trends and make better business decisions.

**Problem statement**

Clustering refers to a multitude of mathematical methods, that play an important role in data mining analysis. It is also called unsupervised learning. Clustering concerns on finding patterns in data fragments and organizing data into classes such that there is high intra-class similarity and low inter-class similarity among them, and finding the class labels and the number of classes directly from the unlabeled data (fig.1).

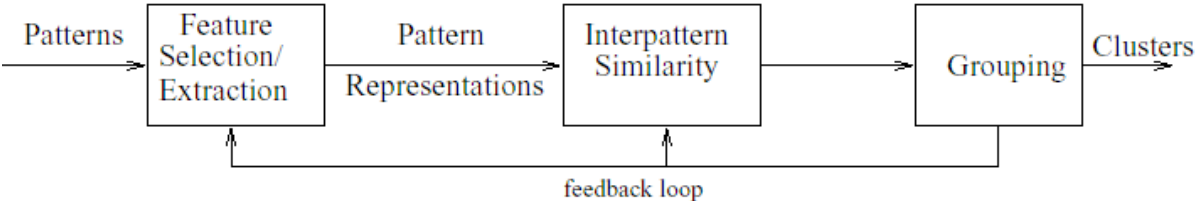


Fig. 1. Clustering process scheme [1].

Several approaches to clustering exist. Hierarchical, Partitioning, and Density-based clustering techniques are the three types of clustering algorithms. The Partitioning clustering algorithm divides data objects into a number of partitions, each of which represents a cluster. For data objects, hierarchical clustering techniques generate a hierarchy or tree of groups. Density-based algorithms organize data objects based on their location in a specific neighborhood and discover clusters in high-density areas.

Most often datasets being extracted are often large, usually containing noise and deviations called outliers. Outlier is a data object pattern that differs from the other patterns, frequently caused by human or system error. As a result, different methods must be applied to effectively identify, detect, and handle noise and outliers [1-3].

*Hierarchical clustering algorithms* create a hierarchy or tree of data objects. The method can start with a single cluster and then divide it into smaller clusters (Divisive or Top down) or start with an object and then group related clusters together (Bottom up).

Starting with each of the n objects in a single object cluster, an agglomerative hierarchical clustering process groups the two closest (or most similar) objects into a cluster, reducing the number of clusters to n-1. The

process is repeated until all of the objects have been grouped into the n-object cluster.

The rigidity of the agglomerative clustering algorithm derives from the fact that all splits or merges are final once completed and cannot be reversed which can be problematic for noisy data.

BIRCH algorithm (Balanced Iterative Reducing and Clustering Using Hierarchies) tries to solve the problem but the disadvantages are difficult scaling because of time complexity and poor noise detection which is crucial for large datasets processing. Both agglomerative and divisive approaches need a hierarchy structure within input data.

*Partitioning clustering algorithms.* When a clustering technique results in the object set being divided into  $K$  clusters, it is referred to as partitioning. The number of clusters  $K$  is fixed, and one can begin with any start partition into  $K$  clusters. The goal is to gradually improve this partition. This method has the advantage of allowing objects to freely move from one cluster to the next. Even if the start partition was bad, the final partition will be good as a result of this.

*K-means.* Mostly used partitioning algorithm is  $k$ -means. The  $k$ -means algorithm is simple to implement, and its design is to avoid the sum of distances by breaking up data points into partitions called  $k$ , where one cluster represents one partition. The algorithm arbitrarily chooses  $k$  data items from a set ( $D = \{d_1, d_2, d_3, \dots, d_n\}$ ), after that assigning initial values for  $m_1, m_2, \dots, m_k$ . Then it repeats assigning each item  $d_i$  to the clusters which has the closest mean and calculating new mean for each cluster until convergence criteria is met. The disadvantages are that only numerical attributes work properly with the  $k$ -means algorithm, however a single outlier has a serious effect.

*Density based clustering algorithms* search the data space for areas of varied density of data points in the data space. It isolates various different density regions and assigns the data points within these regions in the same cluster. Popular examples of density models are DBSCAN and OPTICS.

*DBSCAN.* Data points are divided into three categories by the DBSCAN: core, border, and noise. Core points indicate locations in the inside of a cluster, while Border points indicate sites in the proximity of a core point but are not core points, and Noise points indicate any point that is neither a border nor a core point in a density-based clustering algorithm hierarchy.

Its ability to detect and deal with noise and outliers is a key property, and hence the foundation for its great sensitivity in recognizing and dealing with them.

There are other clustering methods which are not present in the observation such as Graph-based and Model-based clustering, but in this research, we are concerned with three forementioned approaches.

## The proposed method

The proposed method is a combination of the top software solutions for visualization and processing part. An optimal mathematical clustering method is chosen as a core part of the implementation. Each of these methods and our software libraries are subjected to its own investigation.

The described research combined with the process of development will result in a software method consisting of a visual and back-end parts. Both of them are based on the latest software design solutions. Also we check the effectiveness of the output method and the convenience for the end user.

After receiving a request from the user, the server determines its type. For GET requests, the header from which the goal and parameters are derived is parsed. If their combination is provided by the API of the server, it performs the processing provided by this request and generates a response, type JSON / application and the result of processing in JSON format. For POST-queries, the header is parsed, from which the target, parameters and body are obtained, the format of which is determined by the API of the server (fig.2).

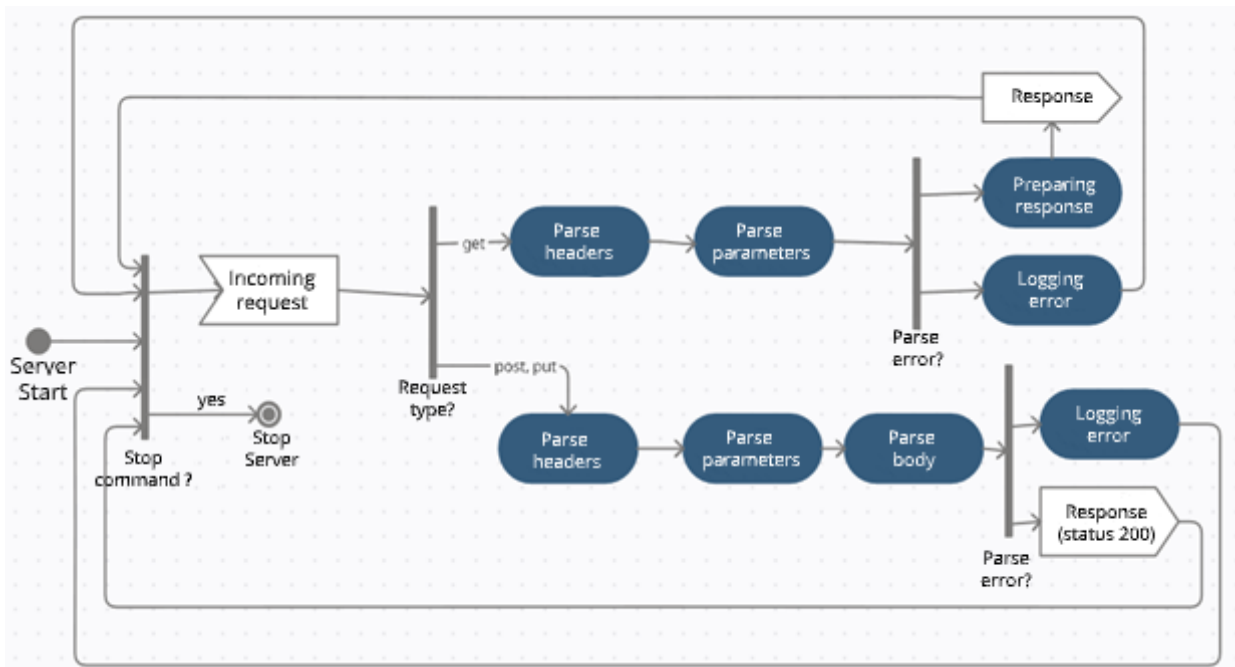


Fig. 2. Diagram of server activity.

The software for the proposed method contains the following modules: questionnaire generation module for creating user test questionnaires in JSON format; module of initial analysis of questionnaires which performs an initial analysis of the questionnaires; examination simulation module which randomly generates the results of the surveys proposed by the initial questionnaire analysis module; module of detailed analysis based on the test results; clustering module which clusters users according to the needs and displays the corresponding

graphs. The post-processing is carried out using the method of determining the optimal mathematical method for clustering personal data of the software security for the given attributes for the application of the specific software method.

### **Conclusions**

The main goal of this research is to create an optimal software solution for grouping software users' personal data by supplied attributes using a mix of the best aspects of existing mathematical methods. A comparison review of these methodologies was conducted, and some of their key characteristics that were essential in the case study were found. In the research, a decision regarding the software solutions to be employed in the implementation of the proposed method is made and carried out the testing of effectiveness of the concept. Mathematical methods have been introduced, taken to the point of looking at the structure of the software modules.

### **References**

1. Clustering algorithms: A comparative approach [<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6333366/>].
2. Ashok, A. R., Prabhakar, C. R. and Dyaneshwar, P. A. (2013). Comparative Study on Hierarchical and Partitioning Data Mining Methods. International Journal of Science and Research (IJSR), India, 2(3), pp. 211-215.
3. Bala, R., Sikka, S. and Singh, J., (2014). A Comparative Analysis of Clustering Algorithms. International Journal of Computer Applications, India, 100(15), pp. 35-39.
4. Network protocols [<https://proglib.io/p/computer-networking/>]
5. UML [[https://www.tutorialspoint.com/uml/uml\\_activity\\_diagram.htm](https://www.tutorialspoint.com/uml/uml_activity_diagram.htm)]

**Oleshchenko L.M., assoc. prof., PhD; ZhangYingzhi, student**

**National Technical University of Ukraine  
“Igor Sikorsky Kyiv Polytechnic Institute”**

## **SOFTWARE ANALYSIS METHOD OF STATISTICAL DATA ABOUT THE TERRITORIAL LIVING STANDARD OF PEOPLE**

### **Анотація**

**К.т.н., доцент Олещенко Л. М., магістрантка Чжан Інчжи  
Програмний метод аналізу статистичних даних про територіальний рівень  
життя населення**

*У роботі проведено дослідження з використання програмних засобів для обчислення та аналізу рівня сталого розвитку і ступеня гармонізації суспільства, оцінки впливу соціальних, економічних, політичних та екологічних факторів на рівень життя населення. Запропоновано програмну систему для оцінки впливу факторів на рівень сталого розвитку та кластеризації регіонів за різними показниками на основі статистичних даних по регіонах України за 2017-2018 роки. Описано методіку визначення коефіцієнта сталого розвитку та ступеня гармонізації та математичні методи обробки даних. Для розроблення програмного забезпечення використано технології Big Data та інструменти Python.*

### **Introduction**

With the rapid development of information and communication technology, the amount of global data is exploding. How to effectively analyze the massive complex data, excavate and realize the potential value and make rational use of it is one of the important topics at present. This paper designed and implemented a big data analysis and prediction system based on the analysis of the territorial living standard of people.

Our task is to figure out what factors and in what percentage depends on the standard of living of people in the country and how they can be analyzed and predicted based on the existing dynamics of factors.

### **Terminology**

*Big Data* – a combination of structured, semistructured and unstructured data collected by organizations that can be mined for information and used in machine learning projects, predictive modeling and other advanced analytics applications.

*Sustainable development* – an organizing principle for meeting human development goals while simultaneously sustaining the ability of natural systems to provide the natural resources and ecosystem services on which the economy and society depend on.

### **Problem statement**

The methodology of assessment and analysis of sustainable development contains a sustainable development model that is an interdisciplinary generalization of models known in natural, economic, and social science, and the methodology of applying formal statistical methods and expert assessment methods to analyze sustainable development processes.

According to this methodology, we will characterize the sustainable development process by two main components: security ( $C_{sl}$ ) and quality ( $C_{ql}$ ) of human life, and we define the generalized measure of sustainable development via a quaternion:

$$\{Q\} = j\omega_{sl}C_{sl} + \omega_{ql}\|\overline{c_{ql}}\| \quad (1)$$

Quaternion  $\{Q\}$  contains an imaginary weighted scalar part  $j\omega_{sl}C_{sl}$  that describes the security of human life and a weighted real vector part that describes the quality of human life within three dimensions: economic ( $I_e$ ), environmental ( $I_{ec}$ ), and social and institutional ( $I_s$ ). At that,  $j$  acquires the value of real one for a normal regular state of society development ( $C_{sl} > 0$ ), and the value of an imaginary one when the society enters into the state of conflict ( $C_{sl} = 0$ ). According to the results of the analysis of the state of the world countries, the Heidelberg Institute for International Conflict Research determined the following conflict levels:

- 0 (Without crisis) — no conflicts in the country;
- 1 (Dispute) — political conflict with no human casualties;
- 2 (Non-violent crisis) — political conflict where one of the parties made threats of violence;
- 3 (Violent crisis) — low-intensity conflict with low number of human casualties, with only light weapons used;
- 4 (Limited war) — intense conflict with medium number of human casualties, with heavy weapons used or high number of personnel involved;
- 5 (War) — intense conflict with high number of human casualties, with heavy weapons used and high number of personnel involved.

Therefore,  $j$  may be determined as follows:

$$j = \begin{cases} 1, & \text{for conflict level 1 - 4;} \\ \sqrt{-1}, & \text{for conflict level 5.} \end{cases} \quad (2)$$

Weighted coefficients  $\omega_{sl}$  and  $\omega_{ql}$  in formula (1) are used in order to level the scales of components of security and quality of life (in case of assessing sustainable development of the world countries ( $\omega_{sl}=1/\sqrt[3]{11}$ ,  $\omega_{ql}=1/\sqrt{3}$ )).

The sustainable development index is the quantitative measure of sustainable development, which takes into account security and quality of human life. For each country, the Euclidian norm of the human life quality radius-vector ( $\overline{Cq1}$ ) shall be as follows:

$$\|\overline{Cq1}\| = \sqrt{I_{ec}^2 + I_e^2 + I_s^2}. \quad (3)$$

Deviation angle  $\alpha$  of radius-vector  $\overline{Cq1}$  from an ideal vector (1,1,1) is determined through values of  $I_{ec}$ ,  $I_e$ ,  $I_s$  as follows:

$$\alpha = \arccos \frac{I_{ec} + I_e + I_s}{\sqrt{3} \cdot \sqrt{I_{ec}^2 + I_e^2 + I_s^2}}, \quad 0 \leq \alpha \leq \arccos \frac{1}{\sqrt{3}}. \quad (4)$$

Thus, the norm of radius-vector  $\overline{Cq1}$  characterizes the quality of human life, and the spatial position of vector  $\overline{Cq1}$  in the system of coordinates ( $I_{ec}$ ,  $I_e$ ,  $I_s$ ) characterizes the measure of “harmonicity” of sustainable development. The equidistance of vector  $\overline{Cq1}$  from each of the indicated coordinates shall correspond to the high sustainable development harmonicity, and this vector’s approach to one of the coordinates shall indicate the priority development in one dimension and neglection of two others.

Let us call value  $G=1-\alpha$  the degree of sustainable development harmonization. It increases if  $G$  approximates 1, and decreases if  $G$  approximates 0. Thus, the life quality component is an integral estimation that simultaneously takes into account all three dimensions of sustainable development, and therefore reflects the interconnection between three inseparable spheres of the society development: economic, environmental, and social. The values of the sustainable development dimensions, which are used to determine the quality of life component, should base upon data on the wide range of phenomena of various natures. Such values should also be integral.

For the purpose of quantitative evaluation of sustainable development dimensions, we will apply the principles of constructing a hierarchical system of indicators and indices determined as L1-norms:

$$I_i = \sum_{j=1}^n \omega_j x_{i,j}, \quad i=1, m, \quad \sum_{j=1}^n \omega_j = 1 \quad (5)$$

in space of indicators that characterize economic, environmental, and social and institutional development of each  $i$  country. Weighted coefficients  $\omega_j$  in formula (5) are determined by experts.

The main objective of the research is to create software analysis method of statistical data about the territorial living standard of people. The specific objective is to mathematical and statistical methods of data analysis, simulation modeling, clustering, Python tools for data analysis.

### **The proposed method**

The data source layer is the statistical website from domestic big data. The data access layer is implemented in collaboration with Nginx using Kafka. Different types of data aggregated from different ports will be sent to different topics for processing. The processed information will be sent to HDFS for

further processing. Flume-ng is used in the data extraction layer, where Source is Kafka and Sink is HDFS. Using Flume as an efficient data extraction tool, the data of different topics can be constructed in different HDFS directories according to the configuration of interceptor, and continuous data transmission can be realized through automatic configuration.

The data storage layer consists of HDFS, Hive, and HBase. The original user log data extracted by Flume-ng is stored in HDFS and corresponds to Hive. It is stored in HBase as long string data for long-term storage. The data processing layer mainly uses Spark technology. As a near-line data platform for hourly data analysis, SparkSQL is mainly used for data processing of ETL, and SparkMilb combined with Python is used for data mining algorithm. Scheduled tasks in ETL processing are scheduled by the external platform Uni Spark. The business data layer is the storage layer of ETL data, which is stored by Mysql database and Redis database. This layer is used as a business invocation for the Web platform, as well as for the data presentation module and the user tag mining module (Fig. 1).

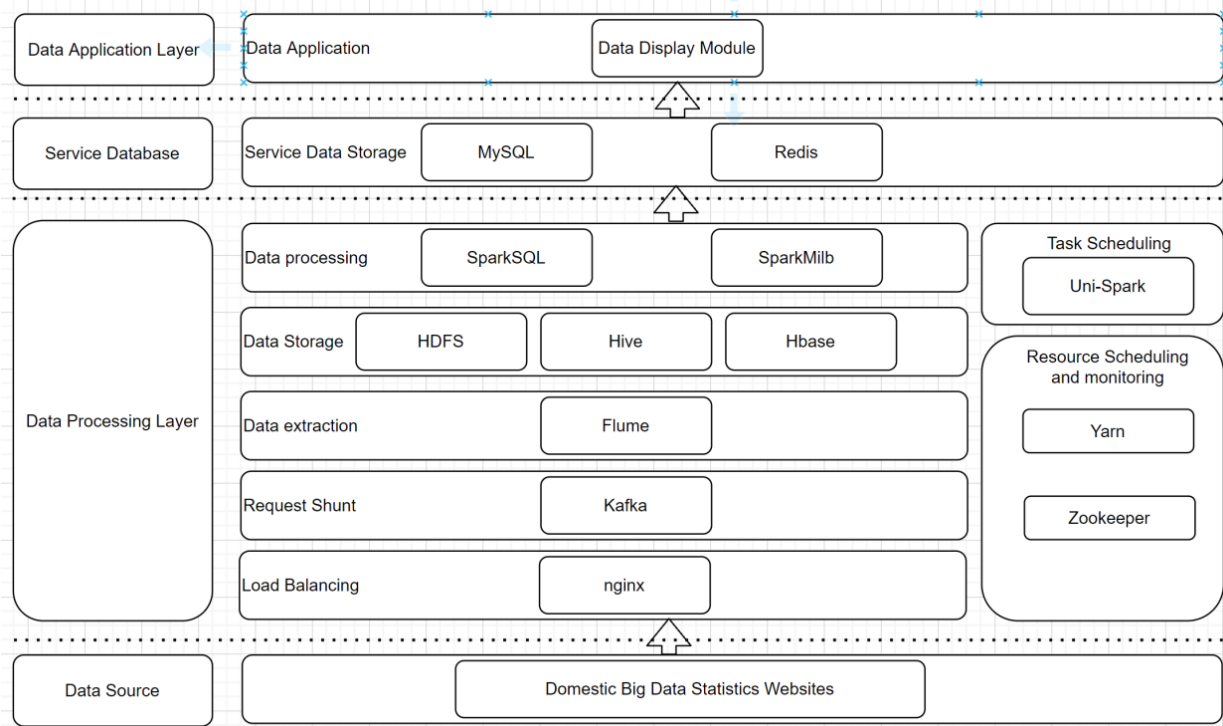


Fig. 1. Data processing Architecture

In order to receive the data by the regions (sustainable development index and harmony level) via the web application we navigate to the home page and upload the .osd data file containing the information on 27 territorial units of Ukraine (24 regions, Autonomous Republic of Crimea, cities Kyiv and Sevastopol). In order to do this we click the “Choose File” button. After that we



choose the region to obtain the information on and press the “Upload” button. After this we receive the resulting data for every year for the chosen region, such as the sustained development index and harmony level (Fig. 2). The developed software also allows to perform correlation and regression analysis of data for different regions.

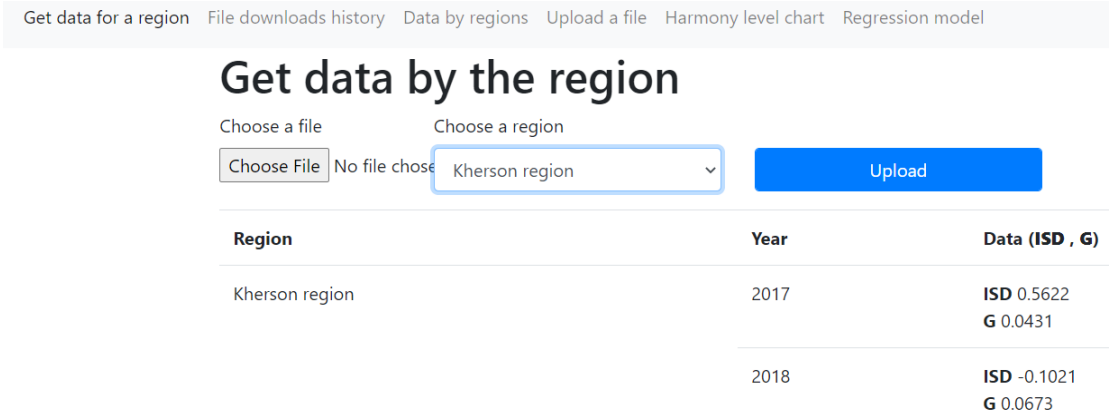


Fig. 2. Receiving the data for a region

About China, we can see three parts of quality of life components: index of social and institutional dimension, index of economic dimension and index of environmental dimension, all the values we can see in Fig. 3.

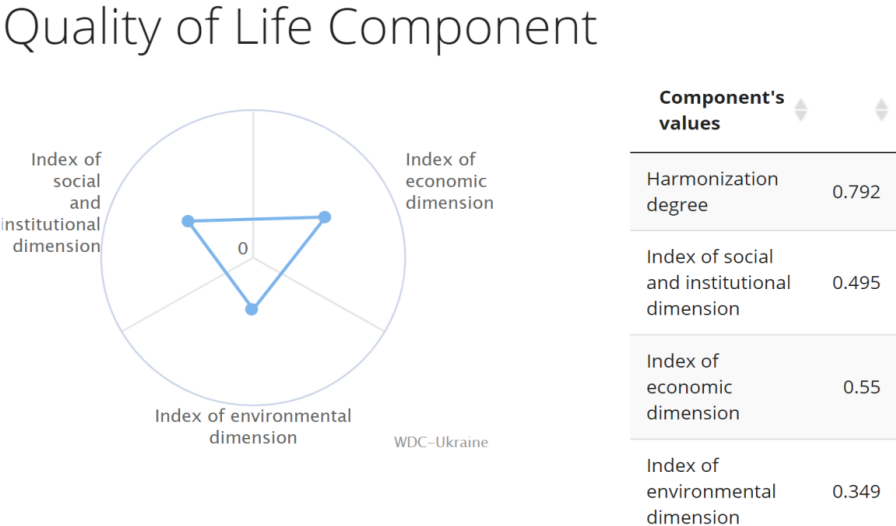


Fig. 3. Quality of Life Component

The results about the security of life component of China we can see in Fig. 4.

## Security of Life Component

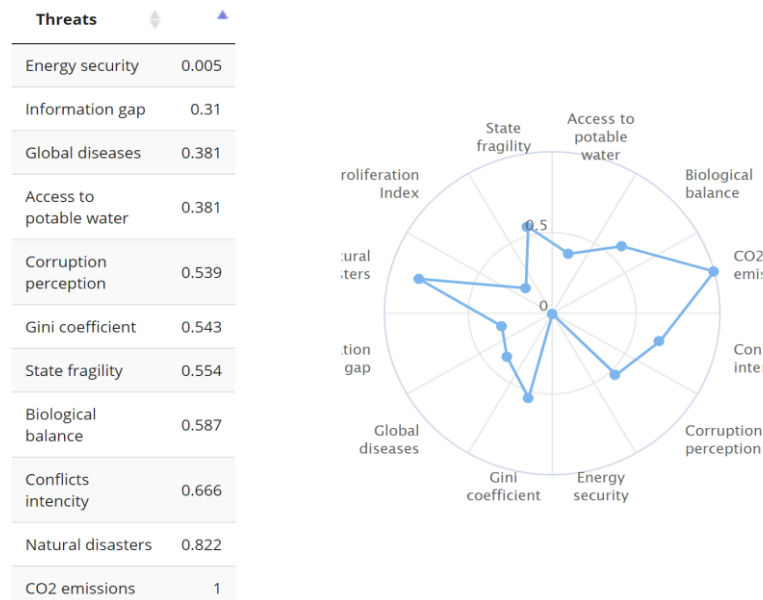


Fig. 4. Security of Life Component

## Conclusions

This research is devoted to the analysis of the level of sustainable development of society. The method of calculating the index of sustainable development of society and the components of quality of life are analyzed. And there is a method that analyze statistics and calculate sustainable development indices. There are many problems in China that need to be solved. Through this method, problems can be clearly identified and solved as soon as possible, so as to provide a better living environment and living standard for the people. Software for analyzing the sustained development of Ukraine's regions are developed, and with the use of it we have analyzed the data and received conclusive results.

## References

1. M. Zgurovsky. Sustainable Development Global Simulation: Opportunities and Threats to the Planet / M. Zgurovsky // Russian Journal of Earth Sciences. — 2007. — Vol. 9, ES2003, doi:10,2205/2007 ES000273.
2. Sustainable Development Analysis: Global and Regional Contexts: in 2 parts / A. Boldak, S. Voytko, I. Dzhyhyrei and others: Scientific Supervisor M. Zgurovsky. — K.: NTUU «KPI», 2009. — Part 1. Global Analysis of Quality and Security of Life. — 2009. — 280 p.

**Ph.D., Associate Professor Mykola Onai,  
Master's student Ahmad Husein**

**National Technical University of Ukraine  
"Igor Sikorsky Kyiv Polytechnic Institute"**

## **METHODS OF PERFORMING OPERATIONS IN ELLIPTIC CURVE OVER $GF(P)$**

### **Abstract**

**Mykola Onai, Ph.D., Associate Professor; Ahmad Husein, Master's student**  
*Methods of performing operations in elliptic curve over  $GF(p)$*

*This paper describes that the fundamental operation of the scalar multiplication on elliptic curve is the operation of calculating of the multiplicative inverse element in  $GF(p)$ . Methods for calculating multiplicative inverse element has been researched. Extended Lehmer algorithm was built. Adaptive extended Lehmer algorithm is proposed. It ensures a significant increase of a speed. Experimental results are given.*

### **Introduction**

Modern cryptographic transformations with an open key are based on transformation in the group of points of the elliptic curve (EC) [1]. One of the main operations in the elliptic curve is the multiplication of the EC point by number (scalar multiplication in the group of points EC) [1, 2].

Any algorithm of multiplying the EC point by the number involves performing a certain number of operations of adding EC points and doubling the EC point. The implementation of these operations in the affine coordinates is based on the Diophantine method.

We will consider the EC in the form of Weierstrass

$$y^2 = x^2 + ax + b \quad (1)$$

over  $GF(p)$ .

Let  $P_1(x_1, y_1)$  the  $P_2(x_2, y_2)$  EC points set by the equation (1). Then, the following  $P_3(x_3, y_3) = P_1 + P_2$  formulas define the point's affine coordinates:

$$x_3 = \lambda^2 - x_1 - x_2 \pmod{p} \quad (2)$$

$$y_3 = -y_1 - \lambda(x_3 - x_1) \pmod{p} \quad (3)$$

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} \pmod{p}, \text{ if } P_1 \neq \pm P_2 \quad (4)$$

$$\lambda = \frac{3x_1^2 + a}{2y_1} \pmod{p}, \text{ if } P_1 = P_2 \quad (5)$$

### Formulation of the task

In formulas(2) – (5), the most calculated and calculated operation is the calculation of a modular multiplicative inverse. Taking into account this fact, the task of research and modification of existing methods of calculation of modular multiplicative inverse in the  $GF(p)$  is actual. Due to this, it is possible to significantly improve the time indicators of the basic algorithms of multiplying the EC point by the number in the affine coordinates.

### Methods for calculating a modular multiplicative inverse in $GF(p)$

There are many algorithms to calculate a modular multiplicative inverse. From the point of view of computing costs, the most attractive methods based on generalized methods of calculating of the greatest common divisor ( $GCD$ ). It is known that the modular multiplicative inverse element exists, when  $GCD(m, b) = 1$  where  $m$  is the module [3]. The generalized algorithms of calculating  $GCD$  sometimes call advanced algorithms.

In contrast to the  $GCD$  calculate algorithms, which output only a number  $d = GCD(m, b)$ , the result of the advanced algorithms is a number  $d$  and two additional numbers:  $x$  and  $y$ , such that the equality is fulfilled:

$$x \cdot m + y \cdot b = d. \quad (6)$$

If you need to find  $b^{-1} \pmod{m}$ , then  $d = 1$  and  $x \cdot m = 0 \pmod{m}$ , therefore equality (6) takes the form of:

$$y \cdot b = 1 \pmod{m}, \quad (7)$$

from here we figure out that  $y = b^{-1} \pmod{m}$ .

We will consider the basic classes of advanced algorithms of calculating  $GCD$ .

### Classic extended Euclid algorithms

The first class of advanced algorithms is classic extended Euclid algorithms, the main ideas of which were launched in India in the V century [2].

In the classical extended Euclid algorithm during operation the algorithm maintains the level of the view:

$$A \cdot m + B \cdot b = u \quad (8)$$

Gordon Bradley was shown [2] that:

$$A \cdot m = u$$

and at the end, calculate the value of the second coefficient by formula

$$B = \frac{u - A \cdot m}{b} .$$

Given this fact, and using Gordon Bradley's idea, we can modify the classic extended Euclid algorithm so that it calculates only  $B$ :

$$B \cdot b = u$$

### ***Binary extended Euclid algorithms***

The second class of advanced algorithms is binary extended Euclid algorithms.

It can be proved that in the case when  $m - \text{odd}$ ,  $b - \text{odd}$ , and the parity condition for is not fulfilled, it will always prove that  $A - \text{even}$  and  $B - \text{odd}$ . That is, it is enough to check for parity only  $B$ . For a case where  $m - \text{odd}$  and  $b - \text{odd}$ , parity of  $A$  and  $B$  coincident. Therefore, it is enough to check for parity only  $B$ .

Thus, for an odd module in binary algorithms it is enough to maintain only the equations  $B \cdot b = u$ . If the module is an even, then for the correct operation of the algorithm it is enough to maintain only the equations  $A \cdot m = u$  and the value  $y$  can then be calculate by formula  $y = \frac{u - A \cdot m}{b}$ .

Therefore, at the beginning of the binary algorithm it is necessary to perform a parity test of the module, and then maintain the parity of the  $A \cdot m = u$  or  $B \cdot b = u$

It is only at the expense of this that it is possible to reduce the number of operations almost twice.

### **Experimental results and modification of the algorithms considered**

In this paper we conducted experimental studies of the time of execution of the modification Gordon Bradley of the extended Euclid algorithm (algorithm #1), proposed by our modification of the classic extended Euclid algorithm (algorithm #2), modification of the extended binary algorithm Euclid proposed by Michael Penk (algorithm #3), we proposed modification of the binary algorithm of the Euclid ( algorithm #4).

Experimental studies were conducted on a PC with the following specifications: CPU Intel Core i5 10th-Gen, 2.9 GHz, RAM 16 GB. All algorithms are implemented in C# programming language.

Table 1 shows that the best results are shown by algorithm #2, which is a modification of the classic extended algorithm of the Euclid. This suggests that first of all it is advisable to carry out further modifications of the first class of extended algorithms of Euclid.

Table 1

	Algorithm operating time, ms			
	Module length (bit)			
	<b>256</b>	<b>512</b>	<b>1024</b>	<b>2048</b>
Algorithm #1	0,041	0,075	0,121	0,353
Algorithm #2	0,035	0,061	0,115	0,325
Algorithm #3	0,379	0,925	2,587	9,534
Algorithm #4	0,345	0,817	2,105	7,595

French scientist Lehmer proposed modification of the classical algorithm of Euclid for calculating *GCD* of two numbers. His idea is that for large numbers the quotient  $\frac{m}{b}$  will not change if in numbers  $m$  and  $b$  to throw a certain least significant digits. Using this idea, we can build an extended Lehmer algorithm. The disadvantage of this algorithm is a fixed number of the least significant digits, which is thrown away while operands can have different length. Therefore, we proposed an adapted extended algorithm of Lehmer, in which each iteration of the algorithm determines the number of least significant digits, which will be rejected, depending on the current length of operands on this iteration. As a result of this modification, the increase in speed is provided in comparison with algorithm №2 on average by 10%.

## Conclusions

As a result of the research, it is shown that the most promising direction of improvement of existing algorithms of search of modular multiplicative inverse element is the development of the idea proposed by Lehmer for calculating *GCD* of two numbers. Thus, in this work proposed modified extended Euclid algorithm, using Gordon Bradley's idea, constructed an extended Lehmer algorithm and offered an adaptive extended Lehmer algorithm, which will provide acceleration of calculating of modular multiplicative inverse in an average of 10%. Using the proposed adaptive extended Lehmer algorithm in the multiplication of the elliptic curve point by number provides a significant increase in speed.

## References

1. Hankerson, D. Guide to Elliptic Curve Curve [Text] / D. Hankerson, A. Menezes, S. Vanstone. – New York : Springer, 2004. – 311 p.

2. Crandall, R. Prime numbers. A Computational perspective. Second Edition [Text] / Richard Crandall, Carl Pomerance. – New York : Springer, 2005. — 604 p.
3. Cohen, Henri A course in computational algebraic number theory [Text] / Henri Cohen, J.H. Ewing, P.R. Halmos. – Berlin ; Heidelberg ; New York : Springer, 1996.

**К.т.н, доцент Онай М.В., магістрант Квітка О.В.**

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

## **МОДИФІКОВАНИЙ МЕТОД ГЕНЕРАЦІЇ КЛЮЧІВ ДЛЯ АЛГОРИТМУ АСИМЕТРИЧНОГО ШИФРУВАННЯ RSA**

### **Abstract**

**Mykola Onai, assoc. prof., PhD; Oleksandr Kvitka, student**

#### ***Modified method of generation of encryption keys for asymmetric cryptosystem RSA***

*This paper concerns the task of acceleration of generation of encryption keys for asymmetric RSA cryptosystem with admissible losses of cryptographic strength. The proposed modification of the classical method is aimed to increase the speed of work by increasing prime numbers quantity which is used to generate encryption keys. The comparative analysis of efficiency of both the classical and the modified algorithms is conducted.*

### **Вступ**

На сьогоднішній день у криптографічних системах захисту інформації широко використовуються асиметричні алгоритми шифрування, серед яких одним з найпопулярніших можна назвати алгоритм RSA [1]. Розповсюдженими прикладами використання є протокол захисту SSL, мережевий протокол SSH, криптографічна програма PGP, а також цифровий підпис. Необхідність збереження надійності шифрування при постійному зростанні обчислювальної потужності сучасних технічних засобів вимагає збільшення довжини криптографічних ключів, що уповільнює роботу алгоритму. Дана тенденція породжує потребу пришвидшити роботу алгоритму.

Більшість існуючих модифікацій методу шифрування RSA направлені на спрощення математичного апарату, що лежить в основі його роботи. Зокрема це використання китайської теореми про залишки [2].

У даній роботі буде розглянуто модифікацію методу шифрування RSA, направлену на пришвидшення генерації ключів шифрування, що є однією з найбільш часозатратних операцій даного методу, шляхом зменшення довжини простих чисел, що для цього використовуються.



### Постановка задачі

Задача полягає в створенні модифікації існуючого методу шифрування RSA з метою зменшення часових витрат при роботі з актуальними з точки зору надійності ключами, тобто довжиною понад 1024 біта, при збереженні достатньої криптостійкості методу.

### Аналіз класичного алгоритму RSA

Класичний алгоритм RSA починається з вибору відкритого ключа [3]. Даний ключ складається з пари цілих чисел  $(e, n)$ . Число  $n$  являє собою добуток двох великих простих чисел  $p$  і  $q$ , що обираються довільно. Також обчислюється функція Ейлера:

$$\varphi(n) = (p - 1)(q - 1). \quad (1)$$

Число  $e$  – довільно обране просте число, що задовольняє наступним умовам:  $e < \varphi(n)$ ; НСД  $(e, \varphi(n)) = 1$ . Також обчислюється ключ розшифрування  $d$  – число, що є оберненим до числа  $e$ , тобто

$$ed \equiv 1 \pmod{\varphi(n)}. \quad (2)$$

Числа  $e$  та  $n$  зберігаються у загальнодоступному довіднику, а числа  $p$  і  $q$  залишаються у секреті. Повідомлення  $M$  перед шифруванням розкладається на цифрові блоки  $m_i$  довжиною, меншою за довжину  $n$ . Кожен блок шифрується окремо, в результаті отримуються цифрові блоки  $c_i$ , що разом утворюють зашифроване повідомлення  $C$ , що передається по відкритому каналу. Зашифрування відбувається за допомогою формули

$$C \equiv M^e \pmod{n}. \quad (3)$$

Розшифровується повідомлення, згідно з теоремою Ейлера, за допомогою піднесення шифрованого тексту  $C$  до степеня  $d$  за модулем  $n$ , тобто  $M \equiv C^d \pmod{n}$ .

### Аналіз модифікації алгоритму RSA

Запропонований модифікований метод полягає в тому, щоб збільшити кількість множників для обчислення  $n$ . Генерація простого числа необхідної розрядності є однією з найбільш часозатратних операцій. Тому збільшення їх кількості дозволить зменшити час генерації відкритого ключа  $n$ , зберігаючи задану довжину. Отже добуток  $n$  буде виглядати наступним чином:

$$n = \prod_{i=1}^k p_i, \quad (4)$$

де  $k$  – оптимальна кількість множників, при якій досягається найкращий часовий показник роботи алгоритму. Функція Ейлера буде виглядати як

$$\varphi(n) = \prod_{i=1}^k (p_i - 1). \quad (5)$$

## Експериментальні дослідження модифікованого алгоритму

Для перевірки роботи запропонованого модифікованого методу асиметричного шифрування проводились заміри швидкості обчислень під час генерації ключів, шифрування та дешифрування для різної кількості простих множників від 2 до  $\frac{n}{4}$ , тобто розрядність кожного окремого множника не менше 4 знаків. Результати проведеного аналізу наведено у вигляді графіків.

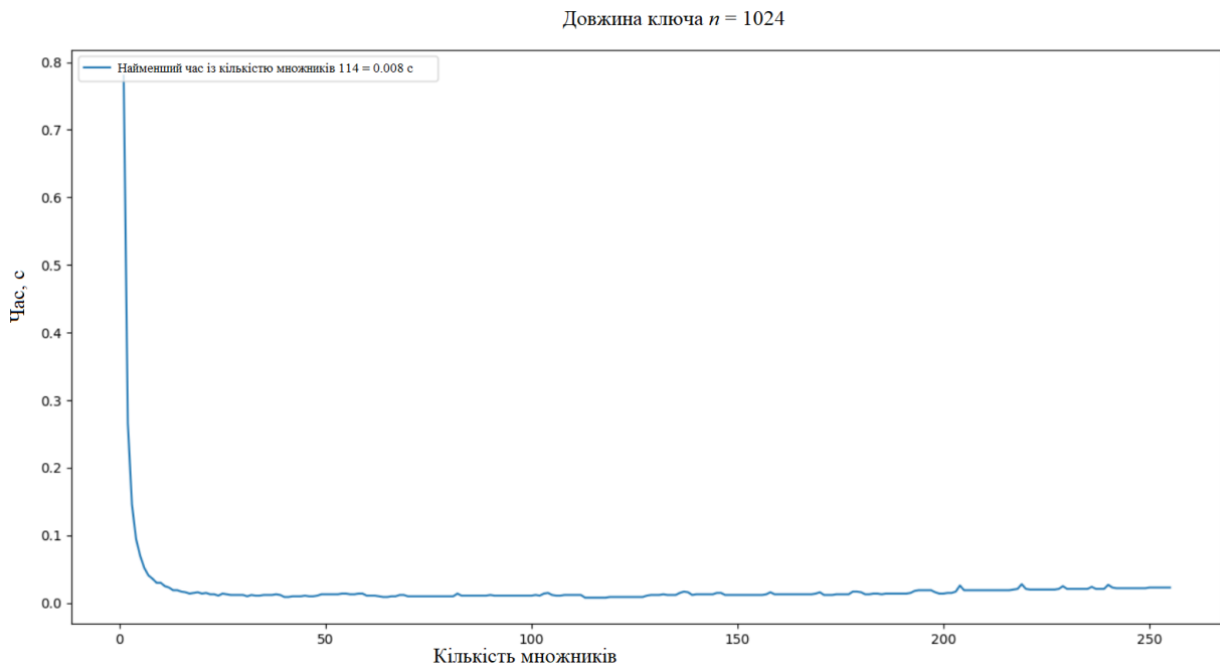


Рис. 1. Графік залежності часу роботи алгоритму від кількості простих множників для довжини ключа 1024 біт

Порівняння графіків залежності часу роботи класичного та модифікованого алгоритмів від довжини ключа демонструє значний приріст швидкості роботи у модифікованого методу. Для 1024 біт швидкість вища у 60 разів. Даний показник зростає із збільшенням довжини ключа.

Недоліком даного підходу є зниження складності факторизації відкритого ключа  $n$ . Аналіз показує, що ймовірність знаходження одного з простих чисел  $p$  і  $q$  у класичному алгоритмі RSA становить  $\frac{1}{2}$  [4]. Тоді ймовірність факторизації  $n$ , що є добутком  $k$  простих множників, буде дорівнювати  $1 - \frac{1}{2^{k-1}}$ . Тобто із збільшенням  $k$  складність факторизації значно падає.

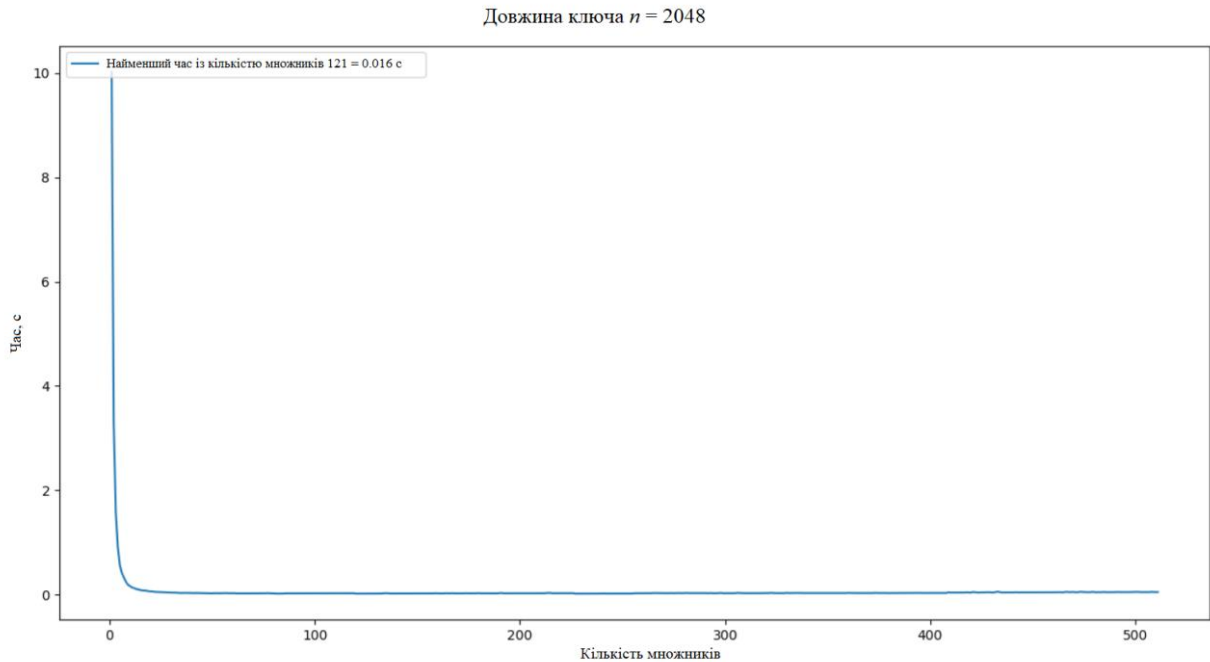


Рис. 2. Графік залежності часу роботи алгоритму від кількості простих множників для довжини ключа 2048 біт

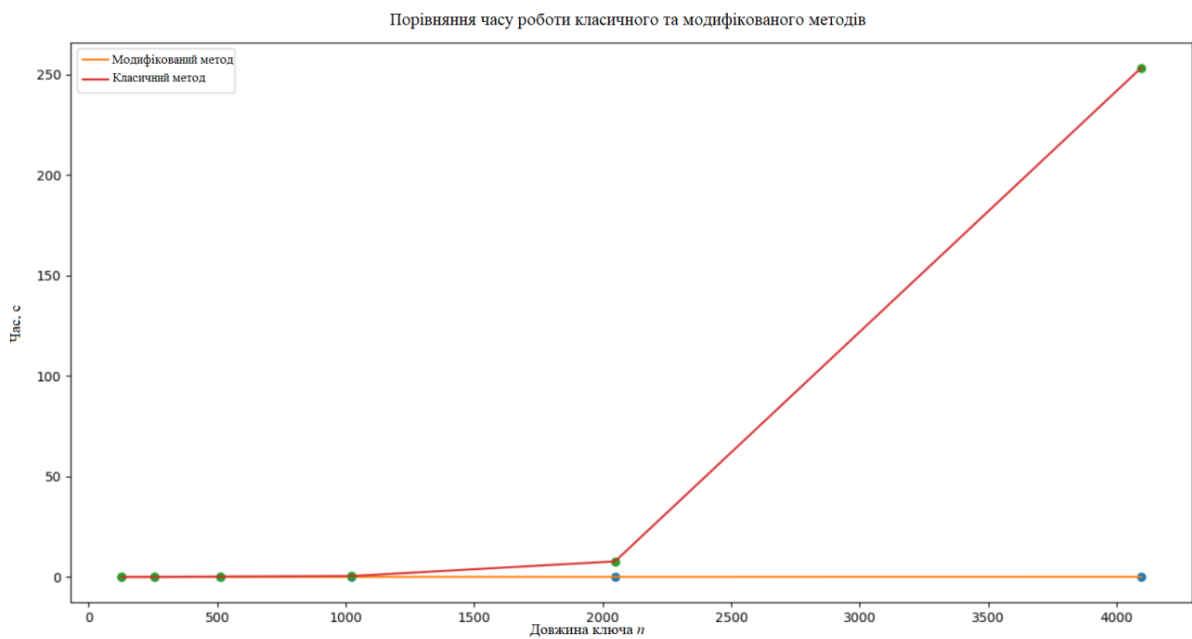


Рис. 3. Графіки залежності часу роботи класичного та модифікованого алгоритмів від довжини ключа

Однак, виключаючи атаку грубою силою, більшість методів криптоаналізу не придатні до зламу модуля з більше ніж двох простих чисел. Із збільшенням числа  $k$  атаки стають складнішими, що призводить до втрати їх ефективності, або стають непридатними в цілому.

Завдяки отриманим даним можна вивести правило вибору кількості простих чисел  $k$  для отримання відкритого ключа  $n$ . Найменші показники часу роботи було отримано, коли кількість множників  $k$  знаходилось у близько до 100, а далі починали дуже повільно зростати. Проте, через значну втрату криптостійкості, пропонується обирати  $k$  близьке до 10, де показники часових замірів вже достатньо близькі до мінімальних. Таким чином, криптостійкість наближено впаде у 2 рази.

Виходячи із отриманих результатів пропонується випадковим чином обирати число  $k$  таке, щоб  $k \in [5; 15]$ . Для підвищення криптостійкості методу пропонується випадковим чином обирати декілька  $k$ , що належить даному діапазону, таким чином, щоб прості множники  $p_i$  були різної довжини. В результаті для факторизації отриманого числа  $n$  буде необхідно знайти невідому кількість простих множників різної довжини, що значно ускладнить дану задачу.

## **Висновки**

У даній статті запропоновано модифікацію існуючого методу шифрування RSA, що дає значний вигреш у часі, у 60 і більше разів, при генерації ключів великої довжини, тобто понад 1024 біта. Серед недоліків даного методу варто зазначити втрату криптостійкості методу у 2 рази.

## **Література**

1. Rivest R., Shamir A., Adleman L. A method for obtaining digital signatures and public-key cryptosystems, Commun. ACM — New York City: ACM, 1978.
2. Нестеренко О. Введение в современную криптографию. Теоретико-числовые алгоритмы, 2011. – 190 с.
3. Онацкий А. В., Йона Л. Г. Асимметричные методы шифрования. – Модуль 2. Криптографические методы защиты информации в телекоммуникационных системах и сетях: учеб. пособие, 2011. – 148 с.
4. D. R. Stinson. Cryptography: Theory and Practice. CRC Press LLC, 1995. – 611 p.

**UDC 004.75**

**PhD, Assoc. Professor Mykola Onai; Master's student Haoling Liu**

**National Technical University of Ukraine  
"Igor Sikorsky Kyiv Polytechnic Institute"**

## **IMPROVEMENT OF KUBERNETES RESOURCE SCHEDULING ALGORITHM**

### **Abstract**

**Mykola Onai., PhD, Assoc. Professor; Haoling Liu, Master's student  
Improvement of Kubernetes resource scheduling algorithm**

*In 2013, with the rapid development of Docker and other container technologies, container-based virtualization technology has quickly become the first choice of major cloud computing vendors and cloud computing developers. Compared with virtual machines, container technology has smaller images, less resource consumption, and the advantages of flexible application deployment and fast startup speed. Many containers rely on container orchestration tools for management and control, which determine how containers interact. Container orchestration tools represented by Kubernetes have gradually become cloud native as a de facto standard, more and more microservices use Kubernetes for deployment and management [1]. The automatic deployment function of Kubernetes allows developers to easily deploy applications, and the automatic management function allows defined services to always run in the state expected by users and automatically expand. The function of shrinking makes it possible for the number of copies owned by the server to increase or decrease as the user access load changes.*

### **Introduction**

The Kubernetes default scheduling mechanism only considers the resource utilization of a single node and does not consider the load situation of the entire cluster. Therefore, this paper proposes a genetic algorithm-based Kubernetes resource scheduling algorithm for this problem. In the initial stage of the population, through random methods Initialize the population, introduce a check dictionary to check the individuals in the population, and repair the individuals that do not meet the configuration requirements; in the stage of calculating the fitness value, the standard deviation of the average load of the cluster is used as the objective function value, and the smaller the standard deviation, it means The cluster load is more balanced; use the roulette selection method to select good individuals to enter the next generation, and use the check dictionary to check and repair the individuals in the population again after the crossover and mutation operations. Kubernetes resource scheduling algorithm consists of two processes: pre-selection and optimization. The pre-selection

process needs to traverse all nodes, which is time-consuming, the improved scheduling algorithm proposes to directly optimize the number of nodes that meet the conditions without having to traverse all, which is expected to improve the scheduling efficiency [2]. For the optimization process, only the CPU and memory usage applied by the pod itself are considered, and the resource utilization of the node itself is not considered.

### **Research Objective**

Object of research is the processes of scheduling resources in Kubernetes in order to get load balancing.

### **Subject of research**

Subject of research is the resource scheduling method in Kubernetes  
Research objective: the development of a method and algorithm to achieve load balance in Kubernetes.

### **Algorithm description**

#### **1 Algorithm design**

The pre-selection process of Kubernetes' default algorithm needs to traverse all nodes. When there are too many nodes, the pre-selection will be time-consuming [3]. Therefore, only the number of nodes that meet the conditions can be selected in the pre-selection process, instead of polling all nodes, because nodes The resource utilization is changing at any time. We are concerned about the final equilibrium efficiency of the cluster, rather than a certain state in the middle. Polling all nodes and then scoring does not mean that the result is optimal. At the same time, the optimization process only considers the number of CPUs. And memory utilization, instead of paying attention to the node's own network utilization, disk utilization and other indicators, and today's Internet applications are complicated and complex, and CPU and memory indicators alone obviously cannot reflect the overall situation of the cluster. The optimization process will be carried out below. Improve, comprehensively consider the node's CPU, memory, network, disk, and other indicators to improve the existing optimization process algorithm model. For the optimization process, the overall idea of the improved resource scheduling algorithm is to rationally schedule pods to improve the load balancing efficiency of the entire cluster. For the calculation of load balancing efficiency, network utilization is also a factor that cannot be ignored for Internet applications, so this article is in the original Some load balancing calculations add network utilization indicators to calculate the final score by comprehensively considering the four dimensions of CPU, memory, network, and disk IO. The calculation formula is:

$$\text{Score} = (1 - L(i) / S(i)) * 10$$

$L(i)$  represents node load,  $S(i)$  represents node performance.

The calculation formula of  $L(i)$  is:

$$L(i) = L(c) + L(m) + L(n) + L(d)$$

$L(c)$  represents CPU utilization,  $L(m)$  represents memory utilization,  $L(n)$  represents network utilization, and  $L(d)$  represents disk utilization.

$$S(i) = n * S(c) + S(m) + S(n) + S(d),$$

$n$  is the number of CPU cores,  $S(c)$  is the CPU frequency,  $S(m)$  is the memory capacity,  $S(n)$  is the network speed, and  $S(d)$  is the disk IO rate.

## 2 Algorithm process

The running logic of the custom scheduler is a polling process: first get all the pods whose tags are my-scheduler and have not yet bound nodes, and then run the two scripts computeL.sh and computeS.sh on all worker nodes to obtain resource utilization. Rate and node performance, and then calculate the score of each worker node, and finally bind the currently scheduled pod with the worker node with the highest score. This binding process is completed by sending a post request to the api server, thereby completing the final scheduling.

The pseudo code as follows:

Input: CPU, memory, IO, network utilization, number and frequency of CPU cores, memory capacity, network bandwidth, and disk capacity of each node.

Output: The binding information between the node and the pod.

```

CHOSEN← null
SERVER← api server
1: while true do
    2: for PODNAME in unscheduled where pod.name==my-scheduler do
        3: for node in nodes do
            4: L← computeL()
            5: S← computeS()
            6: score=(1-L/S)* 10
            7: CHOSEN← node[max(score)]
        8: end for
        9: curl --data {name: $CHOSEN} http:// $SERVER/api/v1/namespaces
        10: /default/pods/$PODNAME/binding/
        11: printf "binding $PODNAME to $CHOSEN "
    12: end for
    13: sleep 1
14: end while

```

## Conclusions

Kubernetes is the core orchestration tool of the microservice architecture. In default method, the preselection process of its default resource scheduling algorithm must traverse all nodes, which is time-consuming when the number of

nodes is large [4]. The improved resource scheduling algorithm optimizes the preselection process and proposes to focus on the final cluster resource utilization, without paying attention to the intermediate state, so there is no need to traverse all nodes to improve resource scheduling efficiency. At the same time, only the two indicators of CPU and memory are considered for the optimization process, and only the CPU and memory requested by the pod, without considering the node itself. For the problem of performance indicators, in my method, because adds network utilization and rate other than CPU, memory and disk on the default of the existing resource scheduling algorithm and implements a custom scheduler, so it can improve resource scheduling algorithm to improve the overall load balancing efficiency of the cluster.

## References

1. Bernstein D. Containers and cloud: From LXC to docker to kubernetes. *IEEE Cloud Computing*, 2014, 1(3): 81–84.
2. Burns B, Grant B, Oppenheimer D, et al. Borg, omega, and kubernetes. *Communications of the ACM*, 2016, 59(5): 50–57.
3. Rodriguez M, Buyya R. Container Orchestration With Cost- Efficient Autoscaling in Cloud Computing Environments. *Handbook of Research on Multimedia Cyber Security*. IGI Global, 2020. 190–213.
4. Scheduler extender. [https://github.com/kubernetes/community/blob/master/contributors/design-proposals/scheduling/scheduler\\_extender.md](https://github.com/kubernetes/community/blob/master/contributors/design-proposals/scheduling/scheduler_extender.md). [2019-12-06].



Д.т.н., доцент Сулема Є.С.; аспірант Дичка А. І.

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

## АНАЛІЗ КОРЕКТУВАЛЬНИХ ВЛАСТИВОСТЕЙ МНОГОЗНАЧНИХ КОДІВ ХЕМІНГА

### Abstract

**Yevgeniya S. Sulema, DSc, assoc. prof.; Andrii I. Dychka, PhD student**  
*Analysis of the corrective properties of multivalued Hamming codes*

*Based on the results of statistical studies, the corrective capabilities of multivalued Hamming codes compared to binary ones are analyzed. The technique of construction of incomplete multivalued Hamming codes, and also a way of their application for errors control in system of automatic identification on the basis of multicolor bar codes are offered.*

### Вступ

Коди Хемінга знаходять широке застосування в галузі телекомунікацій, зокрема в різноманітних технічних системах передачі та/або зберігання даних [1]. У таких системах зазвичай використовують двійкові коди Хемінга.

Але дедалі частіше з метою забезпечення завадостійкості даних виникає потреба в застосуванні многозначних (недвійкових) кодів Хемінга.

Потреба в многозначних кодах Хемінга може виникати тоді, коли первинні дані є недвійковими, наприклад, при обробленні багатоколірних зображень. Так, якщо в зображенні використовують  $q$  кольорів ( $q > 2$ ), то розбивши зображення на прямокутні елементи однакового розміру (чарунки), можна оперувати з двовимірною матрицею, елементами якої є числа з діапазону  $0, 1, 2, \dots, q - 1$ , що позначають кольори чарунок. Відповідно, для забезпечення контролю помилок в масивах  $q$ -кових даних доцільно використовувати многозначні ( $q$ -кові) коректувальні коди, найпростішими з яких є многозначні коди Хемінга.

### Постановка задачі

Необхідно з'ясувати особливості, зокрема коректувальні властивості, многозначних кодів Хемінга у порівнянні з двійковими кодами та оцінити доцільність їх застосування для контролю помилок при обробленні багатоколірних зображень.

## Узагальнення кодів Хемінга

Код Хемінга у будь-якій системі числення можна задати перевіркою матрицею [2]. Перевірна матриця  $q$ -кового  $(n, r)$ -коду Хемінга має вигляд:

$$H_{(n,k)}^{(q)} = \left\| \begin{array}{cccccccc} h_{11} & h_{12} & \dots & h_{1k} & 1 & 0 & \dots & 0 \\ h_{21} & h_{22} & \dots & h_{2k} & 0 & 1 & \dots & 0 \\ \vdots & & & & & & & \\ h_{r1} & h_{r2} & \dots & h_{rk} & 0 & 0 & \dots & 1 \end{array} \right\| = \left\| \begin{array}{c} H_{r \times k} \\ \vdots \\ I_r \end{array} \right\|, \quad (1)$$

де  $h_{ij} \in \{0, 1, 2, \dots, q-1\}$ ,  $n$  – довжина кодових слів,  $k$  – кількість інформаційних розрядів, а  $r = n - k$  – кількість контрольних розрядів в кодових словах.

Стовпці матриці  $H_{(n,k)}^{(q)}$  є ненульовими  $r$ -розрядними послідовностями, у яких перша ненульова компонента дорівнює одиниці. Такий спосіб формування перевіркою матриці забезпечує лінійну незалежність стовпців [3]. Максимальна кількість стовпців перевіркою матриці дорівнює  $(q^r - 1)/(q - 1)$ .

Параметри  $k$  та  $r$  пов'язані між собою співвідношенням

$$k \leq (q^r - 1)/(q - 1) - r. \quad (2)$$

Якщо у співвідношенні (2) виконується рівність, то код Хемінга називають повним. Обмежимося випадком коли  $q$  – просте число.

Якщо задано  $k$ -розрядне  $q$ -кове інформаційне слово  $A = (a_1 a_2 \dots a_k)$ , то  $n$ -розрядне кодове слово  $C = (c_1 c_2 \dots c_n)$  коду Хемінга отримують як  $C = A : (A \cdot H_{r \times k}^T)$ , тобто до  $k$  інформаційних розрядів долучають  $r$  контрольних розрядів,  $r = n - k$ . Операції множення та додавання, що мають місце при множенні матриць - вектор-рядка  $A$  на транспоновану підматрицю  $H_{r \times k}$ , слід виконувати в полі Галуа  $GF(q)$ .

При декодуванні прийнятого  $n$ -розрядного слова  $C' = (c'_1 c'_2 \dots c'_n)$  обчислюють  $r$ -розрядний синдром помилки  $S = (S_1 \dots S_r)$ , компоненти  $S_i$  якого ( $i = 1, 2, \dots, r$ ) отримують як

$$S_i = \left( \sum_{j=1}^k c'_j h_{ij} \right) - c'_{k+i}. \quad (3)$$

У виразі (3) операції слід виконувати за модулем  $q$ .

Нульовий синдром свідчить про те, що в прийнятому слові помилки відсутні.

Якщо  $s \neq 0$ , то знаходять модифікований синдром  $s^m = (s_1^m \dots s_r^m)$ , де  $s_i^m = (s_i / e) \bmod q$ , а  $e$  – величина помилки ( $e = 1, 2, \dots, q - 1$ ). Великою помилки є перша ненульова компонента синдрому, яку знаходять послідовно аналізуючи  $s_1, s_2, \dots, s_r$  зліва направо. Модифікований синдром  $s^m$  є локатором помилки.

Якщо слово  $c$  містить однократну помилку в розряді з номером  $v$ , де  $v = 1, 2, \dots, n$ , то  $s^m$  збігається з  $v$ -м стовпцем матриці  $H_{(n,k)}^{q}$ , вказуючи таким чином на місцезнаходження помилки. Виправлення цього розряду виконують так:  $c_v = (c'_v - e) \bmod q$ .

Код Хемінга, що заданий перевіреною матрицею  $H_{(n,k)}^{q}$ , забезпечує виправлення лише однократних помилок у прийнятих словах, і, отже, має мінімальну кодову відстань  $d_{\min} = 3$ .

У табл.1 наведено деякі повні  $(n, k)$ -коди Хемінга коли  $q$  є простим числом.

### Приклад побудови трійкового коду Хемінга

Побудуємо повний трійковий  $(13,10)$ -код Хемінга (див. табл.1). Для такого коду кількість  $r$  контрольних розрядів у кодових словах дорівнює 3 ( $r = n - k = 13 - 10 = 3$ ).

Можна записати 13 трирозрядних послідовностей, у яких перша ненульова компонента дорівнює одиниці:

1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	1	1	1	2	2	2	1	1	1	0
0	1	2	0	1	2	0	1	2	0	1	2	1

Таблиця 1

Деякі повні  $q$ -кові  $(n, k)$ -коди Хемінга

Основа системи числення			
$q = 2$	$q = 3$	$q = 5$	$q = 7$
Вид скінченного поля			
GF(2)	GF(3)	GF(5)	GF(7)
(7,4) –	(4,2) –	(6,4) –	(8,6) –
(15,11) –			
(31,26) –	(13,10) –	(31,28) –	(57,54) –

Змінивши порядок слідування послідовностей так, щоб наприкінці (справа) опинились ті послідовності, які у своєму складі містять лише одну одиницю, отримаємо перевірну матрицю трійкового (13, 10)-коду Хемінга з параметрами  $k = 10$  – кількість інформаційних розрядів у кодових словах,  $n = 13$  – довжина кодових слів;

$$H_{(n,k)}^{''q''} = \left\| \begin{array}{cccccccccc|ccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 & 1 & 1 & 0 & 1 & 0 \\ 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 1 & 2 & 0 & 0 & 1 \end{array} \right\| = \left\| H_{3 \times 10} \dot{I}_3 \right\|,$$

де  $H_{3 \times 10}$  – підматриця розмірності  $3 \times 10$ ,  $I_3$  – одинична підматриця розмірності 3.

### Побудова кодів Хемінга з $d_{\min} = 4$

Розглянемо методику побудови двійкових кодів Хемінга з мінімальною кодовою відстанню  $d_{\min} = 4$ . Повні двійкові  $(n, k)$ -коди Хемінга з  $d_{\min} = 3$  мають параметри:  $n = 2^r - 1$ ,  $k = n - r = 2^r - 1 - r$ . Їх задають за допомогою перевірної матриці, стовпцями якої є всі ненульові двійкові послідовності довжини  $r$ .

Наприклад, виходячи з множини всіх можливих ненульових 3-розрядних двійкових послідовностей, можна побудувати перевірну матрицю  $H_{(7,4)}^{''2''}$  повного (7, 4)-коду Хемінга з  $d_{\min} = 3$  (рис.1).

$$\begin{array}{cccccccc|ccc} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{array} \Rightarrow \begin{array}{ccc} a & & b \end{array}$$

$$H_{(7,4)}^{''2''} = \left\| \begin{array}{cccc|ccc} a_1 & a_2 & a_3 & a_4 & b_1 & b_2 & b_3 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{array} \right\| = \left\| H_{3 \times 4} \dot{I}_3 \right\|$$

$\epsilon$

Рис.1. Побудова перевірної матриці повного (7, 4)-коду Хемінга з  $d = 3$ :  
 $a$  – можливі ненульові 3-розрядні двійкові послідовності;  
 $b$  – виділення одиничної матриці;  
 $\epsilon$  – перевірна матриця повного (7, 4)-коду Хемінга.

Характерною особливістю двійкових кодів Хемінга є те, що на основі коду Хемінга з  $d_{\min} = 3$  можна утворити код Хемінга з  $d_{\min} = 4$ , який здатний виправляти в прийнятих словах однократні помилки та виявляти двократні помилки (загалом – помилки парної кратності). Для цього в коді з  $d_{\min} = 3$  запроваджують додатковий контрольний розряд, який дорівнює сумі за  $\text{mod } 2$  всіх розрядів слова – інформаційних та контрольних.

Наприклад, для  $(7, 4)$ -коду додатковий контрольний розряд ( $t_4$ ) обчислюють так (див. рис.1):

$$b_4 = (a_1 + a_2 + a_3 + a_4 + b_1 + b_2 + b_3) \text{ mod } 2 = (a_1 + a_2 + a_3 + a_4 + (a_1 + a_2 + a_3) \text{ mod } 2 + (a_2 + a_3 + a_4) \text{ mod } 2 + (a_1 + a_3 + a_4) \text{ mod } 2) \text{ mod } 2 = a_1 + a_2 + a_4 \Rightarrow 1 \ 1 \ 0 \ 1.$$

Це означає, що підматрицю  $H_{3 \times 4}$  слід доповнити рядком 1 1 0 1, а одиничну підматрицю розмірності три замінити на одиничну підматрицю розмірності 4.

Тоді отримаємо перевірну матрицю  $(8, 4)$ -коду з  $d_{\min} = 4$ :

$$H_{(8,4)}^{''2''} = \left\| \begin{array}{cccc|cccc} a_1 & a_2 & a_3 & a_4 & b_1 & b_2 & b_3 & b_4 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{array} \right\| = \left\| H_{4 \times 4} \vdots I_4 \right\|$$

$(8, 4)$ -код Хемінга здатний виправляти 1-кратні та виявляти 2-кратні помилки у 8-розрядних двійкових словах.

Застосуємо такий самий прийом для збільшення  $d_{\min}$ , наприклад, для повного п'ятіркового  $(6, 4)$ -коду Хемінга з  $d_{\min} = 3$ , який задають перевірною матрицею

$$H_{(6,4)}^{''5''} = \left\| \begin{array}{cccc|cc} a_1 & a_2 & a_3 & a_4 & b_1 & b_2 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 2 & 3 & 4 & 0 & 1 \end{array} \right\|.$$

Запровадимо додатковий контрольний розряд ( $b_3$ ), що дорівнює сумі за модулем п'ять всіх розрядів слова:

$$b_3 = (a_1 + a_2 + a_3 + a_4 + b_1 + b_2) \text{ mod } 5 = (a_1 + a_2 + a_3 + a_4 + (a_1 + a_2 + a_3 + a_4) \text{ mod } 5 + (a_1 + 2a_2 + 3a_3 + 4a_4) \text{ mod } 5) \text{ mod } 5 = 3a_1 + 4a_2 + a_4 \Rightarrow 3 \ 4 \ 0 \ 1.$$

Тоді отримаємо перевірну матрицю п'ятіркового  $(7, 4)$ -коду Хемінга

$$H_{(7,4)}^{(5)} = \begin{vmatrix} a_1 & a_2 & a_3 & a_4 & b_1 & b_2 & b_3 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 2 & 3 & 4 & 0 & 1 & 0 \\ 3 & 4 & 0 & 1 & 0 & 0 & 1 \end{vmatrix}.$$

Дослідження показують, що цей код виправляє всі можливі однократні помилки, а 2-кратні помилки виявляє лише частково, і тому його мінімальна кодова відстань не дорівнює 4. Отже, запровадження додаткового контролю всіх розрядів кодового слова за модулем  $q$  у випадку коли  $q > 2$  не приводить до збільшення мінімальної кодової відстані многозначного коду Хемінга на одиницю (з  $d_{\min} = 3$  до  $d_{\min} = 4$ ).

Таким чином, на практиці доцільно застосовувати многозначні ( $q > 2$ ) коди Хемінга лише з  $d_{\min} = 3$ .

### Неповні коди Хемінга

При використанні кодів Хемінга можливі випадки коли значення параметра  $k$  (кількість інформаційних розрядів) менше (іноді – істотно менше) за величину  $(q^r - 1)/(q - 1)$  (див. (2)).

У подібних ситуаціях використовують неповні (скорочені) коди Хемінга. Неповний код Хемінга отримують з відповідного повного коду шляхом викреслювання потрібної кількості стовпців у підматриці  $H_{r \times k}$  перевірної матриці  $H_{(n,k)}^{(q)}$  (див. (1)); при цьому одинична підматриця  $I_r$  змін не зазнає. Після викреслювання деяких стовпців у підматриці  $H_{r \times k}$  коректувальні властивості коду залишаються незмінними, і з'являється здатність додатково виявляти багатократні помилки – залежно від кількості викреслених стовпців.

Наприклад, викресливши два останні стовпці в підматриці  $H_{3 \times 10}$  перевірної матриці  $H_{(13,10)}^{(3)}$ , отримаємо трійковий неповний (11,8)-код Хемінга, який забезпечує виправлення всіх однократних помилок у трійкових 11-розрядних словах, а також часткове виявлення багатократних помилок, зокрема 18% 2-кратних та 16% 3-кратних помилок (рис. 2).

У рамках даного дослідження розроблено програмний продукт, який дозволяє статистично оцінювати коректувальні властивості многозначних кодів Хемінга, включаючи й двійкові коди (рис. 3).

## Застосування многозначних кодів Хемінга

Оскільки многозначні коди Хемінга не забезпечують гарантованого виявлення двократних (парної кількості) помилок у кодових словах – як двійковий код Хемінга з  $d_{\min} = 4$ , то їх самостійне застосування в системах контролю помилок може виявитись малоефективним. Однак, цей недолік нівелюється, якщо многозначний код Хемінга застосувати в парі з іншим многозначним кодом, який здатний виправляти два види спотворень – помилки та стирання, де під помилкою розуміють таке спотворення, коли декодеру не відоме ні місцезнаходження спотворення, ні його величина; а під стиранням – випадок коли місцезнаходження спотворення відоме, а невідомим є лише його величина. Прикладом такого коду слугує коректувальний код Ріда-Соломона.

Об'єктом спільного застосування многозначних кодів Хемінга та коду Ріда-Соломона можуть бути багатоколірні штрихові коди. У таких штрихових кодах з метою гарантування надійного зчитування даних доцільно запровадити дворівневу систему забезпечення завадостійкості (рис. 4) – коли в межах багатоколірних штрихкодових знаків (ШК-знаків), з яких складається штриховий код, використовують многозначний код Хемінга, а на рівні всього штрихкодового зображення – код Ріда-Соломона [4].

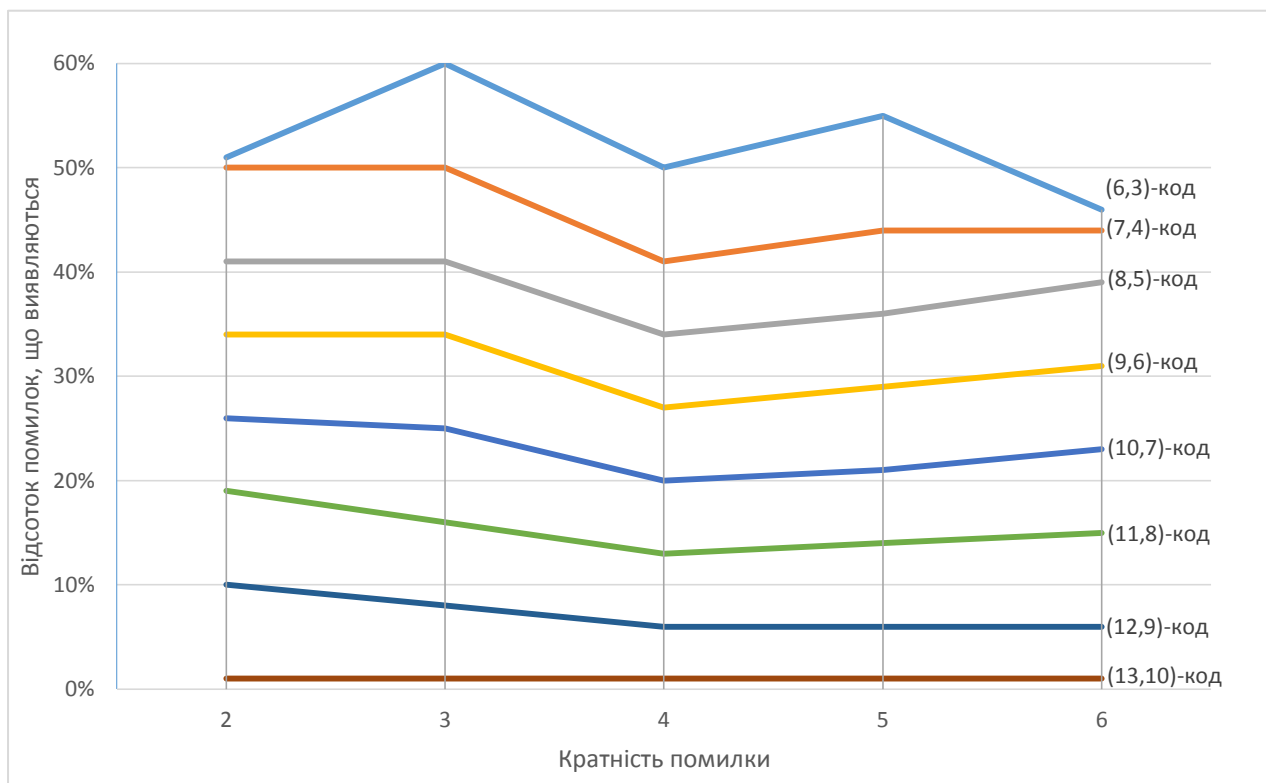


Рис.2. Здатність виявляти багатократні помилки неповними трійковими кодами Хемінга похідними від повного (13,10)-коду

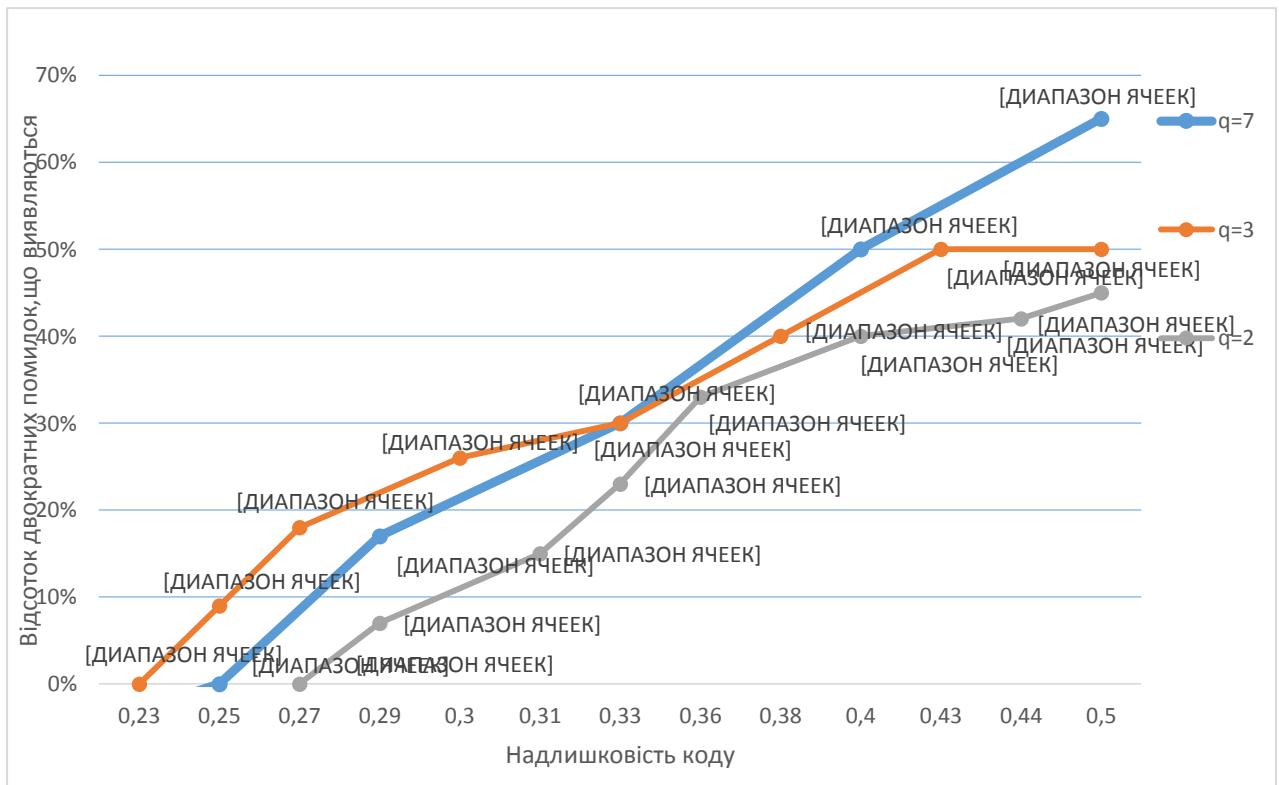


Рис.3. Здатність виявляти 2-кратні помилки неповними двійковими ( $q=2$ ), трійковими ( $q=3$ ), сімковими ( $q=7$ ) кодами Хемінга.

Примітка: (15,11) – повний двійковий код Хемінга; (13,10) – повний трійковий код; (8,6) – повний сімковий код Хемінга

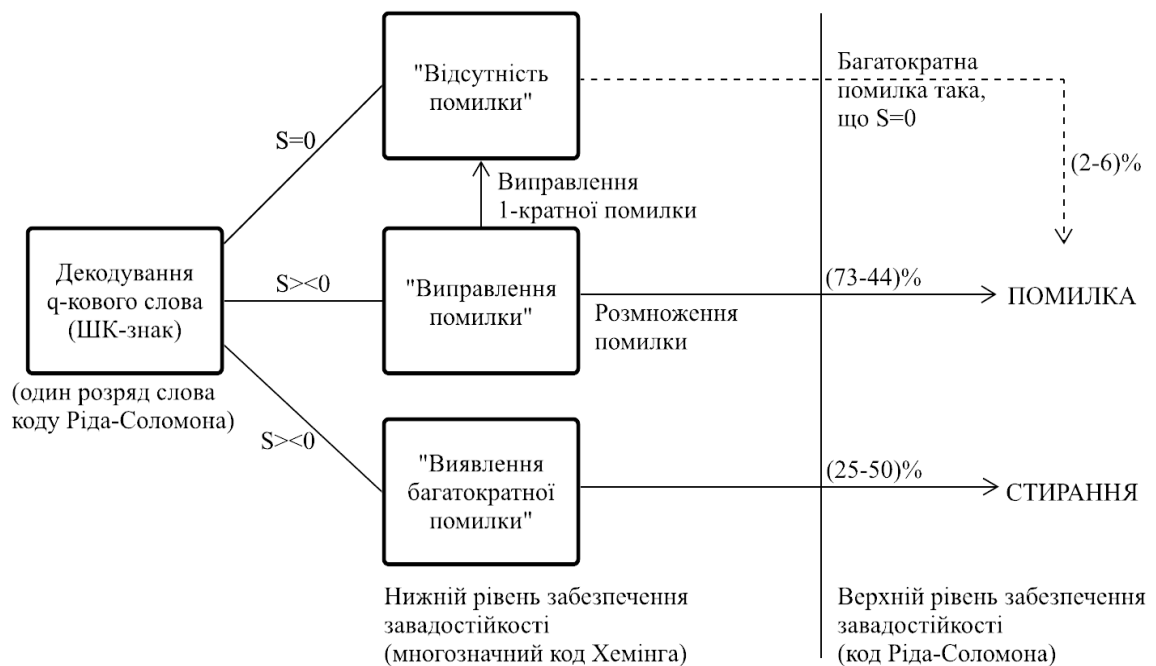


Рис.4. Дворівневе забезпечення завадостійкості багатоколірного штрихового коду



## Висновки

За фіксованого рівня надлишковості многозначні коди Хемінга дозволяють виявляти більший відсоток багатократних помилок порівняно з двійковими кодами Хемінга. Проте, їх недоліком є те, що вони не забезпечують 100%-кового виявлення двократних помилок. Однак, спільне (взаємодоповнювальне) застосування многозначного коду Хемінга та коду Ріда-Соломона, наприклад, у багатоколірному штриховому кодуванні, в середньому на 12-25% підвищує завадостійкість багатоколірних штрихових кодів.

Подальші дослідження слід зосередити на пошуку процедури побудови многозначних кодів Хемінга з мінімальною кодовою відстанню  $d_{\min} = 4$ .

## Література

1. *Blahut R. Theory and Practice of Error Control Codes / Richard E. Blahut // MA: Addison-Welsey, 1983. –500 p.*
2. *Peterson W. Error-Correcting Codes / W. Welsey Peterson, E.J. Weldon, Jr. // – Second Edition: MIT, 1972. –549 p.*
3. *Касами Т. Теория кодирования / Т. Касами, Н. Токура, Е. Ивадари, Я. Инагаки. – М.:Мир, 1978. –576 с.*
4. *Є.С. Сулема, М.В. Онай, А.І. Дичка. Алгоритмічне забезпечення завадостійкості багатоколірних штрихкодів на основі поля  $GF(p)$  // KPI Science News, 2021 / 1. С. 50 – 62*

УДК 004.89 : 004.93'1

Магістрант Телефус І.А., д.т.н., доцент Сулема Є.С.

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

## ОПТИЧНЕ РОЗПІЗНАВАННЯ РУКОПИСНИХ ТЕКСТІВ УКРАЇНСЬКОЮ МОВОЮ З ВИКОРИСТАННЯМ НЕЙРОННИХ МЕРЕЖ

### Abstract

**Illia Telefus, Master student; Yevgeniya Sulema, DSc, Assoc. Prof.**  
*Handwritten Ukrainian text recognition using neural networks*

*This paper deals with the task of Ukrainian handwritten text recognition. The classical OCR algorithms are not widely used for cryic languages. The proposed heuristic allows to speed up the recognition process while looking for defects on source image on the preprocessing step. The comparative analysis of efficiency of both the classical and the modified OCR algorithms is fulfilled. The ways for further research are proposed as well.*

### Вступ

Розпізнавання тексту на зображеннях є важливим завданням машинного навчання, оскільки дозволяє зручно взаємодіяти з даними: редагувати, аналізувати їх, виконувати пошук слів чи фраз, а також оцифровувати рукописи.

Загалом програмна система розпізнавання тексту повинна відповідати на такі два запитання: «де на зображенні текст?», «що означає цей текст?» Результатом використання такої системи буде автоматично виявлений текст і символи, що можуть бути розпізнані. На сьогодні, завдяки використанню сучасних досягнень комп'ютерних технологій, розроблені нові методи обробки зображень і розпізнавання образів.

У статті пропонується евристика, що дозволяє прискорити процес попередньої обробки зображення нейронною мережею, а саме підвищити швидкість пошуку дефектів на етапі попередньої обробки вхідного зображення.

### Постановка задачі

Мета роботи полягає в удосконаленні процесу попередньої обробки зображень, з використанням запропонованої евристики, таким чином, щоб

процес пошуку дефектів відбувався швидше, але не вимагав великої кількості додаткових обчислювальних ресурсів.

### Опис системи

Для побудови програмної системи розпізнавання рукописного тексту реалізувати такі основні етапи. Першим етапом є відокремлення тексту від інших елементів зображення. В результаті можуть бути отримані символи чи їх координати. На другому етапі необхідно виконати обробку зображення, а також можна виконувати «нарізку» символів із зображення для подальшого розпізнавання. Третім етапом є виділення ключових особливостей, які будуть використовуватись при розпізнаванні символів. Останнім, четвертим, етапом є власне розпізнавання символів [1-4].

На рис. 1 зображено основні засоби для побудови системи розпізнавання рукописного тексту, де на вході система отримуватиме певне зображення, а на виході демонструватиме розпізнаний текст.

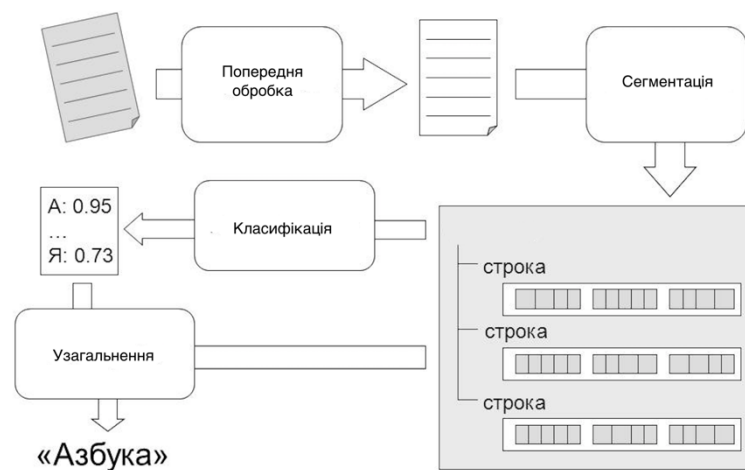


Рис. 1. Етапи розпізнавання рукописних символів

Оскільки зображення, які необхідно розпізнати, нерідко мають дефекти, необхідно мінімізувати кількість цих дефектів. Для цього можна використовувати медіанний фільтр, фільтр Гауса, метод Отсу та інші. Також можливе комбінування цих підходів.

В цій статті пропонується евристика, яка полягає у наступному. Якщо коефіцієнт форми області (відношення квадрата її периметра до її площі) менше деякого значення, а площа — більша деякого значення, то ця область є дефектом на зображенні. Використання комбінації методу Отсу та даної евристики дозволяє збільшити швидкість та точність вилучення дефектів із тексту, а також допомагає виділяти компоненти зв'язності, на основі яких буде проведена класифікація компонентів як частин тексту чи

дефектів. Проте, оскільки дана евристика отримана експериментальним шляхом на датасеті із 500 зображень, значення для порівняння необхідно уточнювати, адже вони не є еталонними. На рис. 2 наведено зображення до та після використання методу Отсу.

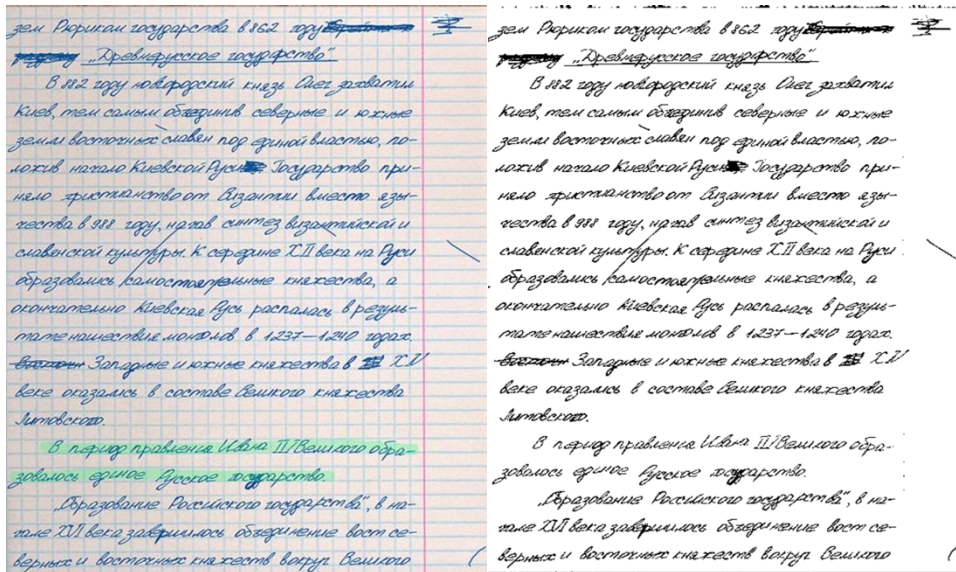


Рис. 2. Зображення до та після використання методу Отсу (зліва направо)

На рис 2. (ліва частина) введемо умовну одиницю виміру – клітинку, позначимо її 1. Тоді, використовуючи запропоновану евристику, розрахуємо коефіцієнт замальованої області в лівому куті:

$$P = (a + b) * 2 = (1 + 4) * 2 = 10; P^2 = 100$$

$$S = a * b = 1 * 4 = 4;$$

$$\frac{P^2}{S} < 30; S > 1$$

Таким чином, знаходимо, що відношення квадрата периметра до площі менше першого експериментального значення, а площа більша другого експериментального значення, тому можемо вважати даний фрагмент дефектом.

Для опису роботи розробленої системи створено UML діаграми, що описують основні варіанти використання та акторів. Діаграму прецедентів зображено на рис. 3.



Рис. 3. Діаграма прецедентів

Насамперед користувачу необхідно обрати зображення для обробки. Для цього в консольному інтерфейсі програмної системи користувач має ввести шлях до зображення в операційній системі. Далі запускається процес обробки зображення, що складається з кількох етапів. Коли процес розпізнавання завершився, користувач бачить у вікні консолі розпізнаний текст, а також запитання про те, куди його зберегти та у якому форматі.

## Висновки

При порівнянні швидкості роботи розроблюваної програмної системи без використання евристики виявлено, що швидкість обробки датасету із 500 зображень (без часу витраченого на навчання нейронної мережі) складає 3 хвилини. Після додавання етапу із використанням евристики час зменшився на 15 секунд. Подальші експерименти показали, що швидкість обробки зросла у середньому на 10%.

Варто зауважити, оскільки для аналізу запропонованої евристики використовувався порівняно невеликий набір даних, то доцільно розширити датасет та повторно провести навчання нейронної мережі.

Також важливим завданням, пов'язаним із розробкою системи, є збір даних для аналізу.

### **Література**

1. Paramud Y., Yarkun V. Algorithmic and software means of handwritten symbol recognition // Bulletin of the National University “Lviv Polytechnic” “Computer Systems and Networks” No 881, Lviv, Ukraine. 2017. – P. 98–106.
2. Yarkun V., Paramud Y. Algorithmic and software synchronization of information exchange // Bulletin of the National University “Lviv Polytechnic” “Computer Systems and Networks” No 857, Lviv, Ukraine. 2016. – P.111–118.
3. Datong Chen, Jean-Marc Odobez, Herve Boulard. Text detection and recognition in images and video frames // Pattern Recognition journal, 2003. – P. 595–608.
4. Convolutional Neural Network [Electronic resource]/ Stanford. – Access mode: <http://ufldl.stanford.edu/tutorial/supervisedConvolutional/NeuralNetwork/>.

**D.Sc., Professor Tereikovskiy Ihor and Student Tianbang Wang**

**National Technical University of Ukraine  
“Igor Sikorsky Kyiv Polytechnic Institute”**

## **SELECTION OF PHONEMES IN THE VOICE SIGNAL USING NEURAL NETWORKS**

### **Анотація**

**Д.т.н, професор Терейковський І.А., магістрант Тяньванг Ван**

***Виділення фонем в голосовому сигналі за допомогою нейронних мереж***

*У статті запропоновано модифікований підхід до застосування нейронної мережі для виділення меж окремих фонем у голосовому сигналі. Показано перспективність впровадження в систему розпізнавання рекурентних нейронних мереж. Розроблена архітектура системи виділення фонем. Показано шляхи подальших досліджень щодо підвищення точності виділення фонем.*

### **Introduction**

Speaking to a machine, making it understand what you're saying, has long been a dream. With the progress of data processing technology and the rapid popularization of mobile Internet, computer technology has been widely used in various fields of society, followed by the generation of massive data, among which voice data has attracted more and more attention. Therefore, it is necessary to adopt relevant methods for software development to solve the problem of voice signal recognition.

Phoneme is the smallest phonetic unit divided according to the natural properties of speech. It is analyzed according to the pronunciation actions in syllables. One action constitutes a phoneme. Our goal is to develop an application program for computer, through the input of phoneme parameters, to train the corresponding neural network model, so as to achieve the purpose of relatively accurate recognition of phoneme.

### **Formulation of the problem**

The main goal of this study is to ensure effective recognition of phonemes in the voice signal.

## **Research Objective**

The main objective of this project is to select phonemes in the voice signal, and convert them to digitized data.

## **Terminology**

Neural network - an interconnected assembly of simple processing elements, units or nodes, whose functionality is loosely based on the animal neuron. The processing ability of the network is stored in the interunit connection strengths, or weights, obtained by a process of adaptation to, or learning from, a set of training patterns.

## **Description of the general approach**

**Discrete Fourier Transform.** Discrete Fourier Transform (DFT) is the most basic method of signal analysis. Fourier Transform is the core of Fourier analysis. It transforms signals from time domain to frequency domain, and then studies the spectral structure and variation law of signals.

**Fast Fourier Transform.** The fast methods to calculate the discrete Fourier transform are FFT algorithm extracted by time and FFT algorithm extracted by frequency. The former arranges the time domain signal sequence according to even and odd, while the latter arranges the frequency domain signal sequence according to even and odd. They all rely on two characteristics: periodicity; The second is symmetry, where the \* stands for conjugate. In this way, the calculation of discrete Fourier transform can be divided into several steps, and the calculation efficiency is greatly improved.

**Deep Neural Networks.** Neural network is an extension based on perceptron, while DNN can be understood as a neural network with many hidden layers. Multi-layer neural network and deep neural network DNN actually refers to the same thing. DNN is sometimes called multi-layer Perceptron (MLP).

According to the position of different layers in DNN, the neural network layers inside DNN can be divided into three types: input layer, hidden layer and output layer, as shown in the example below. Generally speaking, the first layer is the input layer, the last layer is the output layer, and the middle layer is hidden layer.

**Time Delay Neural Network.** TDNN is a convolutional neural network applied to speech recognition problems. It uses FFT preprocessed speech signals as



input, and its hidden layer is composed of two one-dimensional convolution kernels to extract translation invariant features in the frequency domain

**Speech recognition model.** The speech recognition system model is composed of two parts: the acoustic model corresponds to the probability calculation of speech to phoneme, and the language model corresponds to the probability calculation of phoneme to text.

A continuous speech recognition system can be roughly composed of four parts: feature extraction, acoustic model, speech model and decoding part. The specific process is firstly to extract acoustic features from speech data, then to obtain an acoustic model through model training and statistics, which serves as the template for recognition, and to obtain an recognition result through decoding and processing combined with the language model.

The below figure is a flow of Speech recognition model.

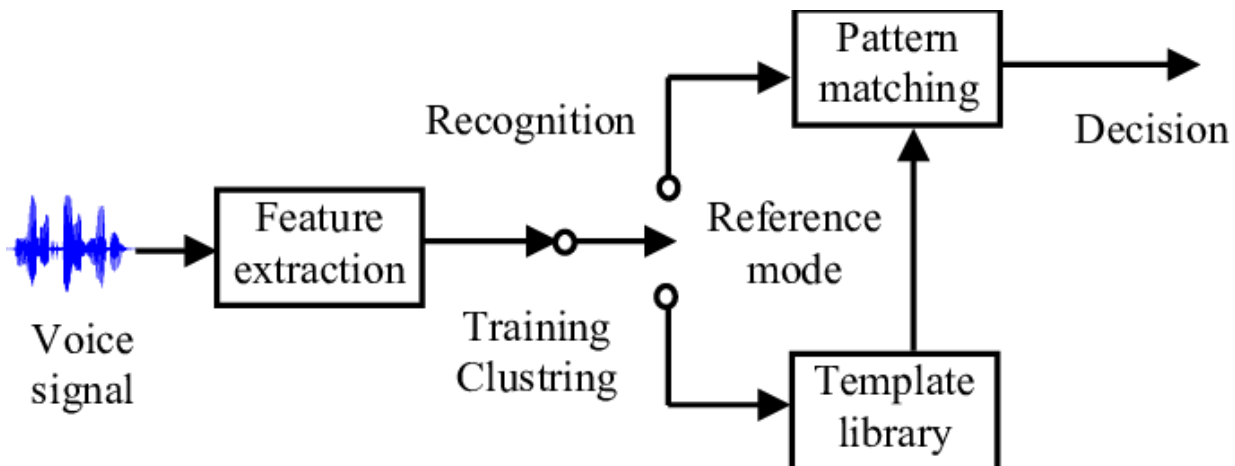


Fig. 1. Phoneme seeing model architecture

### Methodology and future work

The goal of our project is to create an interface that can be used to identify phonemes in sound signals. In order to achieve higher accuracy, DNN will be used to improve the accuracy of phoneme recognition. What needs to be considered is the preprocessing of speech signal in speech recognition system based on artificial neural network. The preprocessing based on speech segmentation preprocesses speech signals according to the phonetic symbols of speech to reduce the amount of data provided to neural network input, greatly improving the sensitivity of neural network input data. The numerical method can reduce the influence of noise on speech signal segmentation and identify the classification region more accurately.

At the same time, the speech signal should be divided into components, that is, phonemes should be selected to classify the speech information. The main focus in this research is to provide a GUI that can be used to easily predict phonemes for further use. For this purpose TensorFlow and Keras are to be used. The first step of the research is to define a model to be trained on a data set containing training phonemes and validated with a test data set. The next step is to link Keras's resulting graphics to the window application, which has a simple layout that shows the phonemes contained in the input speech fragment.

## **Conclusion**

The final result of the design process is to provide a system for analyzing the phonemes contained in speech fragments. The system is to work for desktops and to be designed in the Python programming language. Using DNN as the main type of neural network, relevant and reliable results can be obtained in most cases. TensorFlow is the main backbone framework that provides the DNN network, so it is included in the working module of the system.

## **References**

1. G. K. Berdibaeva, O. N. Bodin, V. V. Kozlov, D. I. Nefed'ev, K. A. Ozhikenov and Y. A. Pizhonkov, "Pre-processing voice signals for voice recognition systems," 2017 18th International Conference of Young Specialists on Micro/Nanotechnologies and Electron Devices (EDM), Erlagol, 2017, pp. 242-245.
2. Berik Akhmetov, Igor Tereykovsky, Aliya Doszhanova, Lyudmila Tereykovskaya (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. Vol 64, No 4 (2018), pp. 425-432. DOI: 10.24425/123541.
3. Nagamine, Tasha / Seltzer, Michael L. / Mesgarani, Nima (2015): "Exploring how deep neural networks form phonemic categories", In INTERSPEECH-2015, 1912-1916.
4. Goh, Y.H.; Raveendran, P. (2009). [IEEE 2009 International Conference for Technical Postgraduates (TECHPOS) - Kuala Lumpur, Malaysia (2009.12.14-2009.12.15)] 2009 International Conference for Technical

- Postgraduates (TECHPOS) - Phoneme segmentation of speech signal. , (), 1–3. doi:10.1109/techpos.2009.5412045
5. Deng, Li; Hinton, Geoffrey; Kingsbury, Brian (2013). [IEEE ICASSP 2013 - 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) - Vancouver, BC, Canada (2013.05.26-2013.05.31)] 2013 IEEE International Conference on Acoustics, Speech and Signal Processing - New types of deep neural network learning for speech recognition and related applications: an overview. doi:10.1109/icassp.2013.6639344

**D.Sc., Professor Tereikovskiy Ihor and Student Isik Oguzhan**

**National Technical University of Ukraine  
“Igor Sikorsky Kyiv Polytechnic Institute”**

## **SPAM RECOGNITION SYSTEM**

### **Анотація**

**Д.т.н, професор Терейковський І.А., магістрант Ишик Огузхан**  
***Інформаційна система розпізнавання спаму***

*У статті запропоновано архітектуру інформаційної системи інформаційної системи розпізнавання спаму. Завдяки впровадженню нейронних мереж та фільтрації можна підвищити ефективність запобігання та блокування спаму. Наведено результати дослідів, що підтверджують результати дослідження. Показано шляхи подальших досліджень для підвищення точності фільтрації.*

### **Introduction**

Spam is a major problem in the Internet area. Although there is not a unique description for spam (also called junk mail) and how it differs from legitimate mail (also called non-spam or genuine mail). Users of the network, in addition, are forced to spend time every day on processing useless for them advertising messages. Spam can be a carrier of Trojans and computer viruses - an attacker sends these programs to a certain range of addresses (belonging to users of a particular computer network) in order to gain access to computer systems, disable them or obtain sensitive data To reduce this time, they use anti-spam filters, which can erase an important message as spam.

### **Formulation of problem**

The main goal of the study is to provide an effective way to recognize spam and prevent it from reaching end users based on user patterns and email content and headings.

### **Research Objective**

The objective of this work is the analysis - development and research of methods of spam recognition based on the use of neural network theory.

## **Terminology**

**Spam.** It is an unauthorized mass anonymous sending of e-mails. Over the past year and a half, spam has become an acute problem, the relevance of which is constantly growing. There is no definite definition of spam that is internationally accepted, consensus has been achieved and the framework of which is fully defined. However, sending a large number of copies of the same message to people who have not requested to receive such a message is called spam. It is also called involuntary communication. It is estimated that about 15-30 million spam messages a day are sent in Ukraine today, and between 5 and 10 billion unsolicited advertisements a day worldwide. Moreover, according to experts, in the next two years the amount of spam will at least double.

There are the following categories of spam mailings.

**Anonymous mailing.** Mailing with a hidden or falsified return address. Bulk mailing: mass mailing is a business for spammers and a problem for users. A small mailing mistakenly made by a person who is not a professional spammer may be spam, but not spam.

**Unsolicited mailing.** They are e-mail messages sent to hundreds of thousands of e-mail accounts at the same time. These messages can be of commercial content, or they can be e-mail messages sent to propagate a political opinion or to create public opinion on a subject. Obviously, subscription mailings and conferences should not fall into the category of "spam" (although the condition of anonymity already largely guarantees this).

**Virus programmers.** Virus programmers create viruses to capture computers and turn them into spam machines. According to the data, the distribution systems used allow you to remember the status of each destination address and repeat delivery attempts through different intermediary machines until the moment of successful delivery. This feature dramatically reduces the effectiveness of blacklists (RBL systems).

**Mailing services.** These, in fact, are spammers. These are completely legal companies or just organized groups of qualified people. Such companies have software that sends a large number of emails per unit time (up to 100 emails per second).

**Amateur spammers.** There are two categories - novice spammers and unintentional spammers.

**Database collectors.** This type of player serves the needs of senders and collects email addresses for them, which they combine into an address database. They use a variety of software that collects addresses on the Internet, checks their functionality and creates a database of addresses.

### Description of the general approach

UML modeling language is used for this purpose. So in fig. 1 shows a diagram of the packages.

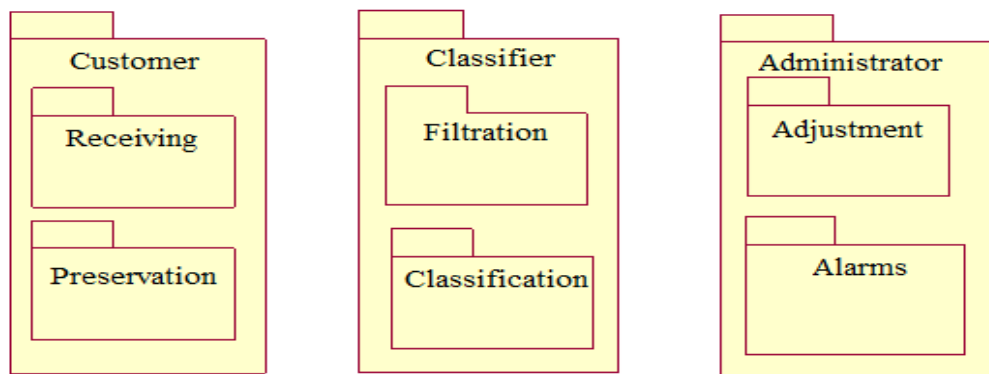


Fig. 1. Package diagram

The specified diagram shows the packages and the packages that make up the detection system:

"Client" - a package that is responsible for receiving the email and saving it for further analysis. This package includes sub-packages "Receipt" and "Disorder", the names of which correspond to their functionality.

"Classifier" - a package responsible for the classification of e-mail. It includes sub-packages "Filtration" and "Classification". The first subpackage is responsible for filtering the letter in order to obtain a set of formal features of the letter and a set of informative words that are part of it. The second sub-package is responsible for classifying the letter.

"Administrator" - a package that is responsible for the administration of the recognition system. This package consists of two subpackages "Settings" and "Alarm". The first subpackage is designed to give the detection system operator the ability to configure e-mail interception settings and recognition settings. The second package provides signaling of the recognition results.

### **Development of a basic neural network model**

NMs designed to process textual information are significantly different from the networks used in solving other practical problems. The reasons for the differences are both the structure of the input information and the nature of the word processing tasks. We will consider the principles of self-learning NM construction on the basis of the Lippmann-Hamming model. Note that other models can be used as a basis. However, consideration of the Lippmann-Hamming model is recommended because of its simplicity and the possibility of considering all the main aspects of the self-learning process.

The distance between the vectors is calculated according to the Hamming rule:

$$\left\{ \begin{array}{l} \rho_k = \sum_{i=1}^K \rho_i(x_i^k, \xi_i), \\ \forall(x_i^k = \xi_i) \Rightarrow \rho_i = 1, \\ \forall(x_i^k \neq \xi_i) \Rightarrow \rho_i = 0. \end{array} \right.$$

### **Architecture and main characteristics**

Developed anti-spam system is a powerful self-learning spam filter for Outlook and Outlook Express. It automatically detects and sorts spam, eliminating the need for users to manually view and weed out unwanted emails. It saves time and protects users from potentially dangerous content. Key features of Agnitum Spam Terrier include:

- Self-learning spam filter;
- Easy-to-use interface integrated into MS Outlook or MS Outlook Express;
- Simple installation/removal procedure;
- This is a free product.

Personal training filter. The core of the program is the Bayesian statistical method. The filter works independently of the message transmission protocol, evaluating emails already received by the email client. Not only the content of each letter is processed, but also other metadata, such as attachments and their size, delivery time, "garbage" in HTML-formatted messages, etc., which makes the algorithm even more efficient. The advantage of the Bayesian spam filter is that it can be trained individually for each user. The spam that a user receives is often specific to the network activity and interests of that user only. The spam probabilities of the words contained in the letters received by the user are unique to each user and may change over time due to corrective training in cases where the

filter incorrectly evaluates the letter. The Bayesian filter determines the spam probability of words and messages based on data individual to each user.

## **Conclusion**

In this thesis different spam detection and filtering methods are reviewed. Internet is widely used nowadays and the increase in the spam emails causes time and money loss with disrupted users. Spam emails cause a waste of time and money for the individuals having approximately a hundred spam each day. Spam filtering is an important topic for saving people from unsolicited commercial emails. Spam filtering started as keyword filtering and black-list white-list approach. In the recent more aggressively created spam mails requires faster, self adoptable high accuracy filters. In the process of solving the following results were obtained:

1. It is shown that a promising way to ensure sufficient quality of spam among e-mails is to improve the mathematical support of the information system. It is also shown that it is advisable to use a neural network model such as the Kohonen map for recognition.

2. The architecture of the recognition information system has been developed, which is adapted to the application of the proposed mathematical software based on the use of a neural network model such as the Kohonen map.

3. The mathematical and software of the information system based on use of the device of neural networks is developed.

4. Experimental studies have confirmed the prospects of using the developed mathematical software for spam recognition among e-mails.

5. It is recommended to implement the developed recognition system in general-purpose information systems in which there is a need to recognize spam with a moderate error.

## **References**

1. Anderson T. Statistical analysis of time series / Anderson T.; per. with. English I. G. Zhurbenko. - M.: MIR, 1976.-- 757 p.
2. Varenov D.A. Use of free software as part of the hardware and software complex of the Federal Center for Information and Educational Resources. / DA Varenov, A. Yu. Kuznetsov // Seventh Conference of Free Software Developers: Abstracts. - M., 2010.-- S. 12 - 14.



3. Information and Computer Security: Spyware and Protection Methods, Gürol Canbek, Şeref Sağıroğlu, December 2006, Graphic Publishing, ISBN 975-6355-26-3
4. Anderson T. Statistical analysis of time series / Anderson T.; per. with. English I. G. Zhurbenko. - M.: MIR, 1976 .-- 757 p. Autonomous Load Balancing, Part 1: Cisco Content Switching Module: Load Balancing Using Cisco CSM, SASP, and IBM Enterprise Workload Manager // IBM developerWorks: [online resource]. - Electron. text. Dan. - Access mode: <http://www.ibm.com/developerworks/ru/library/ac-ewlmload1/>. Ivens K. Computer networks / Ivens K.; per. with. English SPb. : Peter, 2006.304 p.

**К.т.н., старший викладач Хіцко Я.В., магістрантка Мірошник В.І.**

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

## **АЛГОРИТМІЧНО-ПРОГРАМНИЙ МЕТОД ГЕНЕРАЦІЇ ХМАР ТЕГІВ БІРЖОВИХ НОВИН**

### **Abstract**

**Yana Khitsko, senior lecturer, PhD, Vitalina Miroshnyk, student**  
***Algorithmic programming method for stock news tag cloud generation***

*This paper is about developing software for tag clouds of pricing news using Python programming language. Thousands of news are published every second and it is very important to weed out non-pricing information. Among the tools for tag cloud generation and text analysis, the authors singled out the WordCloud visualization technique, which helps with tag cloud generation and suite of libraries NLTK for natural language processing for English. The proposed method simplifies the search for valuable information for traders and investment companies, which reduces the time spent on the selection of key news, which are thousands on the market every day. The software will help to process large data sets that are almost impossible to process manually and will generate a tag cloud of pricing news for the selected ticker. The result is a combination of existing solutions with the proposed method for tag cloud generation and screening method.*

### **Вступ**

Світові ринки рухаються за рахунок очікувань наступних подій, які можуть мати дуже різні результати, а значить і різні реакції учасників ринку на них. Через це для згаданих учасників ринку дуже важливою є оцінка ймовірності кожного повороту подій, щоб продумати свої дії при настанні тої чи іншої події.

Протягом кожного місяця на ринку виходить величезна кількість економічної статистики, яка дає інвесторам ґрунтовну інформацію про такі аспекти ринку, як, наприклад, поточний стан економіки певної країни і яку динаміку розвитку можна очікувати від неї у майбутньому. Більшість економічної статистики, що виходить на ринку, несе в собі інформацію про рівень та динаміку інфляції, безробітності та інше.

Щосекунди виходять тисячі новин і дуже важливим є відсіювання шумової інформації. Біржеві трейдери готові купувати підписки на сервіси лише важливих новин задорого, аби бути в курсі найважливіших для них новин.

Серед безлічі інформації, яка щодня виходить на ринку, постає гостра проблема відсіювання тієї, що не буде впливати на теоретичні рішення трейдерів, фінансових аналітиків та інвесторів. Це, наприклад, автоматично створені новини компаній, що не несуть ніякого сенсу, але можуть суттєво забивати стрічку новин за день.

Найбільш явною зацікавленою стороною є біржеві трейдери, які аналізують інформацію для прийняття важливих рішень щодо ситуацій на ринку. Для них пошук ключових слів в новинах певної категорії є необхідним для виявлення, наприклад, схожих ситуацій на біржевому ринку. Треба також виявити тільки ключову інформацію, що буде корисною для надання аналітичних послуг інвесторам або для використання даної інформації у межах компанії для збору унікальних статистичних даних.

Слід зазначити, що генерація хмар тегів – не нова проблема. Під час пошуку відомих рішень даної задачі, було виявлено багато алгоритмів утворення хмар тегів та готових бібліотек для їх генерації. Однак просто генерація тегів не вирішує задачу знаходження важливих слів у біржових новинах. Конкурентною перевагою запропонованого методу відсіювання шумового контенту є власне відсутність відкритих існуючих рішень.

### **Постановка задачі**

Метою даного дослідження є розробити додаток, що генерує хмару тегів з новин, відсіювши шумові слова, використавши запропонований метод, що дозволить вдосконалити існуючі методи генерації хмар тегів.

Окремими завданнями дослідження є формування критеріїв слів, що є непридатними для аналізу, розроблення методу відсіювання шумових слів, тобто слів, що не мають впливу на біржу, а також генерування хмар тегів новин.

### **Термінологія**

*Хмара тегів* – візуальне подання списку слів, зазвичай з використанням кольору та розміру шрифту для представлення помітності або частоти зображуваних тегів. За допомогою хмар слів можна візуалізувати термінологію з певної теми у більш наочний спосіб. [1, 2]

*Стоп-слова* або *шумові слова* – слова, які відфільтровуються до або після обробки даних природною мовою і які не несуть смислового навантаження, тому їх користь та роль несуттєва. Поняття є дещо абстрактним, оскільки не існує єдиного універсального списку стоп-слів, які використовуються всіма інструментами обробки природної мови, ані узгоджених загальних правил їх визначення.

*POS (Part Of Speech) tagging* (з англ. розмічування частин мови) – це процес позначення слова в тексті як відповідного певній частині мови, на основі як його визначення, так і контексту. Спрощену форму цього зазвичай навчають дітей шкільного віку, ідентифікуючи слова як іменники, дієслова, прикметники, прислівники тощо. [3]

*Тікер (Stock symbol)* – унікальна серія літер, присвоєних цінному паперу у цілях торгівлі. Використовується як скорочений спосіб опису частини акцій компанії. Одна компанія може мати кілька тікерів. [4]

*Біржа* – це організований торговельний майданчик, на якому відбувається гуртова торгівля товарами або цінними паперами у вигляді стандартизованих біржових угод. На біржі укладаються угоди по біржових товарах, в результаті чого утворюється динаміка ціни тільки під впливом ринкового попиту та пропозиції, що дає змогу орієнтуватися учасникам ринку та прогнозувати хід торгів в майбутньому.

*Біграма* – це послідовність двох суміжних елементів із рядка лексем, які зазвичай є словами. Біграма – це  $n$ -грама для  $n = 2$ . Вони використовуються в більшості мовних моделей для розпізнавання мовлення.

### **Аналіз методу генерації хмар тегів з відсіюванням шумових слів**

Зазначимо, що дане дослідження фокусується на розробленому програмному забезпеченні для генерації хмар тегів, реалізованих мовою Python. Було розглянуто кілька різних інструментів для генерації хмар тегів – це Word It Out, Word Cloud Generation (англомовні сервіси для створення хмар слів, але мають обмежене фільтрування даних), а також кілька бібліотек для Python. Для полегшення роботи з обробкою тексту, було проаналізовано такі бібліотеки, як NLTK, Gensim, polyglot, TextBlob та інші. Перший пакет бібліотек є найпоширенішим і має найобширніший спектр методів обробки тексту.

Серед інструментів для генерації хмар тегів і аналізу текстів було виділено наступні інструменти: бібліотека WordCloud, що допомагає в генерації хмари тегів і пакет NLTK, набір бібліотек і програм для символічної та статистичної обробки природної англійської мови. Дані інструменти дозволяють спростити розробку користувацького інтерфейсу та полегшити роботу з початковими даними.

WordCloud має змогу генерувати хмару тегів обробляючи текст для пошуку найвживаніших слів, а також на основі масивів даних, сформованих зі слів і їх вагових коефіцієнтів, що визначаються ззовні. Задля можливості відсіювання шумових слів, було обрано другий варіант роботи даної бібліотеки. Спершу сформуємо список пар зі слів і вагових

коефіцієнтів за допомогою запропонованого нижче методу, використовуючи пакет NLTK.

Запропонований метод відсіювання шумових слів полягає у наступних кроках:

1. Відфільтрувати новини по потрібному проміжку дат.
2. Провести аналіз часу новини – якщо вона вийшла після закриття біржі, дата її впливу на ринок вважається наступним робочим днем.
3. Відсіяти новини, які є автоматично згенерованими звітами, що не несуть в собі смислового навантаження.
4. З кожної новини прибрати стоп-слова, використавши стандартний для NLTK англословний список стоп-слів.
5. Розібрати кожну новину на частини мови використавши POS tagging і прибрати непотрібні.
6. Окремим випадком при аналізі частин мови дослідити тікери, оскільки вони є невід'ємною частиною новин, що стосуються бірж.
7. Визначити в новинах біграми з перевіркою на те, чи не стало словосполучення біграмою після видалення шумових та інших слів.
8. Сформуванню словника ваг тегів новин:
  - 8.1. Підрахувати кількість згадок кожного слова і біграм.
  - 8.2. Відминусувати від обох слів зі словника, що належать біграмам, кількість згадок відповідних біграм.
  - 8.3. Відминусувати перетин кількості слів новин, що вийшли в один день і є однакові за посилком.
  - 8.4. Підвищити ваги слів для новин у ті дні, у які були рухи ціни за головним тикером аномальними. [5]

Запропонований метод спрощує пошук цінної інформації, що знижує затрати часу на виділення ключових новин, які щодня тисячами виходять на ринку. Так, на рис. 1 бачимо, що при формуванні хмари тегів без жодної фільтрації, матимемо непридатну для аналізу хмару новин, у той час як на рис. 2 продемонстровано збалансований набір слів, по яким можна робити певні висновки. Таким чином, програмне забезпечення допоможе обробляти великі масиви новинних даних, які майже неможливо обробити вручну і згенерує хмару тегів слів, що мають вплив на біржу, для обраного напрямку дослідження, чим зніме з плечей трейдерів, інвесторів, фінансових аналітиків багатьох компаній щоденну монотонну роботу.



## Висновки

У даній статті було запропоновано метод відсіювання шумових слів для генерації хмар тегів на основі візуального представлення WordCloud. Даний підхід дозволяє знизити затрати часу зацікавлених осіб на опрацювання великих масивів новин, які виходять на різних новинних ресурсах щодня.

Такий метод допоможе знайти маловідомі залежності між учасниками біржового ринку, оскільки після обробки великих обсягів новин лишається візуально сприйнятна і придатна для швидкого аналізу хмара тегів оброблених новин.

Отже, у результаті отримано комбінацію існуючих рішень з запропонованим методом для генерації хмар тегів і методу відсіювання шумових слів. Для подальшої роботи у даному напрямку доречно буде провести порівняльний аналіз ефективності запропонованого методу при різних вхідних даних і заміні фільтрів методу, оскільки з плином часу змінюються формати новин, поняття про аномальність модулів рухів обраних для аналізу компаній тощо.

## Література

1. Owen Kaser and Daniel Lemire. Tag-Cloud Drawing: Algorithms for Cloud Visualization // CoRR. — 2007.
2. Helic, D., Trattner, C., Strohmaier, M. and Andrews, K. Are Tag Clouds Useful for Navigation? A Network-Theoretic Analysis // Journal of Social Computing and Cyber-Physical Systems. — 2011. — P. 33-55.
3. Charniak, Eugene. «Статистичні методи розпізнавання природних мов» AI Magazine 18(4) – 1997. – P. 33–44.
4. Eckett, Stephen, ed. The UK Stock Market Almanac 2005. Petersfield: Harriman House – 2004. – ISBN 1-897597-46-0.
5. Sharpe, W. F. "Mutual Fund Performance". Journal of Business. 39 (S1) – 1966. – P. 119–138.

**К.т.н., старший викладач Хіцко Я.В., студент Нечипоренко В.О..**

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

## **МЕТОД ДЛЯ АНАЛІЗУ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ ВІДГУКІВ КОРИСТУВАЧІВ**

### **Abstract**

**Iana V. Khitsko, senior lecturer, PhD; Nechyporenko Vladyslav, student**  
***Method for analysis of software quality based on user reviews***

*This document addresses the task of creating a method and software for qualitative analysis of software quality based on user feedback. There are many methods and ways to analyze the software quality, but none of them use user feedback. Because users are valuable software consumers, their opinion directly affects the software quality and the rating of the company that developed it. A method that uses user feedback to display results is proposed. A comparison of available methods and services for software quality analysis is performed. Ways of further research are also offered.*

### **Вступ**

Аналіз якості програмного забезпечення – це дуже важливий процес, оскільки на сьогодні майже кожна компанія має своє програмне забезпечення і вкладає багато коштів у його розвиток. До того ж, від якості програмного забезпечення залежить загальне ставлення користувачів і клієнтів компанії, що впливає на рейтинг компанії і її фінансові показники. Тому компанії дуже важливо знати, наскільки якісним є її програмне забезпечення. Існує багато методів і сервісів для аналізу якості програмного забезпечення. Проте, вони здебільшого використовують програмний код для аналізу. Це не є досить ефективним, оскільки набагато важнішим є думка користувача щодо розробленого програмного забезпечення, оскільки він є його кінцевим споживачем. До того ж, аналізуючи відгуки користувачів, можна не тільки визначити, наскільки якісним є програмне забезпечення, але і виявити критичні помилки під час роботи з ним для їх подальшого виправлення. У цій статті пропонується розглянути новий метод та програмне рішення для аналізу якості програмного забезпечення за допомогою відгуків користувачів.



## **Постановка завдання**

Завдання полягає в створенні методу для аналізу якості програмного забезпечення на основі відгуків користувачів та програмного забезпечення до нього, яке було б доступно кожному й мало всі необхідні інструменти для відображення результатів роботи методу.

## **Термінологія**

*Якість програмного забезпечення* – набір властивостей продукту (сервісу або програм), що характеризують його ступінь відповідності до початкових вимог [1].

*Аналіз якості програмного забезпечення* – це систематизований збір і аналіз інформації для підвищення якості програмного забезпечення і збільшення попиту на продукцію чи послугу в рамках обмеженого бюджету [1].

*Відгук користувача* – це письмова чи усна змістовна думка користувача про програмне забезпечення з тією чи іншою оцінкою з описом тих чи інших проблем програмного забезпечення [2].

*Метрика програмного забезпечення* – це міра, що дає змогу отримати числове значення деяких властивостей програмного забезпечення та його специфікацій [3].

## **Опис методу для аналізу якості програмного забезпечення на основі відгуків користувачів**

Метод для оцінювання якості програмного забезпечення на основі відгуків користувачів складається з чотирьох етапів: збір відгуків, поділ відгуків на категорії для підрахунку значень тих чи інших метрик, підрахунок значень метрик і параметрів за відповідними формулами, формування результатів.

На першому етапі, необхідно отримати відгуки про програмне забезпечення в електронному вигляді. Збір відгуків відбувається за допомогою відповідної системи, на якій розміщено програмне забезпечення, що аналізується, наприклад, App Store, Google Play, Steam тощо.

На другому етапі, необхідно проаналізувати текст у відгуках і поділити їх згідно з метриками, що були обрані в моделі метрик методу. Поділ відбувається згідно з ключовими словами.

Загальна методологія для створення методу включає такі етапи: визначення цілей оцінки, розробка моделі якості програмного забезпечення, генерація моделі метрик, пошук базових метрик, визначення похідних метрик, оцінка метрик, формалізація метрик і визначення граничних значень метрик [4].

Після етапу визначення похідних метрик маємо фінальну модель метрик, що зображена на рис. 1. Саме завдяки похідним метрикам, що зображені на найнижчому рівні моделі, а саме: операційна система, пристрій, локальні функції, функції сторонніх сервісів, орієнтація дисплею, точність, зручність, країна користувача, обробка даних, мова тексту, мова озвучення, спеціальні можливості, буде відбуватися поділ відгуків на другому етапі.

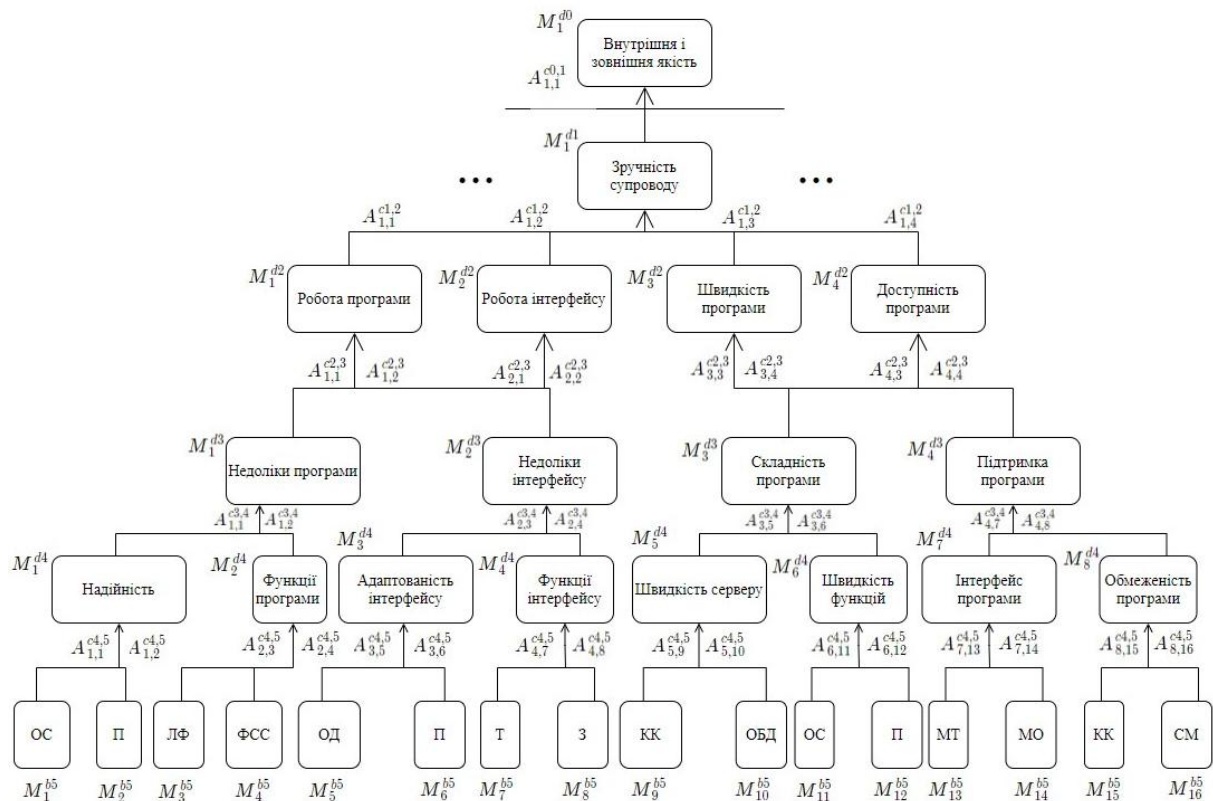


Рис. 1. Модель метрик

На третьому етапі, будуть підраховуватися значення метрик і параметрів за відповідними формулами. Загальний вигляд формул для підрахунку похідних метрик:

$$m_j^{bi} = \sum_{p=0}^n f(m_j^{bi}) * x(r_p, m_j^{bi}) * c(r_p) * z(r_p) * q(r_p),$$

де  $r$  – відгук, що оцінюється,  $x(r_p, m_j^{bi})$  - функція, що визначає чи належить відгук до метрики  $m_j^{bi}$ ,  $f(m_j^{bi})$  – функція, що визначає коефіцієнт важливості метрики  $m_j^{bi}$ , що залежить від кількості відгуків згідно з заданою метрикою і її оцінки, що була визначена на етапі оцінки метрик,  $c$  - коефіцієнт, що залежить від довжини оцінюваного відгука,  $z$  – коефіцієнт, що залежить від рейтингу відгуку і  $q$  – коефіцієнт, що залежить

від якості відгуку. Значення коефіцієнтів належить відрізку  $[0;1]$ , значення функції  $f(m_j^{bi}) - [0;q]$ , де  $q$  – оцінка метрики від 1 до 4. Формули метрик і параметрів, що розташовуються вище – це сума метрик чи параметрів знизу помножена на оцінку метрики чи параметра, що були визначені на етапі оцінки метрик. Формули параметрів, що розташовуються на рівні  $M_j^{d2}$  використовують відгуки, що належать до них і мають такий вигляд:  $m_j^{d2} = qp/n(m_j^{d2})$ , де  $q$  – оцінка параметра,  $p$  – сума параметрів, що розташовуються на рівень нижче і  $n(m_j^{d2})$  – сума значень відгуків, що належать до цього параметру. Відповідно загальна оцінка  $M_I^{d0} = M_I^{d1}$  й дорівнює сумі параметрів, що розташовуються на рівні  $d^2$  поділену на загальну кількість відгуків.

На останньому етапі, відображаються результати роботи методу, а саме: загальне значення і його аналіз – чим воно менше, тим краще, оскільки коефіцієнти дорівнюють 0 у позитивному випадку і 1 - у негативному випадку.

## Висновки

Описаний метод та програмне забезпечення до нього розв'язують проблему спілкування користувачів із розробниками програмного забезпечення і надають можливість отримати зворотній зв'язок від користувача щодо розробленого програмного забезпечення. До того ж, розроблений метод допомагає виявляти недоліки програмного забезпечення і його слабкі місця.

Якщо порівнювати цей метод із наявними, то можна однозначно виділити його переваги, а саме: добре визначені метрики й параметри, адаптованість методу, і простота реалізації і використання. До того ж, цей метод аналізує програмне забезпечення з допомогою відгуків користувачів, на відміну від багатьох методів, що використовують програмний код для аналізу якості програмного забезпечення [5].

У подальшому було б доцільно попрацювати над аналізом тексту відгуків із метою його покращення й автоматизувати процес вибору ключових слів.

## Література

1. Software quality [Електронний ресурс]. — Режим доступу до ресурсу: [https://en.wikipedia.org/wiki/Software\\_quality](https://en.wikipedia.org/wiki/Software_quality).
2. Review [Електронний ресурс]. — Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/Review>.

3. Software metric [Электронный ресурс]. — Режим доступа до ресурсу: [https://en.wikipedia.org/wiki/Software\\_metric](https://en.wikipedia.org/wiki/Software_metric).
4. Методика оценки качества программных средств [Электронный ресурс]. — Режим доступа до ресурсу: <https://cyberleninka.ru/article/n/metodika-otsenki-kachestva-programmnyh-sredstv/viewer>.
5. Методы оценки и измерения характеристик информационных систем [Текст] / И.Ю. Коцюба, А.В. Чунаев, А.Н. Шиков. — М. : ИВЦ “ НИУ ИТМО”, 2015. — 262 с.

**Асистент Струцинський О.А., магістрант Щербина В.О.**

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

## **АЛГОРИТМ ЗБЕРІГАННЯ ТА АНАЛІЗУ КОРИСТУВАЦЬКИХ МЕТРИК МОНІТОРИНГОВИМИ СИСТЕМАМИ**

### **Abstract**

**Oleksii A. Strutsynskyi, assistant; Vadym Shcherbyna, student**

*Algorithm of storage and analysis of user metrics by monitoring systems*

*This paper demonstrates the developed algorithm of storing and analyzing data by monitoring systems. Usually monitoring systems use database. This leads to the creation of the algorithm which will optimize the saving data process and will provide statistics calculation. This algorithm describes a developed data structure to store system data and calculate statistics.*

### **Вступ**

Зростає кількість та різноманітність програмного забезпечення, що в свою чергу збільшує кількість моніторингових систем програмного забезпечення.

Моніторингова система програмного забезпечення [1] виконує збір, аналіз та збереження визначених показників моніторингу. Найбільш ресурсоємними процесами є аналіз та збереження даних тому, що більшість моніторингових систем використовують базу даних. Такий спосіб є неефективним, тому що дані постійно потрібно записувати та отримувати з бази даних. В свою чергу, значні часові ресурси витрачаються на аналіз, який використовується для прогнозування майбутніх показників експлуатації програмного забезпечення.

У даній статі пропонується розробити алгоритм зберігання та аналізу показників моніторингу, який дозволяє обробляти велику кількість користувацьких метрик. Даний підхід полягає у використанні модифікованої структури зберігання даних в пам'яті та модифікацію обробки статистичних показників для подальшого прогнозування майбутніх показників експлуатації програмного забезпечення.

### **Постановка задачі**

Задача полягає в розробці алгоритму для автоматизованого моніторингу програмного забезпечення з можливістю зберігання та обробки користувацьких метрик таким способом, щоб швидкість обробки даних була доволі високою, дані оброблялись в режимі реального часу та з можливістю подальшого використанні для прогнозування майбутніх показників експлуатації програмного забезпечення.

Для досягнення поставленої задачі, необхідно: проаналізувати існуючі алгоритми зберігання та обробки даних; проаналізувати вхідні дані; розробити алгоритм, який найкраще буде підходити для зберігання та обробки користувацьких метрик в системах моніторингу програмного забезпечення.

### **Термінологія**

*Моніторинг* – це процес безперервного або періодичного спостереження за станом якісних і кількісних показників, які характеризують певну діяльність. [2]

*Метрика* – це не оброблені дані про стан будь-якого показника, які можна відслідковувати, обробляти і зберігати в системі.

*Критерій* – ознака, яка використовується для класифікації або оцінки якогось явища, або процесу.

*Websocket* – це мережевий протокол для передачі даних.

*Структура даних* – це спосіб організації та представлення даних, який використовується для побудови алгоритмів.

### **Опис способу**

Розроблений алгоритм для аналізу та зберігання моніторингових даних буде застосовуватись для їх обробки в реальному часі. Більшість існуючих моніторингових систем є автоматичними, тобто вони не потребують якогось втручання людини. Проте існує потреба в моніторингу метрик, які необхідні користувачу, але не передбачені автоматичним моніторингом, для покращення якості роботи програмного забезпечення. В зв'язку з цим було прийнято рішення створити автоматизовану моніторингову систему, яка надасть змогу користувачам задавати проміжок часу протягом якого метрики будуть зберігатись в системі, а також встановлювати критерії для них. Для реалізації такого алгоритму необхідно:

- сконфігурувати потрібні метрики,
- задати часовий інтервал,
- налаштувати критерії метрик.

На основі проведеного аналізу було з'ясовано, що для оптимізації і

економії часу при подальшому прогнозуванні необхідно створити структуру даних, яка буде обраховувати статистичні показники та визначати стаціонарність ряду при додаванні нової метрики.

Наведемо основні етапи запропонованого алгоритму:

1. Отримання даних з клієнтської частини для подальшої обробки та прогнозування. Клієнтська частина надсилає необхідні метрики за допомогою з'єднання по `websocket`, яке надає змогу безперервно надсилати дані з клієнтської частини на серверну. Клієнтська частина має змогу відсилати метрики з будь-якою періодичністю, а серверна частина буде створювати чергу і обробляти їх. Такий підхід дозволяє зберігати всі дані і запобігає їх втраті.
2. Зберігання користувацьких метрик на серверній частині. Для збільшення продуктивності користувач може задавати час впродовж якого будуть зберігатись метрики. Для такого підходу розроблено структуру даних, яка буде працювати за принципом FIFO (first in first out) та зберігати дані для певних проміжків часу і потім видаляти їх, коли вони вже не потрібні. [3]
3. Обробка користувацьких метрик та визначення комірки в яку вони будуть додані.
4. Обробка та аналіз статистичних показників [4].
5. Видалення метрики, коли пройшов час її зберігання в системі. Коли настає даний етап всі значення видаляються з комірки, яка відповідає за певний проміжок часу і статистичні показники перераховуються за формулами з пункту (3)
6. Резервне збереження даних в системі. Через певні проміжки часу дані, які зберігаються в нашій структурі, записуються у файл для відновлення у випадку збою, або перезапуску системи і подальшого використання методів прогнозування.

Розглянемо більш детально обробку та аналіз статистичних показників. Коли дані потрапляють в систему і зберігаються в розробленій структурі даних, вони потрапляють в певну її комірку. Час впродовж якого в певній комірці зберігаються дані розраховується за формулою:

$$n = \frac{N}{T} \quad (1)$$

де  $N$  – розмір структури даних, який задається клієнтською частиною,  $T$  – проміжок часу впродовж якого дані для певної метрики будуть зберігатись в системі.

При додаванні значень для певної метрики у відповідну комірку, статистичні показники будуть перераховуватись.

Знаходження дисперсії за формулою (2)

$$D = \frac{\sum_{i=1}^n x_i^2}{n} - \left( \frac{\sum_{i=1}^n x_i}{n} \right)^2, \quad (2)$$

де  $n$  – кількість елементів в часовому ряді.

Знаходження середнього квадратичного відхилення  $\sigma$  за формулою

$$\sigma = \sqrt{D}, \quad (3)$$

Визначення стаціонарності ряду. При додаванні нового значення метрики, порівнюється значення дисперсії для попереднього часового інтервалу з новим значенням і визначається стаціонарність ряду. Якщо значення дисперсії є не зміним, то ми можемо стверджувати, що часовий ряд є стаціонарним.

## Висновки

Запропонований алгоритм збору та аналізу користувацьких метрик оптимізує процес зберігання отриманих користувацьких даних і доцільний для подальшого прогнозування. Даний алгоритм дає можливість зменшити час на збереження та обробку даних. Це дозволяє підготувати і обробити дані для подальшого використання.

У подальшому було б доцільно оптимізувати запропонований алгоритм збереження на аналізу даних моніторингової системи для оптимізації процесу резервного збереження даних та їх відновлення.

## Література

1. *Boulon J. Chukwa: A large-scale monitoring system* / J. Boulon, A. Konwinski, R. Qi, A. Ribkin, E. Yang, M. Yang // The first workshop on Cloud Computing and its Applications (CCA'08), Chicago USA, October 22-23, 2008. – С 1-5
2. *Lee Y. A Hadoop-based packet trace processing tool* / Y. Lee, W. Kang, Y. Lee // Traffic Monitoring and Analysis LNCS 6613, Springer, 2011 – С 51-63
3. *Rattenbury T. Principles of Data Wrangling: Practical Techniques for Data Preparation* / T. Rattenbury, J. M. Hellerstein, J. Heer, S. Kandel, C. Carreras // 2017. – С. 50-75
4. *Smithson M. Statistics with Confidence: Confidence Intervals and Statistical Guidelines* / M. Smithson // 2002. – С 25-40



**D.Sc., Professor Tereikovskiy Ihor and Student Abdulwahid Ibrahim**

**National Technical University of Ukraine  
“Igor Sikorsky Kyiv Polytechnic Institute”**

## **PREDICTING WEB SERVER LOAD**

### **Анотація**

**Д.т.н, професор Терейковський І.А., магістрант Ібрахім Абдулвахід  
Прогнозування навантаження на веб-сервер**

*У статті запропоновано архітектуру інформаційної системи прогнозування навантаження на веб-сервер на основі апарату марківської апроксимації. Запропоновано підвищити ефективність прогнозування шляхом застосування модифікованої марківської моделі. Показано, що шляхи подальших досліджень пов'язані з підвищенням точності прогнозування.*

### **Introduction**

In modern conditions of intensive development of information and communication technologies, the task of ensuring the stable operation of a separate Web server, as an element of the structure of the COP is relevant and important. This task is especially important for small and medium-sized businesses that have limited resources and are in the development stage. In the event of a change in the load on a single Web server, a situation often arises that leads to a decrease or loss of its performance. In such organizations, it takes from several hours to several days to restore its stable operation, which leads to significant losses.

The change in load can occur in the case of the impact on the server of external difficult-to-formalize factors, such as increasing the volume of tasks, increasing the number of customers, transformational changes in the scale of the enterprise. All this can lead to the loss of valuable information, unforeseen and significant material costs in an active transition to the Internet economy. To provide early warning of possible future malfunctions of a particular server and take into account these risks when planning the development of enterprise resources, it is advisable to use methods of analyzing the server, focused on both current and projected estimates of changes in physical (numerical) server parameters. expert assessments of enterprise development trends.

To solve the problem of ensuring the stable operation of the Web server, systems are used to monitor and control the performance of servers on individual

parameters. Examples of such systems are PWS IT Monitoring and NMS Aurora. However, the functions of such systems do not provide a solution to the problem of forecasting, and do not allow to identify dangerous trends in the state of the server as a whole object.

Quite often the approach to forecasting the state of the server is used by visual analysis of graphs of time series of numerical parameters, but the results of the forecast largely depend on the qualifications and subjective opinion of the network administrator.

Studies of domestic and foreign scientists indicate that a promising way to solve the problem of forecasting the load on the web server is to apply the theory of Markov chains

### **Formulation of the problem**

The main goal of the study is to provide an effective forecast of the load on the web server.

### **Research Objective**

The object of the research is the processes of forecasting the load on the web server.

### **Terminology**

A web server is a server that accepts HTTP requests from clients, usually web browsers, and issues HTTP responses to them, usually along with an HTML page, image, file, media stream, or other data.

Markov chain (Markov model) is a sequence of random events with a finite or countable number of outcomes, where the probability of each event occurring depends only on the state achieved in the previous event. It is characterized by the property that, speaking loosely, with a fixed present, the future is independent of the past.

### **Description of the general approach**

A random process of changing the states of a certain system is called a Markov chain (Markov process) if, for an arbitrary moment in time, the probability of the system's transition to a new state  $p_{i,j}$  depends only on the current state of the system  $S_i$  and does not depend on how the system got into the current state:  $p_{i,j} = f(S_i)$ .

To define a Markov chain, you must specify the following parameters:

- the probabilities of system transitions between states in one step of the process ( $p_{i,j}$ );

- the number (N) and signs of states that the experimental system can get into;
- the probabilities of the stay of the system by states at the zero moment of time ( $P_i(0)$ ).

An example of representing a Markov chain as a graph is shown in Fig. 1.

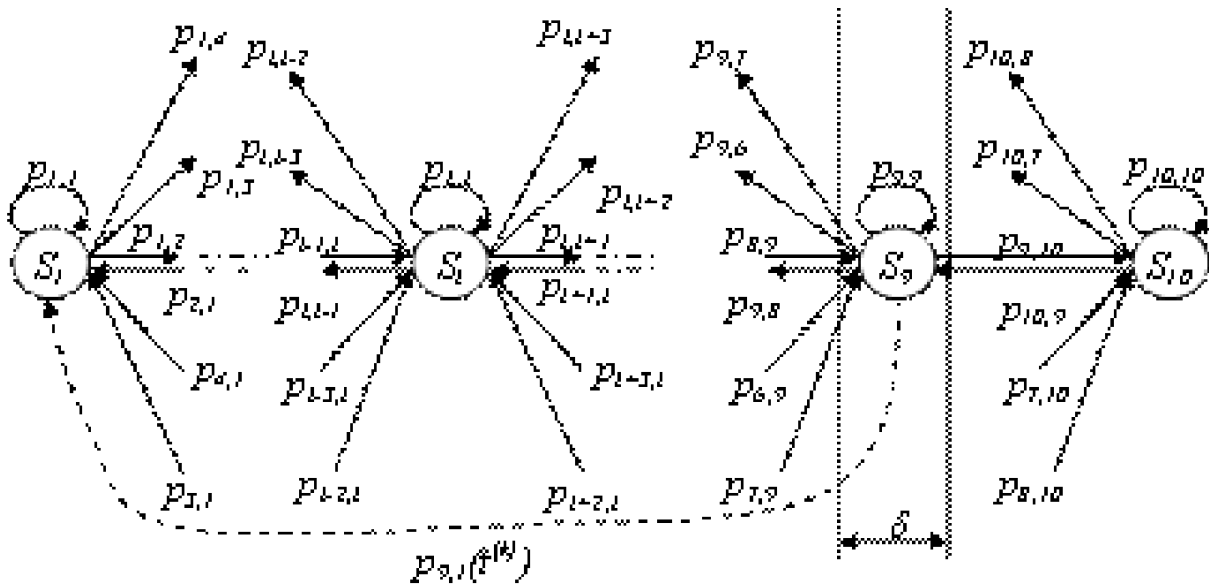


Fig. 1 An example of representing a Markov chain as a graph

### Load prediction system architecture

As a result of the research carried out, the architecture of the forecasting system has been developed. UML modeling language was used. The developed component and use case diagrams are shown in Fig. 2-3.

In Fig. 2 shows a diagram of the components of the forecasting system, consisting of three modules "Sniffer", "Statist" and "MarkPr". These modules define the interface for obtaining load parameters, their primary statistical analysis and forecasting using a Markov template.

The diagram of precedents is shown in Fig. 3. When constructing this diagram, it is taken into account that it consists of 4 actors: "Sensor", "Analyzer", "Administrator" and "System Operator". The actors "Sensor" and "Analyzer" are used to obtain load parameters, its analysis and forecasting. The actors "System Operator" and "Administrator" are actors who interact with the system.

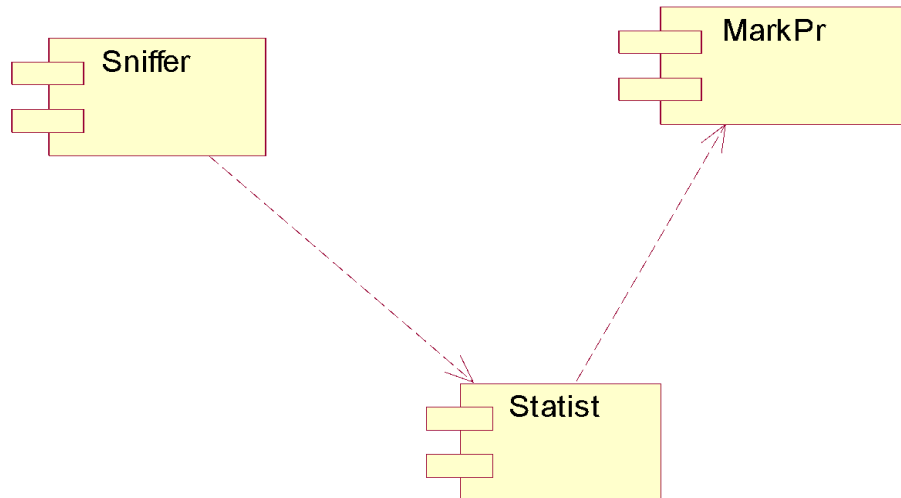


Fig. 2. Component diagram

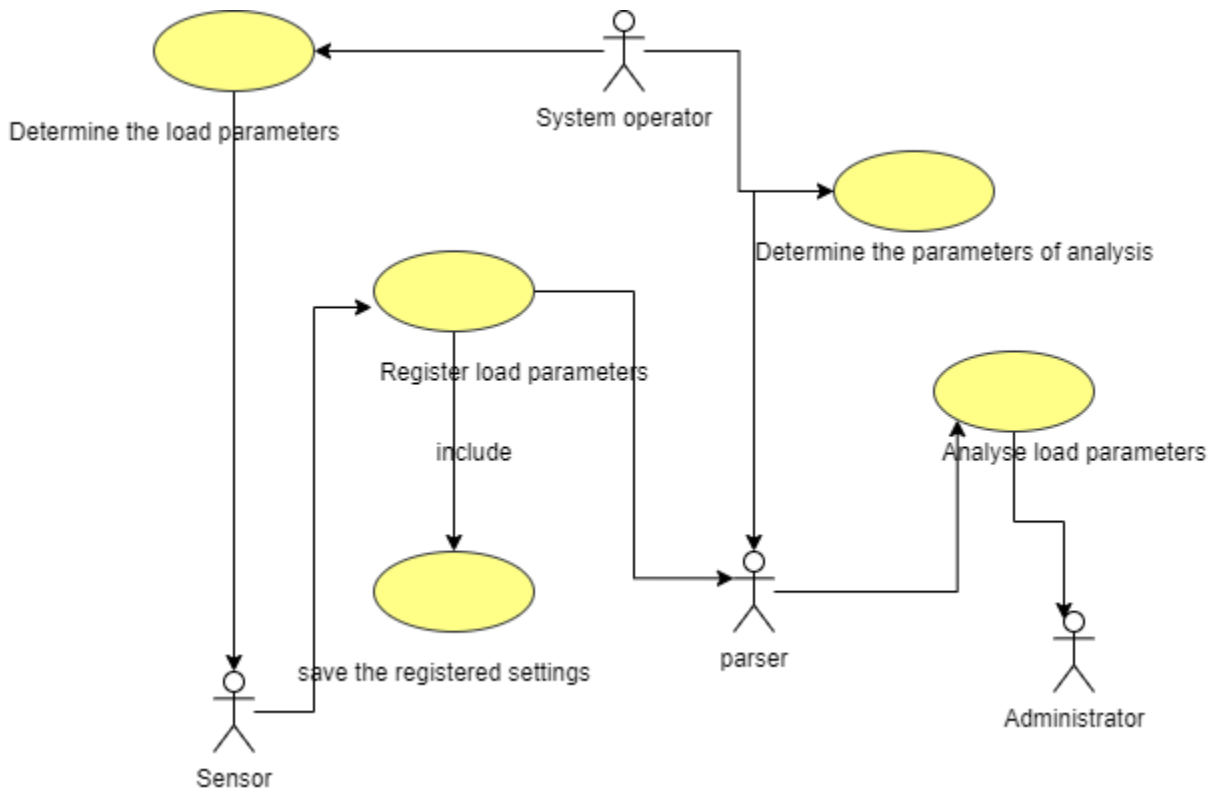


Fig. 3. Diagram of precedents

## Conclusion

In this paper the scientific and applied problem of development of effective models, methods and means of forecasting of loading on the web server on the basis of the theory of Markov processes was considered. In the process of solving the following results were obtained:

It is shown that a promising way to ensure sufficient quality of forecasting the load on the web server is to improve the appropriate mathematical apparatus. It is also shown that the task of forecasting the load on the web server can be considered as a task of predicting the parameters of the dynamics of the technical system, which allows to use for its solution the tested apparatus of Markov approximation of the dynamics of parameters.

The architecture of the forecasting system is developed, which is adapted to the application of the proposed mathematical software based on the apparatus of Markov approximation of the dynamics of parameters.

## References

1. Anderson T. Statistical analysis of time series / Anderson T.; per. with. English I. G. Zhurbenko. - M.: MIR, 1976 .-- 757 p. Autonomous load balancing, Part 1: Cisco Content Switching Module: Load balancing using Cisco CSM, SASP, and IBM Enterprise Workload Manager // IBM developerWorks: [online resource]. - Electron. text. Dan. - Access mode: <http://www.ibm.com/developerworks/ru/library/ac-ewlmlload1/>.
2. Ivens K. Computer networks / Ivens K.; per. with. English SPb. : Peter, 2006.  $\frac{3}{4}$  304 p.
3. Baranov PA Detection of anomalies on the basis of the analysis of homogeneity of parameters of computer systems: dissertation ... Candidate of technical sciences: 05.13.19 St. Petersburg, 2007 155 p. (In Russian).
4. Varenov D.A. Use of free software as part of the hardware and software complex of the Federal Center for Information and Educational Resources. / DA Varenov, A. Yu. Kuznetsov // Seventh Conference of Free Software Developers: Abstracts. - M., 2010 .-- S. 12 - 14.

УДК 539.3

К.ф.-м.н., ст. викладач Бай Ю.П., д.т.н., професор Ориняк І.В.,  
студент Городнюк С.О.

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

## ЗАСТОСУВАННЯ КОМБІНАЦІЙ ТОЧНИХ АНАЛІТИЧНИХ РІШЕНЬ ТА МЕТОДУ КОЛОКАЦІЙ ДЛЯ ВИЗНАЧЕННЯ ВЛАСНИХ ЧАСТОТ ПЛАСТИН

### Abstract

Yulia Bai, sen. lect., PhD (Phys. & Math.); Igor Orynyak, prof., Dr. Sci. (Engin.);  
Serhii Horodniuk, student

*Application of combinations of exact analytical solutions and collocation method for  
determination of natural frequencies of plates*

*The idea of paper is in employment of combinations of exact analytical solutions in the collocation method to determine the natural frequencies of rectangular plates. The existing approaches to solving the problem of plate deformation are investigated and discussed. Different types of boundary conditions that affect the complexity of the solution are considered. The accuracy of the proposed method with minimal number of terms is compared for case of simply supported rectangular plate, for which the analytical solution is available.*

### Вступ

Проблема вивчення власних коливань прямокутних пластин має більш, ніж 200-річну історію, але і в наш час привертає увагу багатьох досліджень з використанням широкого спектру методів [1-4]. Точні розв'язки даної задачі доступні лише для певних граничних умов і конфігурацій області, тому наближені розв'язки мають велике значення у тих випадках, коли аналітичні методи дають збій або стають занадто громіздкими.

У даній статті досліджується задача про вільні коливання квадратної пластини та пропонується підхід до її розв'язання за допомогою методу колокації.

### Постановка задачі

Розглядається квадратна ізотропна пластина, що займає область  $D: \{(x, y), 0 \leq x, y \leq a\}$ . Задача динамічного деформування даної пластини

зводиться до розв'язання диференціального рівняння 4-го порядку в часткових похідних:

$$\frac{\partial^4 w}{\partial x^4} + 2 \frac{\partial^4 w}{\partial y^2 \partial x^2} + \frac{\partial^4 w}{\partial y^4} = \omega^2 w, \quad (1)$$

де  $w(x, y)$  - невідома функція пружних переміщень,  $\omega$  - умовна частота, що підлягає визначенню. В даній роботі розглядаються граничні умови шарнірного обпирання пластини на всіх чотирьох гранях:

$$w(0, y) = 0, \quad M_x(0, y) = 0 \Rightarrow \frac{\partial^2 w}{\partial x^2}(0, y) = 0, \quad (2a)$$

$$w(a, y) = 0, \quad M_x(a, y) = 0 \Rightarrow \frac{\partial^2 w}{\partial x^2}(a, y) = 0, \quad (2б)$$

$$w(x, 0) = 0, \quad M_y(x, 0) = 0 \Rightarrow \frac{\partial^2 w}{\partial y^2}(x, 0) = 0, \quad (2в)$$

$$w(x, a) = 0, \quad M_y(x, a) = 0 \Rightarrow \frac{\partial^2 w}{\partial y^2}(x, a) = 0. \quad (2г)$$

Тут  $M_x, M_y$  - згинаючі моменти в пластині, що визначаються на основі переміщень  $w(x, y)$  за формулами:

$$M_x = - \left( \frac{\partial^2 w}{\partial x^2} + \nu \frac{\partial^2 w}{\partial y^2} \right), \quad M_y = - \left( \frac{\partial^2 w}{\partial y^2} + \nu \frac{\partial^2 w}{\partial x^2} \right), \quad (3)$$

де  $\nu$  - коефіцієнт Пуассона матеріалу пластини.

Задача полягає у знаходженні множини власних частот  $\omega$  та дослідженні форм пружних переміщень і згинаючих моментів для кожної частоти. В роботі Лейсса [1] для прямокутної пластини  $a \times b$ , шарнірно опертої по чотирьом граням, невідома функція пружних переміщень  $w(x, y)$  обирається у вигляді:

$$w_{n,m}(x, y) = \sin \frac{n\pi x}{a} \cdot \sin \frac{m\pi y}{b}, \quad n, m = 1, 2, 3, \dots \quad (4a)$$

В цьому випадку значення відповідних власних частот визначаються за формулою:

$$\omega = \pi^2 \left( (n/a)^2 + (m/b)^2 \right). \quad (4б)$$

В даній роботі пропонується інший підхід до розв'язання задачі, що базується на застосуванні точних аналітичних розв'язків та методу колокації.

### Термінологія

*Вільні коливання* — це коливання, які відбуваються в механічній системі під дією внутрішніх сил системи після короткочасного впливу зовнішньої сили.

*Метод колокації* — це метод чисельного розв'язання звичайних диференціальних рівнянь, диференціальних рівнянь в часткових похідними

та інтегральних рівнянь. Ідея методу полягає в тому, що необхідно вибрати простір можливих розв'язків (зазвичай це многочлени до деякого степеня) та набір точок в області (вузли колокації) та за допомогою обраних функцій задовольнити диференціальне рівняння або граничні умови в точках колокації.

*Сітка вузлів колокації* — система точок колокації в області, що досліджується, яка може бути як рівномірною, так і нерівномірною.

*Дисперсійне рівняння* — це рівняння для визначення власних частот  $\omega$ , яке виражає умову існування нетривіальних розв'язків однорідної системи алгебраїчних рівнянь, до якої зводяться граничні умови задачі.

### Опис методу

Введемо в розгляд рівномірну сітку вузлів колокації на гранях пластини:

$$\begin{aligned}
 P &= P_1 \cup P_2 \cup P_3 \cup P_4, \\
 P_1 &= \left( x = 0; y = \frac{a \cdot i}{(N+1)} \right), \\
 P_2 &= \left( x = \frac{a \cdot i}{(N+1)}; y = 0 \right), \\
 P_3 &= \left( x = a; y = \frac{a \cdot i}{(N+1)} \right), \\
 P_4 &= \left( x = \frac{a \cdot i}{(N+1)}; y = a \right), \quad i = 1..N.
 \end{aligned}$$

На рис. 1 зображена сітка вузлів  $P$  для випадку  $N = 4$ .

Невідома функція пружних переміщень  $w(x, y)$  розшукується у вигляді:

$$w(x, y) = \sum_{i,j=1}^{32} F_i(x_j, y_j, \lambda, \omega) \cdot b_{i,j}, \quad (5)$$

де функції  $F_i(x, y)$  - часткові розв'язки диференціального рівняння (1),  $b_{i,j}$  - невідомі коефіцієнти.

Розглянемо наступний набір з 16 функцій  $F_i(x, y, \lambda, \omega)$ :

$$\begin{aligned}
 F_1 &= e^{\lambda x} \cos \varphi y, \quad F_2 = e^{\lambda x} \sin \varphi y, \quad F_3 = e^{\lambda x} \cosh \psi y, \quad F_4 = e^{\lambda x} \sinh \psi y, \\
 F_5 &= e^{-\lambda x} \cos \varphi y, \quad F_6 = e^{-\lambda x} \sin \varphi y, \\
 F_7 &= e^{-\lambda x} \cosh \psi y, \quad F_8 = e^{-\lambda x} \sinh \psi y, \\
 F_9 &= \cos \lambda x \cosh \varphi y, \quad F_{10} = \cos \lambda x \sinh \varphi y, \\
 F_{11} &= \cos \lambda x \cos \psi y, \quad F_{12} = \cos \lambda x \sin \psi y, \\
 F_{13} &= \sin \lambda x \cosh \varphi y, \quad F_{14} = \sin \lambda x \sinh \varphi y, \\
 F_{15} &= \sin \lambda x \cos \psi y, \quad F_{16} = \sin \lambda x \sin \psi y,
 \end{aligned} \quad (6)$$

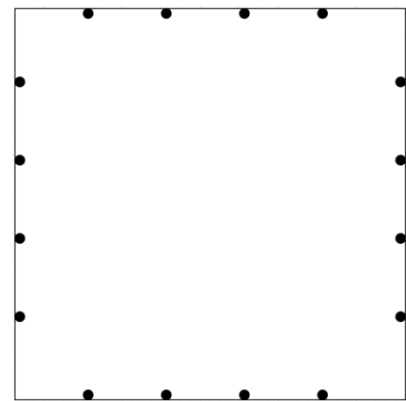


Рис. 1. Сітка вузлів колокації



де  $\varphi = \sqrt{\omega^2 + \lambda^2}$ ,  $\psi = \sqrt{\omega^2 - \lambda^2}$ ,  $\lambda^2 = k\omega^2$  - допоміжний параметр, в якому множник  $k$  варіюється в межах від 0 до 1.

Наступні 16 функцій  $F_i(x, y, \lambda, \omega)$ , ( $i = 17..32$ ) отримуються на основі формул (6) шляхом заміни  $x \rightarrow y$ ,  $y \rightarrow x$ ,  $\lambda \rightarrow \gamma$ , де  $\gamma = \sqrt{\omega^2 - \lambda^2}$ .

Представлення (5), (6) разом з формулами (3) використовуються далі для запису граничних умов (2а)-(2г), які, в свою чергу, призводять до однорідної системи 32 лінійних алгебраїчних рівнянь відносно невідомих коефіцієнтів  $b_{i,j}$ :

$$A \cdot b_{i,j} = 0. \quad (7)$$

Елементи  $a_{i,j}$  матриці  $A$  обчислюються за формулами:

$$a_{i,j} = F_i(x_j, y_j), \quad i = 1..32, \quad j = 1..16, \quad (x_j, y_j) \in P;$$

$$a_{i,j} = \frac{d^2 F_i(x_j, y_j)}{dy^2}, \quad i = 1..32, \quad j = 17..24, \quad (x_j, y_j) \in \{P_1, P_3\};$$

$$a_{i,j} = \frac{d^2 F_i(x_j, y_j)}{dx^2}, \quad i = 1..32, \quad j = 25..32, \quad (x_j, y_j) \in \{P_2, P_4\}.$$

Однорідна система (7) має нетривіальні розв'язки лише в тому випадку, коли її детермінант дорівнює нулю. Ця умова призводить до трансцендентного рівняння

$$\Delta(\omega) = 0, \quad (8)$$

що дозволяє визначити невідомі власні частоти  $\omega$ .

Побудоване частотне рівняння (8) досліджувалось чисельно. Встановлено, що його корені можуть бути двох типів — прості та двократні. На рис. 2 показана поведінка функції  $\Delta(\omega)$  в околі коренів двох різних типів. Уточнення коренів проводилось із застосуванням відомих чисельних процедур.

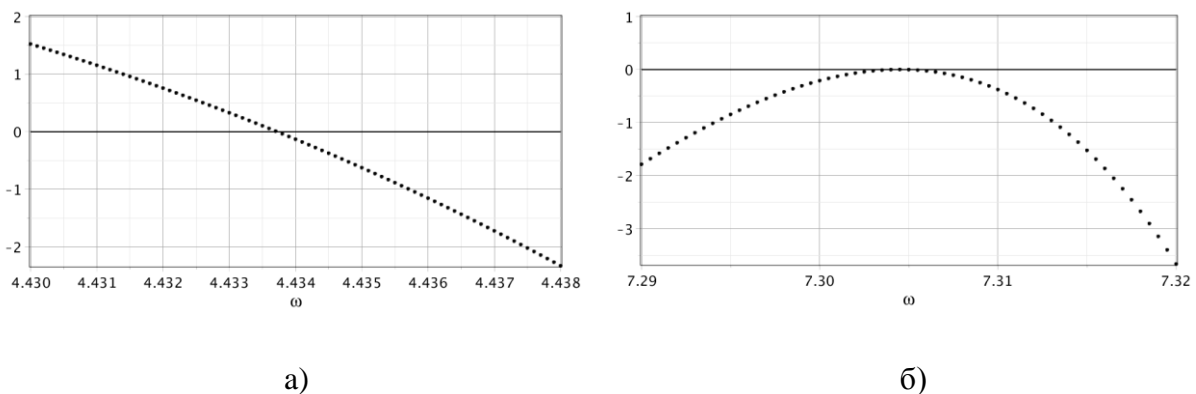


Рис. 2. Поведінка функції  $\Delta(\omega)$  в околі простого (а) та двократного (б) кореня

В таблиці 1 представлені розраховані на основі рівняння (8) власні частоти  $\omega$  для квадратної пластини та для порівняння - значення частот,

отримані за формулою (4б). Оскільки робота мала пошуковий характер, то довільним чином було обрано  $\lambda^2 = 0.8\omega^2$ . Як показують проведені розрахунки, тільки перша частота визначається із задовільною точністю, наступні частоти суттєво відрізняються від очікуваних. Запропонований підхід може бути вдосконалений за рахунок вибору іншої множини функцій  $F_i(x, y, \lambda, \omega)$  та збільшення кількості вузлів колокації.

Таблиця 1

Нормовані власні частоти для шарнірно-опертої квадратної пластини

$\{n, m\}$	Частоти, розраховані на основі рівняння (8)	Частоти, розраховані за формулою (4б), [1]
{1, 1}	4.4337	4.4428
{1, 2}, {2, 1}	7.3045	7.0248
{2, 2}	8.1036	8.8857
{2, 3}, {3, 2}	8.5797	9.9345
{3, 3}	9.9689	13.3286
{3, 4}, {4, 3}	10.0223	14.0496

## Висновки

В роботі запропоновано підхід для дослідження задачі про власні коливання прямокутної пластини, який ґрунтується на застосуванні точних аналітичних розв'язків та методу колокації. Числові результати отримані для випадку квадратної пластини, що має шарнірне обпирання на всіх гранях. Проведено співставлення отриманих результатів з відомими в літературі. Запропонований підхід може поширений на інші граничні умови закріплення граней пластини. Відмінність та перевага методу від існуючих, полягає в тому, що шукана частота є внутрішнім параметром побудованих функцій, тому вони дають необмежене число частот при обмеженій кількості вибраних функцій.

## Література

1. *Leissa A. W.* The free vibration of rectangular plates // *J. Sound Vibr.* – 1973. – 31. – P. 257–293.
2. *Gorman D. J.* Free vibration analysis of the completely free rectangular plate by the method of superposition // *J. Sound Vibr.* – 1978. – Vol. 57. – P. 437–447.
3. *El-gamel, M., Mohsen, A., Abdrabou, A.* Sinc-Galerkin solution to the clamped plate eigenvalue problem // *SeMA Journal.* – 2017. – Vol. 74. – P. 165–180.

4. *Orynyak I. V., Bai Y. P.* Application of exponential functions in weighted residuals method in structural mechanics. Part II: static and vibration analysis of rectangular plate // *Mechanics and Advanced Technologies*. – Vol. 91, № 1. – 2021. – P. 7–21.

**К. ф.-м. н., доц. Вовк Л.Б., студент Грідько О.С.**

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

## **СИСТЕМА АНАЛІЗУ ТЕНДЕНЦІЙ КРИПТОВАЛЮТНОГО РИНКУ ЗА ДОПОМОГОЮ ЧАСОВИХ РЯДІВ**

### **Abstract**

**Liliia Vovk, assoc.prof., PhD; Oleksandr Hrydko, student**

*System of cryptocurrency market trends analysis using time series.*

*This paper presents the description of cryptocurrency market trends analysis based on time series. Here we will consider general information about cryptocurrencies, different variations of time series, artificial neural networks and their application in the cryptocurrency market. The most accurate methods for this problem solution are proposed. The ways for further research are mentioned as well.*

### **Вступ**

Швидкий розвиток інформаційних технологій, розвиток глобалізації та загальної комп'ютеризації призводить до змін у сучасній економіці. Особливо це стосується змін у фінансовій системі – сфері платежів та розрахунків, від якої напряму залежить розвиток економіки та матеріальне забезпечення країни. Спочатку використовували металеві монети, потім паперові банкноти, а зараз – онлайн транзакції. У період нестабільності та недовіри до фінансової системи, збільшується інтерес до віртуальних грошей або криптовалюти.

Однією з переваг криптовалюти є прозорість транзакцій. Технологія блокчейн, на якій побудована криптовалюта, це вид бази даних в яку можна тільки вносити інформацію і немає можливості редагувати чи видаляти записи. Дана система добре захищена від зломів та шахрайства, що є надійним методом ведення облікових записів у будь-якій сфері. Це особливо важливо в країнах, де високо розвинута корупція.

Тому важливо розуміти як ріст цін на криптовалютному ринку може вплинути на світову економіку і як бути готовим до цього.

### **Постановка задачі**

Задача полягає у розробці системи аналізу тенденцій криптовалютного ринку, за допомогою якої можна буде отримати прогноз на подальший розвиток ціни певної криптовалюти.

На вхід система буде отримувати дані про ціну певної криптовалюти за визначений період часу, результатом буде – прогноз ціни.

Запропонований програмний модуль за даною тематикою складається з наступних окремих частин:

а) Обробка вхідних даних. На вхід будуть подаватись дані про bitcoin за певний період часу. Корисність даного модуля полягає в форматуванні некоректно введених або пропущених даних;

б) Побудова моделей. На даному етапі буде побудовано обрані моделі на основі стохастичних моделей та штучних нейронних мереж;

в) Визначення точності;

г) Візуалізація даних;

### **Огляд математичних методів**

Нижче наведені методи прогнозування, які будуть використовуватися в даній роботі:

- Авторегресія з ковзним середнім

Модель авторегресії є ефективним інструментом для розуміння і прогнозування наступних значень часового ряду, яка включає в себе регресування змінної по значеннях ряду у минулому. Об'єднавши дану модель з моделлю ковзного середнього, отримаємо ARMA (p, q), яка використовується для прогнозування стаціонарних часових рядів [1].

- Авторегресія з інтегрованим ковзним середнім

Головним недоліком моделі ARMA, яка описана вище є те, що описаний ряд має бути стаціонарним. Однак на практиці багато часових рядів, які відносяться до різних сфер – економічної, фінансової, навіть та ж ціна bitcoin, можуть містити тренд або мати сезонність, тобто є нестаціонарними за своїм характером. Таким чином, використання моделі ARMA є недостатнім для коректної роботи та отримання хороших результатів в цій роботі. Тому пропонується модель ARIMA(p, d, q), яка вирішує дану проблему за рахунок порядку інтегрованості d [2].

- Нейронна мережа

Нейронна мережа – це послідовність нейронів, зв'язаних між собою. Нейрон це обчислювальна одиниця, яка отримує на вхід дані, проводить над ними обчислення та передає далі. Нейронна

мережа поділяється на три основних шари – вхідний, прихований, вихідний(результат), що показано на наступному рисунку 1.

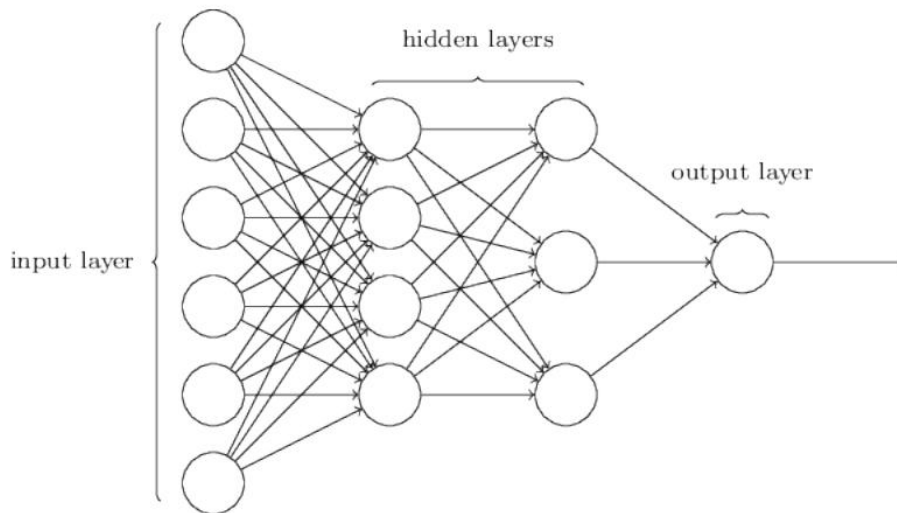


Рис. 1. Архітектура нейронної мережі

Нейронні мережі використовуються для класифікації, передбачення та розпізнавання даних. Одні з найвідоміших:

- 1) згорткова нейронна мережа CNN, зазвичай використовується для роботи з фотографіями;
- 2) рекурентна нейронна мережа RNN використовується для обробки послідовних даних, таких як часові ряди, текстові дані, фінансові дані, аудіо дані.

Однією з найвідоміших RNN є мережа довго коротко-строкової пам'яті LSTM, її перевагою є те, що дана мережа краще працює з довготривалими залежностями (запам'ятовування інформація на довгий період).

### Опис алгоритму

- 1) Попередня обробка даних включає в себе нормалізацію – це зміна масштабу даних із вихідного діапазону таким чином, щоб усі значення знаходилися в межах 0 та 1. Для побудови моделей прогнозування потрібно розбити вхідні дані на два списки – навчальний та тестовий. Навчальний датасет використовується для навчання моделі, тестовий – визначення точності роботи моделі.
- 2) Побудова моделей ARIMA та LSTM. Для моделі ARIMA(p, d, q) потрібно підібрати оптимальні параметри p, d, q. В результаті маємо отримати передбачення для різних криптовалют.

3) Візуалізація результатів. На графіку мають міститись реальні дані за певний період часу та результат передбачення за той же самий період часу, результат роботи моделей ARIMA та LSTM показано на рисунках 2 та 3.

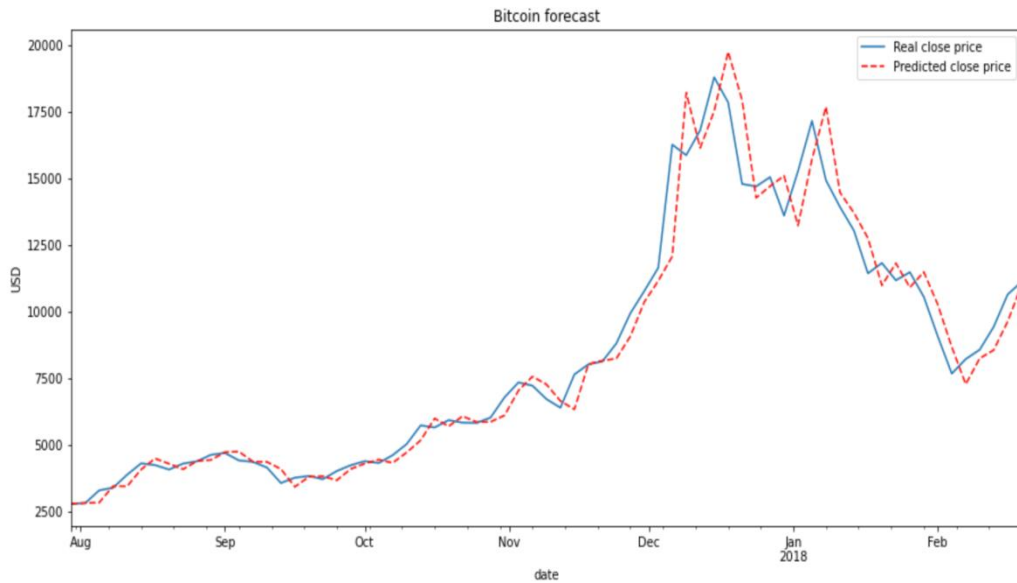


Рис. 2. Результат роботи ARIMA моделі

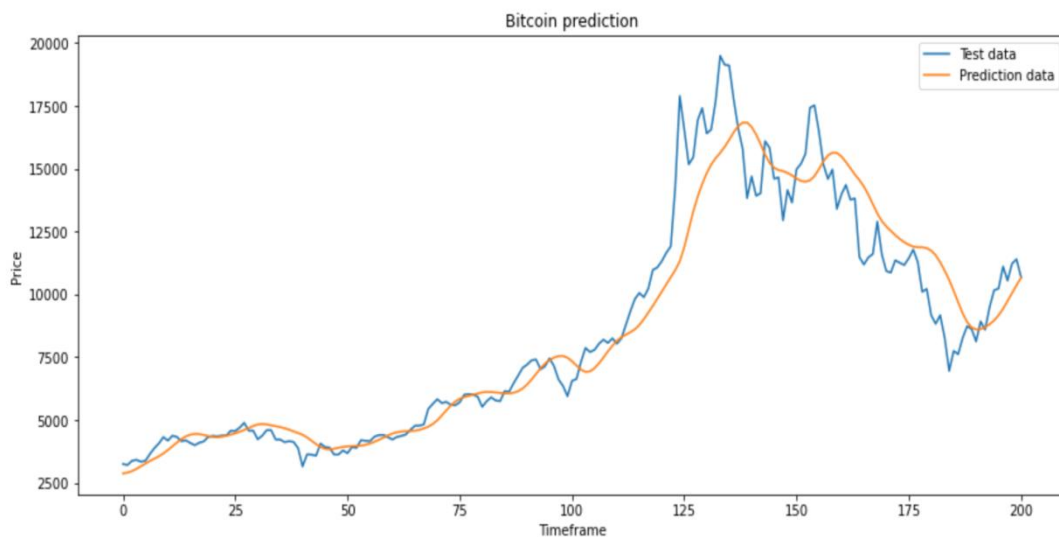


Рис. 3. Результат роботи LSTM

4) Точність. Для оцінки роботи даних моделей, в даному дослідженні буде наведена таблиця з середньо квадратичними та середньо-абсолютними помилками.

## **Висновки**

У даній роботі розглянуто стохастичні моделі та нейронна мережа на основі часових рядів, їх застосування та різні варіації. Було обрано моделі ARIMA та LSTM, як методи, які показують найкращу точність для цього завдання.

Метою наступних досліджень є проведення експериментів та покращення запропонованих методів для збільшення точності прогнозування.

## **Література**

- 1) Time series analysis forecasting and control/ George E. P. Box Gwilym M. Jenkins Gregory C. Reinsel Greta M. Ljubg – 2016. – 659p.
- 2) Аналіз часових рядів/ Бідюк П.І., Романенко В. Д., Тимошук О.Л. – 2010. – 300с.
- 3) Neural network and deep learning/ Michael Nielsen – 2018. – 211p.



**Канд. фіз.-мат. наук Вовк Л.Б., студентка Мілевська О.О.**

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

## **МАТЕМАТИЧНІ МОДЕЛІ В ЕПІДЕМІОЛОГІЇ ДЛЯ ПРОГНОЗУВАННЯ COVID-19**

### **Abstract**

**Liliia Vovk, assoc. prof., PhD; Olha Milevska, student**

***Mathematical Models in Epidemiology for Covid-19 Forecasting***

*It is difficult to anticipate the size of the population that may get affected by the COVID-19 and the typical time required for its control as the world haven't faced a similar pandemic with such a high impact in the recent past. The pandemic requires a quantitative understanding of the time evolution of this complex and non-linear process through computer modeling. Statistical and mathematical analysis of reported data can provide valuable insights into the trend of the spread and thus can assist in planning various social measures to contain the spread of the virus as quickly as possible.*

### **Вступ**

У зв'язку з ситуацією в світі, математичні моделі є як ніколи необхідними для оцінки і аналізу кількості інфікованих, шляхів передачі захворювання, процесу одужання, смертності та інших значущих параметрів, які варіюються в залежності від рівня зареєстрованих випадків COVID-19.

### **Постановка задачі**

В багатьох країнах світу було проведено дослідження хвороби та вжито конкретних та різноманітних заходів, які є важливими для контролю поширення хвороби. Проте зараз такі важливі фактори, як щільність населення, недостатні докази різних симптомів, механізм передачі та відсутність належної вакцини, ускладнюють боротьбу з такою високо-інфекційною та смертельною хворобою [1].

У зв'язку з цим необхідно дослідити та проаналізувати підходи для прогнозування вірусних інфекцій. Математичне моделювання в епідеміології дає можливість змоделювати появу й поширення хвороботворних мікроорганізмів. У цій роботі ми представляємо огляд та короткий аналіз найважливіших епідеміологічних моделей для прогнозування COVID-19, а також наведено висновки як заключний розділ цього дослідження.

## Теоретичні основи моделювання

Поширення вірусу залежить від багатьох факторів, деякі з яких розглядаються класичною моделлю **SIR**, яка є досить спрощеною і не може врахувати зростання кількості сприйнятливих осіб. Зокрема, одне з основних припущень класичної моделі SIR полягає в тому, що існує однорідне змішування інфікованих  $I$  та чутливих  $S$  популяцій і що загальна популяція  $N$  є постійною в часі. Крім того, передбачається, що часовий масштаб моделі SIR досить короткий, щоб можна було знехтувати народженням і смертями (крім смертей, спричинених вірусом), і що кількість смертей від вірусу невелика в порівнянні з живим населенням. [2].

**Susceptible** ( $S$ , Сприйнятливі особи): це ті особи, які не інфіковані, однак мають ризик зараження. Сприйнятлива особа може заразитися або залишитися сприйнятною. У міру того, як вірус поширюється від свого джерела або з'являються нові джерела, інфікується все більше людей, отже, сприйнятлива популяція зростає на певний період часу (період сплеску).

**Infectious** ( $I$ , Інфіковані особи): це ті особи, які вже були інфіковані вірусом і можуть передати його тим особам, які є сприйнятливими. Інфікована людина може залишатися інфікованою і може бути вилучена з інфікованої популяції, у випадку одужання або смерті.

**Recovered** ( $R$ , Одужалі особи): це ті особи, які вже були інфіковані вірусом і тепер одужали, отримавши імунітет [3].



Рис. 1. Модель епідемії SIR

Модель SIR представлена системою трьох звичайних диференціальних рівнянь:

$$\frac{dS(t)}{dt} = -\beta I(t) \frac{S(t)}{N} \quad (1)$$

$$\frac{dI(t)}{dt} = \beta I(t) \frac{S(t)}{N} - \gamma I(t) \quad (2)$$

$$\frac{dR(t)}{dt} = \gamma I(t) \quad (3)$$

де  $\beta$  - швидкість зараження,  $\gamma$  –це швидкість одужання.

Модель **SEIR** моделює хронологію епідемічного явища. У своїй класичній формі моделює взаємну та динамічну взаємодію людей між чотирма різними станами: сприйнятливими ( $S$ ), експонованими ( $E$ ), інфікованими ( $I$ ) і одужалими ( $R$ ).

**Exposed** ( $E$ , Експонована, незахищена) частина населення, яка була інфікована, але ще не має симптомів: її можна назвати латентною фазою, і на цій стадії захворювання може бути інфекційним, частково інфекційним або неінфекційним [4].

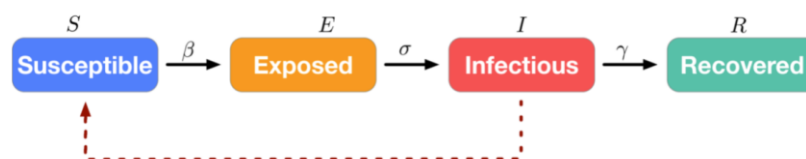


Рис. 2. Модель епідемії SEIR

Класичну модель SEIR можна описати системою звичайних диференціальних рівнянь:

$$\frac{dS(t)}{dt} = -\beta I(t) \frac{S(t)}{N} \quad (4)$$

$$\frac{dE(t)}{dt} = \beta I(t) \frac{S(t)}{N} - \sigma E(t) \quad (5)$$

$$\frac{dI(t)}{dt} = \sigma E(t) - \gamma I(t) \quad (6)$$

$$\frac{dR(t)}{dt} = \gamma I(t) \quad (7)$$

де  $\beta$  - швидкість зараження,  $\gamma$  - швидкість одужання,  $\sigma$  - відставання між інфекційним контактом і проявом симптомів.

У моделі SEIR  $\beta$  є вектором, який переносить людей з категорії  $S$  в категорію  $E$ , а  $\sigma$  переносить людей з категорії  $E$  в категорію  $I$  [5]. Характеристикою цієї моделі є те, що сума чотирьох категорій дорівнює загальній популяції ( $N$ ) у будь-який момент часу:

$$S(t) + E(t) + I(t) + R(t) = N \quad (8)$$

У класичній моделі SEIR категорія  $R$  також охоплює осіб, які померли від хвороби, а також не враховує природні народжуваність і смертність населення протягом періоду захворювання [6].

Модель **SIRD** подібна до вищеописаних моделей, проте також враховує категорію **Deceased** ( $D$ , померлі). Модель SIRD представлена системою звичайних диференціальних рівнянь:

$$\frac{dS(t)}{dt} = -\beta \frac{S(t)I(t)}{S(t) + I(t)} \quad (9)$$

$$\frac{dI(t)}{dt} = \beta \frac{S(t)I(t)}{S(t) + I(t)} - \gamma I(t) - \varepsilon I(t) \quad (10)$$

$$\frac{dR(t)}{dt} = \gamma I(t) \quad (11)$$

$$\frac{dD(t)}{dt} = \varepsilon I(t) \quad (12)$$

де  $\beta$  - швидкість зараження,  $\gamma$  - швидкість одужання,  $\varepsilon$  - рівень

смертності.

Основні гіпотези цієї моделі полягають у тому, що одужалісуб'єкти більше не сприйнятливі до інфекції, і що кількість смертей через інші причини (відмінні від захворювання, що розглядається) нехтується. Крім того, припускається, що регіон, що розглядається, ізольований від інших регіонів, і це є розумним припущенням, коли застосовуються заходи стримування, такі як заборона на подорожі [7].

## Висновки

Сьогодні використовуються декілька підходів для прогнозування спалаху COVID-19. Серед запропонованих дослідниками підходів найпопулярнішими є SIR, SEIR, SIRD.

У момент часу  $t$  населення поділяється на чотири класи: сприйнятливий, інфікований, одужалий і мертвий, позначені відповідно  $S(t)$ ,  $I(t)$ ,  $R(t)$  і  $D(t)$ . Оскільки COVID-19 - це нове захворювання, попередній імунітет не існує, отже, кожна людина вразлива до COVID-19. Після зараження недугою сприйнятливі особи переходять до інфікованого класу, після чого або одужують, або вмирають від інфекції.

Враховуючи складність захворювання, багато авторів та дослідників реалізували різні варіації класичних моделей SIR, SEIR, SIRD, що стосуються як рівнянь, так і параметрів, або користувались різними методами підгонки, щоб зробити модель, що представляє реальність, якомога ближче. Проте SIR, SEIR, SIRD є базовими математичними моделями епідемії та беруться за основу будь-яких досліджень.

## Література

1. Ian C. A SIR model assumption for the spread of COVID-19 in different communities [Електронний ресурс] / С. Ian, М. Argha – Режим доступу до ресурсу: <https://www.sciencedirect.com/science/article/pii/S0960077920304549>.
2. COVID-19 dynamics with SIR model [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://www.lewuathe.com/covid-19-dynamics-with-sir-model.html>.
3. UCLA Statistical Machine Learning Lab. Learning Epidemic Models for COVID-19 [Електронний ресурс] / UCLA Statistical Machine Learning Lab. – 2020. – Режим доступу до ресурсу: <https://covid19.uclaml.org/model.html>.
4. Shi P. SEIR Transmission dynamics model of 2019 nCoV coronavirus with considering the weak infectious ability and changes in latency duration. / Shi P, Cao S, Feng P., 2020.
5. Peng L. Epidemic analysis of COVID-19 in China by dynamical modeling / Peng L, Yang W, Zhang D., 2020.

6. Alberto Godio. SEIR Modeling of the Italian Epidemic of SARS-CoV-2 Using Computational Swarm Intelligence [Электронный ресурс] / Alberto Godio, Francesca Pace, Andrea Vergnano. – 2020. – Режим доступа до ресурсу:  
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7277829/>.
7. Giuseppe C. Calafiore. A time-varying SIRD model for the COVID-19 contagion in Italy [Электронный ресурс] / Giuseppe C. Calafiore, Carlo Novara, Corrado Possieri. – 2020. – Режим доступа до ресурсу:  
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7587010/>.

Асистент Ковальчук-Химюк Л.О., бакалаврант Юдін Г.Є.

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

## АНАЛІЗ МЕТОДІВ ВИЗНАЧЕННЯ ЕМОЦІЙНОГО ЗАБАРВЛЕННЯ ПІСЕНЬ

### Abstract

**Ludmila Kovalchuk-Khymiuk, assistant; Hlib Yudin, student**

*Analysis of methods for defining emotional colouring of songs*

*This paper conducts the analysis of existing methods for defining moods of songs, which may be useful for recommending songs in music streaming services. First, the notion of «mood of a song» is formalized. Then, different methods are classified by the type of input data (text versus audio) and examined.*

### Вступ

Музика стала невід’ємною частиною життя людей у сучасному світі. Щоб насолоджуватися улюбленими піснями та відкривати для себе нові, використовуються спеціальні музичні платформи (Spotify, Youtube Music, Apple Music тощо). Для рекомендації нових пісень застосовуються різні підходи та алгоритми, зокрема: колаборативне фільтрування, пошук подібних виконавців, пошук за жанром, аналіз трендів. Натомість не розглядається така їхня важлива складова, як емоційне забарвлення. Музика допомагає людині вивільнити її емоції, пережити емоційно складні моменти в її житті, відчувати, що вона не самотня в своїх почуттях, що інші люди відчувають те саме. Музичні стримінгові платформи дозволяють здійснювати пошук плейлістів на основі ключових слів (наприклад, збірка сумних пісень — «sad songs»), проте ці плейлісти створюються людьми вручну. Було б зручно мати спосіб автоматично визначити емоційне забарвлення пісні, щоб таким чином поліпшити якість рекомендацій та пошуку пісень (наприклад, визначати настрій прослуханих пісень та пропонувати нові пісні з таким самим настроєм).

### Постановка задачі

Метою даної роботи є аналіз наявних методів визначення емоційного забарвлення музичних композицій для подальшого створення власної системи розпізнавання настрою пісень.

### **Формалізація емоційного забарвлення**

Оскільки дана робота присвячена розпізнаванню емоційного забарвлення (настрою) пісень, то перш за все потрібно формально описати поняття «настрій пісні», щоб пізніше збудувати на його основі математичну модель визначення пісенних емоцій.

Найбільш очевидним є якісний спосіб визначення цього поняття: окремі настрої можна описати як окремі класи. А отже, задача визначення емоційного забарвлення пісні зводиться до задачі класифікації: встановити, до якого класу/класів (до якого настрою) належить дана пісня. Можливою проблемою є недвозначність класів: за цим принципом, пісня або може належати до певного класу настроїв, або ні, хоч на практиці трапляються двозначні випадки. Її можна виправити введенням нових класів (наприклад, «меланхолія» і «сум»), але існує ризик створити їх надто багато — людині буде складно віднести пісню до одного конкретного класу з багатьох схожих. Ще один варіант — ввести нечітку математику і функцію приналежності музичної композиції до даного класу [1].

Для кількісного визначення емоційного забарвлення можна скористатися моделями людських настроїв, складених психологами Thayer [2] та Russell [3]. Підхід, на якому базуються ці моделі, полягає у тому, що всі настрої можна охарактеризувати двома числовими неперервними величинами: валентність (valence/stress — наскільки сумним чи щасливим є настрій) та збудження (arousal/energy — наскільки енергійним є настрій). Ці величини, що позначаються VA, коливаються в інтервалі  $[-1; 1]$ , різні значення обох цих величин визначають різні настрої. Таким чином, дана задача зводиться до задачі регресії, оскільки вихідним сигналом є дві неперервні числові величини. Між дискретною та неперервною моделями простежується сильний зв'язок: «*Linear mapping techniques between the discrete and dimensional models revealed a high correspondence along two central dimensions that can be labelled as valence and arousal.*» [4].

### **Математичні моделі визначення настрою пісні (вербальний підхід)**

Насамперед варто зазначити, що існують два різні підходи до задачі визначення емоційного забарвлення треку: з використанням акустичної інформації (мелодії) та вербальної (тексту пісні). Автори вважають, що більший потенціал мають акустичні підходи. По-перше, текст пісні не завжди є у доступі, особливо коли пісня нова. А по-друге, пісня може мати мало слів і все одно нести емоційне забарвлення (також розглядаються випадки, коли слухач просто не розуміє чи не розбирає текст пісні). Утім вербальні підходи все одно мають право на існування, особливо у

поєднанні з акустичними підходами. Перед самим застосуванням моделей тексти пісень зазвичай нормалізують (позбуваються незначущих частин слова, як-от закінчення чи суфікс).

Одним з найпростіших вербальних підходів є наївний баєсівський класифікатор [5, 6], який побудований на формулі Баєса та припущенні, що події входження слова в текст пісні незалежні. За цією моделлю, для пісні  $s$  серед усіх настроїв  $m$  обирається такий, для якого імовірність  $P(m|L)$  приналежності пісні з текстом  $L$  до настрою  $m$  найвища:

$$m_{select}(s) = \operatorname{argmax}_m P(m|L) = \operatorname{argmax}_m P(m) \prod_{w \in L} P(w|m)^{TF(w,L)},$$

де  $w$  — слово з пісні,  $O(w, m)$  — частота слова  $w$  в тексті  $m$ . Імовірність  $P(w|m)$  того, що пісня з настроєм  $m$  містить слово  $w$ , обчислюється наступним чином:

$$P(w|m) = \frac{1 + O(w,m)}{|F| + \sum_{w \in F} O(w,m)},$$

де  $O(w, m)$  — частота спільної появи слова  $w$  і настрою  $m$ ,  $F$  — множина всіх слів з тренувального набору. Модель можна доповнити, якщо додати вагу словам з приспіву й назви пісні або якщо додатково розглянути інформацію про виконавця.

Окрім баєсівського класифікатора, для розпізнавання настрою також використовують SVM-класифікатор (support vector machine) [5]. Його суть — підібрати  $n-1$ -розмірну гіперплощину, щоб розділити наявні  $n$ -розмірні навчальні дані та щоб відстань між гіперплощиною та найближчою точкою була максимальною. Навчання відбувається шляхом мінімізації функції вартості. Якщо розглянути контекст даної задачі — текст кожної пісні представляють у вигляді числового вектора,  $i$ -та координата якого — важливість/вага слова  $w_i$  в даній пісні, розмірність вектора — загальна кількість різних слів у тренувальному наборі даних. Вагу слова  $w$  в даній пісні визначають так:

$$\text{weight}(w) = \text{tfidf}(w) = \text{tf}(w) * \text{idf}(w)$$

Тут  $\text{tf}(w)$  — частота слова (term frequency), відношення числа входження слова  $w$  в текст пісні до числа слів в пісні,  $\text{idf}(w)$  — зворотна частота документу (inverse document frequency), відношення загальної кількості пісень до кількості пісень, в яких зустрічається слово  $w$ . Перший множник оцінює важливість слова в пісні, другий визначає, наскільки поширеним є слово серед усіх пісень (чим більше воно поширене, тем менша його важливість).

В роботі [7] також описується застосування методу найближчих  $k$  сусідів ( $k$ -nearest neighbours). Його суть проста: щоб встановити, до якого класу належить нова пісня, варто перевірити  $k$  найближчих пісень до неї. Якщо серед них переважають пісні одного настрою, то новій пісні



присвоюється цей самий настрій. Для оцінки близькості застосовують різні метрики (наприклад, евклідову). Пісні представляють як вектори параметрів, до яких належать кількість рядків тексту, слів, символів, значень  $tf$ ,  $idf$  та  $tfidf$ .

### Математичні моделі визначення настрою пісні (акустичний підхід)

Якщо для вербального підходу вхідними даними є слова і пов'язані з ними вектори, то в акустичному підході розглядаються звукові характеристики пісні. Однією з можливих матмоделей є модель гаусової суміші (Gaussian mixture model, GMM) [8]. Згідно з нею, усі наявні дані згенеровані з суміші скінченного числа нормально розподілених випадкових величин з невідомими параметрами. Навчання моделі полягає у визначенні цих параметрів. В якості вхідних даних використовують значення інтенсивності (гучності), тембру та ритму пісні. Розпізнавання вхідного образу відбувається за наступним ієрархічним алгоритмом: спочатку обчислюється імовірність того, що пісня з гучністю  $I$  належить до групи настроїв  $G_1$  (задоволення і депресія) або  $G_2$  (захват і тривога). Після встановлення групи настроїв, до яких належить пісня, обчислюється імовірність того, що пісня з тембром  $T$  і ритмом  $R$  має кожен окремих настрій з обраної групи. Дана модель є ієрархічною (спочатку визначається група настроїв, потім — сам настрій), але можливий і неієрархічний підхід.

Також у [8] обґрунтовується використання SVM-класифікатора, але тепер в якості точок  $n$ -вимірному простору виступають не вектори важливості слів, як у вербальному підході, а вектори звукових характеристик (як-от гучність, тембр і темп). Але оскільки кожному пісню чи кожен уривок пісні можна представити у вигляді кількох числових параметрів (звукових характеристик), ніщо не заважає зробити узагальнення та використовувати замість SVM будь-який інший метод машинного навчання для задач регресії. Наприклад, для передбачення значень валентності й збудження можна використати логістичну регресію. Маючи вектор ознак  $x$ , модель обчислює вихідний сигнал за формулою: 
$$h(x) = \frac{1}{e^{-w^T x} + 1}$$
 де  $w$  — вектор вагів, що встановлюється під час навчання. Отримане значення лежить в межах від 0 до 1. Щоб модель машинного навчання змогла навчитися більш складних функцій (які відображають вектор звукових характеристик на вектор VA), можна застосувати нейронну мережу (наприклад, стандартний багатошаровий перцептрон).

Під час отримання звукових характеристик зазвичай відкидається велика частина даних (неструктурований аудіофайл зводиться до кількох числових значень). Цілком можливо, що при цьому втрачаються важливі

дані або взагалі обчислюються ті звукові характеристики, що не несуть практичної цінності. У такому випадку доцільно застосувати глибинні нейронні мережі, які здатні самостійно виділяти необхідні звукові характеристики. До них належить згорткова нейронна мережа (її також використовує сервіс Spotify для рекомендації пісень). Мережі цього типу зазвичай використовують для задач комп'ютерного зору, але їх можна використовувати і для задач з аудіофайлами. При цьому вхідними даними можуть бути спектрограма пісні, мел-спектрограма, мел-частотні кепстральні коефіцієнти тощо [9]. Мережа, яку використовує Spotify, складається з чотирьох згорткових шарів, шару global temporal pooling та трьох повнозв'язних шарів, на виході повертає вектор з 40 значень, серед яких — гучність, темп, тактовий розмір, лад тощо.

## **Висновки**

Розглянуто різні методи для визначення емоційного забарвлення пісень, а також способи формалізувати саме поняття «настрій пісні». Загалом розглянуті методи можна розділити на дві групи: вербальні (використовують загальну інформацію про текст пісні, як-от кількість слів, вектори важливості слів чи інші характеристики) та акустичні (обробляють звукові характеристики або безпосередньо аудіофайли в неструктурованому вигляді). Найбільш поширеними наразі є методи з використанням штучних нейронних мереж, оскільки вони здатні самостійно визначати необхідні характеристики аудіофайлу, а даних для їх навчання дуже багато. Потенційну користь має варіант використовувати окремий метод для текстових даних, окремий — для аудіо. Загалом створення автоматизованої системи визначення настрою пісень поліпшить якість їх рекомендацій у музичних сервісах, а також допоможе композиторам-початківцям створювати музику з конкретним емоційним забарвленням.

## **Література**

1. *Yi-Hsuan Yang. Music Emotion Classification: A Fuzzy Approach / Yi-Hsuan Yang, Chia-Chu Liu, Homer H. Chen // Proceedings of the 14th ACM International Conference on Multimedia, Santa Barbara, CA, USA. — 2006. — P. 81 - 84.*
2. *Robert E. Thayer. The Biopsychology of Mood and Arousal / Robert E. Thayer. — Oxford : Oxford University Press. — 1990. — 256 p.*

3. *Russell J. A.* A circumplex model of affect / Russell, J. A. // *Journal of Personality and Social Psychology*. — 1980. — Vol. 39, № 6. — P. 1161 - 1178.
4. *Eerola T.* A comparison of the discrete and dimensional models of emotion in music / Eerola T., Vuoskoski J. K. // *Psychology of Music*. — 2011. — Vol. 39, № 1. — P. 18 - 49.
5. *T. Dang.* Machine Learning Approaches for Mood Classification of Songs toward Music Search Engine / T. Dang, K. Shirai. — Hanoi : 2009 International Conference on Knowledge and Systems Engineering. — 2009. — P. 144 - 149.
6. *S. Raschka.* MusicMood: Predicting the mood of music from song lyrics using machine learning / S. Raschka. — Michigan : Michigan State University. — 2014. — 9 p.
7. *P.W.M. Kanters.* Automatic Mood Classification for Music : HAIT Master Thesis series nr. 09-001 / P.W.M. Kanters. — Tilburg : Tilburg centre for Creative Computing (TiCC). — 2009. — 45 p.
8. *Sonal P.Sumare.* Automatic Mood Detection of Music Audio Signals: An Overview / Sonal P.Sumare, D.G.Bhalke // *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)*. — 2015. — P. 83 - 87.
9. Devopedia. 2021. "Audio Feature Extraction." Version 8, May 23. Accessed 2021-09-09. <https://devopedia.org/audio-feature-extraction>

**Д.ф.-м.н., проф. Лось В.М., магістрантка Без'язична К.Ю.**

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

## **МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ПРОГОЗУВАННЯ ЗАХВОРЮВАНОСТІ НА КІР З ВИКОРИСТАННЯМ МЕТОДУ ВАЛЬДА**

### **Abstract**

**Los Valerii, prof., Dr.S.; Beziazychna Kateryna student**  
*Mathematical and software system for predicting measles incidence using the Wald method*

*This paper presents the description of an algorithm used to predict incidence of measles using Wald's method. Forecasting methods are discussed. The most accurate method for solving this problem is proposed. The ways of further research are also indicated.*

### **Вступ**

Інфекційні захворювання турбують людство ще з давніх давен і по цей час залишаються однією з найбільших проблем. Статистика смертей від інфекційних хвороб, на даному етапі, уже перевищує у сотні або навіть у тисячі разів кількість жертв у наслідок воєнних сутичок в історію світу. На сьогоднішній день вчені доводять, що інфекційні хвороби можуть бути причиною виникнення більш складних захворювань, таких як розсіяний склероз, артрити та деякі форми раку, що до цих пір вважались “неінфекційними”[1].

Протягом останніх 40 років екологічний підхід до захворювань став основним поняттям до епідеміології. Серед усіх інфекційних захворювань – кір стала класичним прикладом успішного паразитування, що являється самообмежуючою інфекцією, де їй притаманна коротка тривалість, перетікання хвороби у стані середньої тяжкості та достатньо низький рівень летального результату, хоча медики стверджують, що хвороба має достатньо нестабільну поведінку[2]. Дивлячись на це, кір вважається й досі дуже небезпечною для людства оскільки немає комплексного підходу до її лікування.

Тож розуміючи складність перебігу хвороби, постає актуальним питання, щодо аналізу кількості захворювань на кір до цього часу та кількості випадків у наступні роки та століття. Знаючи, приблизний

прогноз кількості захворілих можна скласти детальний план по вакцинації, закупці ліків та захисних засобів, збільшення кількості ліжок для інфекційних відділень і можливість виграшу у часі для знаходження оптимального плану лікування хворих.

Способи моделювання інфекційної захворюваності починають свій активний розвиток ще з початку ХХ століття. За останні роки кількість робіт на цю тематику набирає швидких темпів, завдяки великій кількості, доступної для аналізу, статистичних даних.

Епідеміологічні прогнози виконуються на різні проміжки часу та в залежності від цього потрібні для різних ситуацій. Наприклад, короткочасні прогнози створені для оперативного управління та виявлення різких змін у протіканні захворюваності. Найкориснішими вважаються прогнози від 2 до 6 місяців, оскільки такий період дає можливість підготуватися до надзвичайних ситуацій і проведення мір безпеки для населення, але є менш точним ніж короткі прогнози, збільшення точності в більшості випадків неможливе, оскільки використовуються для оцінки необхідної кількості ліків та вакцинації для населення, підготовки лікарень та спеціалістів[3].

Для прогнозування захворюваності існує багато математичних методів, що діляться на декілька секцій:

1. Регресійні методи:
  - 1.1 Лінійна регресія
  - 1.2 Модель Серфлінга
  - 1.3 Багатофакторні поліноміальні регресійні моделі
  - 1.4 Авторегресійні моделі
  - 1.5 Експоненційне згладжування
2. На основі машинного навчання:
  - 2.1 Байєсівські мережі
  - 2.2 Штучні нейронні мережі
3. На базі фільтрацій:
  - 3.1 Вейвлет-декомпозиція

Задачею алгоритмів є прогнозування захворюваності на базі вже відомих даних, але кожен з алгоритмів має нюанси, щодо термінів для прогнозування та проміжку часу для аналізу вже відомих даних. Також деякі алгоритми мають досить складну інтерпретацію та обчислюваність, або повинні використовуватись у поєднанні з іншими алгоритмами прогнозування.

Для розроблюваного алгоритму використовуємо дані з доступної статистики Всесвітньої організації охорони здоров'я.

## **Мета**

Робота математичного та програмного забезпечення для прогнозування захворюваності на кір з використанням власного алгоритму розробленого на основі методу Вальда.

## **Термінологія**

**Критерій Вальда** – це критерій, що використовується в прийнятті рішень в умовах невизначеності.

**Точність прогнозу моделі** – близькість розрахункових значень до фактичних спостережень за період апроксимації.

**Рівень спорадичності** – це рівень захворюваності, випадки якого розподілені відносно рівномірно і не пов'язані епідемічною залежністю або самим джерелом інфекції.

## **Опис алгоритму (Основна частина).**

Запропонована методика аналізу та прогнозування захворюваності на кір включає в себе такі основні етапи послідовних дій:

1. Обробка вхідних даних у форматі csv.
2. Обрахування середніх значень кількості захворювань на кір на великому проміжку часу за кожний місяць.
3. Розраховуємо контрольний рівень спорадичності на кір за той самий період.
4. Для аналізу методом Вальда необхідно визначити початкові та кінцеві точки, а також контрольний рівень з пункту 3, для можливості визначення перевищення фактичного рівня.
5. Розраховуємо ці початкові та кінцеві точки для обраного періоду у пункті 2.
6. У результаті реалізації алгоритму отримуємо кінцеві точки, що відповідають максимальному прогнозованому рівню, початкові відповідають мінімальному прогнозованому рівню, та контрольні точки. Далі візуалізуємо результати.

## **Результати і обговорення**

Проведено тестування алгоритмів для прогнозування захворюваності та визначено, які працюють гірше або краще за критеріями:

1. час роботи алгоритму;
2. час обробки даних;
3. кількість точок для прогнозування;
4. точність прогнозу.

На Рис 2 можна спостерігати результати роботи алгоритму на основі критерію Вальда. Для прикладу було взято 4 країни – Україна, Німеччина, США та Великобританія.

Дані для прогнозування були взяті за період з 1980 – 2010 рік:

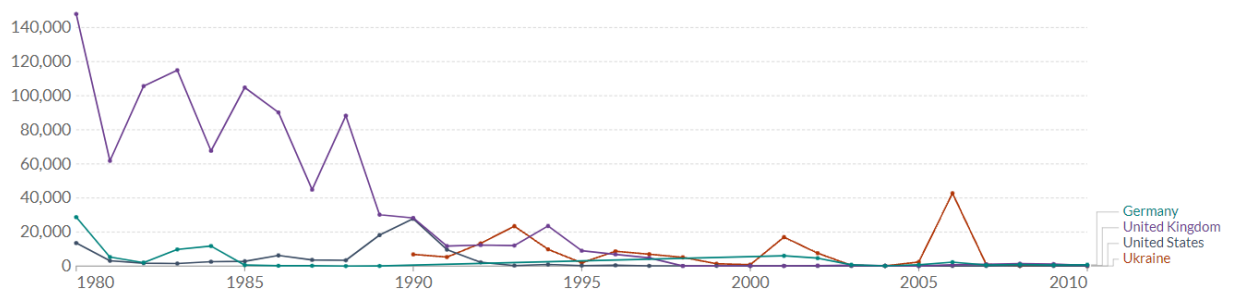


Рис 1. Дані 1980-2010 рік

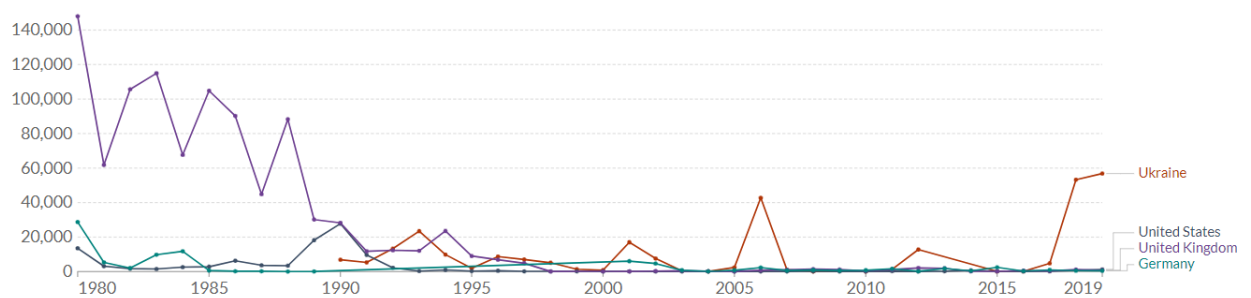


Рис 2. Результат прогнозування на 9 років (2011-2019)

За співставленням результатів прогнозування і даних затверджених Всесвітньою організацією охорони здоров'я, алгоритм показує високу точність.

## Висновки

В ході дослідження проблематики поставленої задачі було розглянуто різні алгоритми для прогнозування захворювання та було обрано методи, що були використані у розробці.

Оскільки існуючі алгоритми мають багато недоліків саме у вирішенні поставленої задачі було вирішено удосконалити алгоритм та використати метод Вальда для аналізу та прогнозування. Результати прогнозування мають достатньо високу точність. Отже, запропонований алгоритм можна використовувати для прогнозування захворюваності на кір.

У подальшому можна модифікувати алгоритм та застосовувати для прогнозування інших захворювань і також можливо враховувати інші показники для покращення точності моделювання.

## Література

1. Інфекційні хвороби : підручник / О.А. Голубовська, М.А. Андрейчин, А.В. Шкурба та ін. ; за ред. О.А. Голубовської. – 3 –є вид. – К. : ВСВ “Медицина”, 2020. – 688 с. + 12 с. кольор. вкл. ; двокольор. вид. ISBN 978-617-505-814-5
2. Langmuir, Alexander D. "Epidemiology." Chapter in Biological Foundations of Health Education Proceedings of the Eastern States Health Education Conference, April 1-2, 1948. New York, N. Y.: Columbia University Press, 1950.
3. Myers M. F., Rogers D. J., Cox J., Flahault A., Hay S. I. Forecasting Disease Risk for Increased Epidemic Preparedness in Public Health // *Advances in Parasitology*. — 2000. — Vol. 47. — P. 309–330.
4. Soebiyanto R. P., Kiang R. K. Modeling Influenza Transmission Using Environmental Parameters // *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science*. — 2010. — Vol. XXXVIII, part 8. — P. 330–334.
5. Burkom H. S., Murphy S. P., Shmueli G. Automated Time Series Forecasting for Biosurveillance // *Statistics in Medicine*. — 2007. — Vol. 26, № 22. — P. 4202–4218.



УДК 519.688

**К.т.н., доцент, ст. наук. співр., Маслянюк П.П.,  
студент Луцьк М.С.**

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

## **МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ CRM СИСТЕМИ ДЛЯ ПІДПРИЄМСТВ ТЕХНОЛОГІЧНО-БУДІВЕЛЬНОГО БІЗНЕСУ**

### **Abstract**

**Maslianko P. Pavlovych, assoc. prof., PhD; Maksym Lutsyk, student**  
*Mathematical and software part of CRM system for construction and technology corporation*

*This paper concerns the task of building CRM system for construction and technology corporations. A feature of the system is additional software for analyzing customer related information and providing suggestions on relationship management for each of those. The analysis is performed based on a combination of the following methods: abc, xyz, rfm. Usage of the developed system should provide an increase in the indicators of business efficiency.*

### **Вступ**

Будівельно-технологічна компанія – це різновид будівельної компанії, яка надає будівельні та інжинірингові послуги впродовж всіх фаз реалізації проекту. Зазвичай інжинірингові послуги – це розробка різноманітної проектної і тендерної документації, схем, проведення інженерно-геологічних досліджень.

Як і будь-який бізнес, цей тип бізнесу працює з людьми, які є його клієнтами, а клієнти – один із найважливіших активів для компанії. В умовах жорсткої конкуренції, коли ринок насичений товарами і послугами, приблизно однаковими за ціною та якістю, грамотна комунікаційна стратегія допоможе утримати клієнтів, підвищити їхню лояльність. Володіючи повними та актуальними даними по кожному замовнику, можна організувати ефективну підтримку покупця та керувати продажами. А на основі аналізу накопиченої інформації можна покращити внутрішні бізнес-процеси та визначити перспективні напрямки розвитку.

Реалізувати ці вимоги для покращення бізнесу без застосування комп'ютерних технологій досить складно, тому на допомогу керівникам і менеджерам було розроблено концепцію CRM-системи. Ця концепція

передбачає використання певного прикладного програмного забезпечення, що дозволяє автоматизувати взаємодію з клієнтами та управління внутрішніми процесами бізнесу. Історично склалося так, що галузь не надто активно використовувала програмне забезпечення CRM для управління взаємовідносинами з клієнтами, але наразі ситуація змінюється.

Програмне забезпечення CRM є ефективним та відносно новим способом для будівельної галузі для автоматизації керування проектами та взаємодії з клієнтами. Переваги стандартизації процесу продажів і розуміння всіх аспектів процесу укладання угоди можуть забезпечити швидку окупність інвестицій. Також компанії отримують конкурентну перевагу за рахунок чіткої комунікації з клієнтами і всередині компанії.

Окрім вищенаведених факторів ефективності CRM, невід'ємною частиною є аналітика, адже розуміючи, звідки беруться найприбутковіші проекти, компанія може створити найбільш ефективну стратегію щодо їх реалізації.

Щоб зрозуміти, де у будівельно-технологічної компанії найбільші можливості для збільшення доходів, CRM повинна:

- сегментувати клієнтів і персоналізувати пропозиції. Персоналізація значно впливає на ставлення клієнта до компанії. Наприклад, можна персоналізувати акційні пропозиції для цього клієнта, за групою товарів та послуг які він найчастіше використовує;
- аналізувати рентабельність — виявляти групи клієнтів, з якими компанія отримує найбільший прибуток. Виходячи з цієї інформації також можна надавати найвигідніші пропозиції для найбільш цінних клієнтів аби збільшити їх лояльність;
- відслідковувати події по клієнтам - наприклад, оцінювати об'єм покупок, щоб підключити клієнтів до програми лояльності або прогнозувати попит;
- проводити аналіз виконання плану - оцінювати його виконання в розрізі філій або співробітників, знаходити чинники впливу;
- виявляти найбільш рентабельні послуги і товари серед асортименту;
- оцінювати роботу менеджерів з продажу в різних напрямках.

### **Постановка задачі**

Метою дослідження є розробка системи управління відносинами з клієнтами для підприємств технологічно-будівельного бізнесу та відповідного програмного та математичного забезпечення для подальшого аналізу клієнтів компанії, оцінки характеру відносин з ними та надання

рекомендацій щодо покращення взаємодії на основі даних CRM системи. Інформація отримана в результаті роботи системи має допомогти покращити показники ефективності бізнесу, а саме збільшити кількість замовлень, повторних звернень та суму грошового обороту в цілому.

Предметом дослідження є представлення системи управління відносинами з клієнтами для технологічно-будівельного бізнесу, її математичні, програмні та інші види забезпечення призначені для аналізу клієнтських даних та оцінки характеру взаємовідносин з клієнтами.

Завданням є розробка системи управління відносинами з клієнтами для підприємств технологічно-будівельного бізнесу та відповідного програмного та математичного забезпечення для подальшого аналізу клієнтів компанії, оцінки характеру відносин з ними та надання рекомендацій щодо покращення цих відносин.

Для вирішення поставленого завдання використовувалися такі методи: методи теорії системного аналізу, системної інженерії, моделювання та проектування систем Data Science, методи класифікації, методи візуалізації даних, python розробки.

Вимоги до системи:

- забезпечення функціоналу для зберігання та відображення даних у спеціальних формах про: клієнтів, постачальників, робітників клієнтів та постачальників, потенціальних клієнтів, потенціальні угоди, проекти, наряди на роботу, рахунки виставлені клієнтам, робітників компанії, продукти компанії, склади продукції, договори на обслуговування.
- система повинна надавати можливість виконувати abc, хуз, rfm види аналізу на основі клієнтських даних;
- система повинна надавати можливість виконувати комплексний abc, хуз, rfm аналізу;
- в результаті виконання комплексного abc-хуз-rfm аналізу клієнту має бути присвоєний відповідний клас;

### **Термінологія**

*Customer relationship management (CRM)* – система управління відносинами з клієнтами.

*Microsoft Dynamics CRM* – пакет програмного забезпечення для управління взаємовідносинами з клієнтами, розроблений компанією Microsoft та орієнтований на організацію продажу, маркетингу та надання послуг.

## Компонентна модель CRM системи

Архітектура системи має відповідати поставленим вимогам та мати необхідні властивості аби всі необхідні дані могли зберігатися та оброблятися в ній. Тобто система має включати в себе усі сутності та зв'язки між ними які задіяні в бізнес-процесах компанії.

Систему пропонується запроваджувати шляхом кастомізації вже існуючого продукту Microsoft Dynamics CRM. Під кастомізацією системи розуміється можливість налаштування об'єднаних в один контейнер параметрів різних компонентів системи. Розробка та налаштування функціонування системи здійснюється через графічний інтерфейс у внутрішньому редакторі системи. Редактор надає можливість створювати сутності та їх поля, налаштовувати набори правил функціонування системи, створювати форми для роботи користувача з сутностями системи та уявлення[1].

Таке рішення було прийнято шляхом порівняння існуючих інструментів та методів розв'язку поставленої задачі. Саме MS Dynamics CRM було обрано через гнучкість конфігурації яка дозволяє забезпечити виконання максимуму налаштувань з мінімальним зміною коду та в найкоротші терміни [1]. Також система підтримує велику кількість інтеграцій, що забезпечує подальші шляхи для розвитку.

Компонентна модель системи управління відносинами з клієнтами що реалізує вказані функції зображена на рисунку 1.

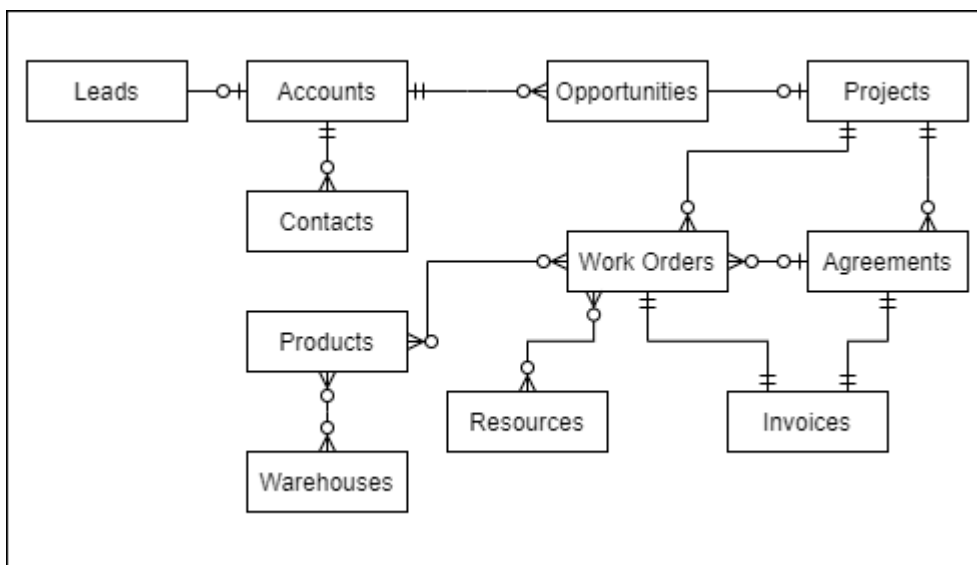


Рисунок 1 - Модель системи управління відносинами з клієнтами для компанії технологічно-будівельного бізнесу. Діаграма компонентів в нотатції UML

В системі передбачені такі сутності:

- accounts, яка зберігає данні про замовників, їх будівельні майданчики та постачальників;
- contacts, яка зберігає дані про робітників клієнтів;
- leads, яка зберігає дані про потенціальних клієнтів;
- opportunities, яка зберігає дані про потенціальні угоди;
- projects, яка зберігає дані про проекти компанії;
- work orders, яка зберігає дані про робочі наряди компанії;
- agreements, яка зберігає дані про договори на обслуговування;
- invoices, яка зберігає дані про рахунки на оплату;
- resources, яка зберігає дані про робітників компанії;
- products, яка зберігає дані про продукти які використовує компанія;
- warehouses, яка зберігає дані про склади продукції та наявні продукти на них.

Запропонована модель системи формалізована з використанням UML нотатії, що дозволяє формалізувати взаємозв'язок компонентів.

### **Математичні моделі реалізації abc, xyz, rfm аналізів**

ABC-аналіз - в розрізі аналізу клієнтської бази це метод, який дозволяє створити рейтинг клієнтів за сумою їх покупок та виявити ту категорію клієнтів, яка приносить компанії максимальну виручку. Тобто цей метод дозволяє виявити найбільш важливих клієнтів для компанії [2].

В основі ABC-аналізу лежить принцип Парето, який свідчить про те, що 20% клієнтів компанії приносять 80% всього товарообігу. Основна мета розглядуваного методу – класифікація клієнтів на три групи:

- А – найцінніші (20% клієнтів забезпечують 80% продажів);
- В – проміжні (30% клієнтів забезпечують 15% продажів);
- С – найменш цінні (50% клієнтів забезпечують 5% продажів).

XYZ-аналіз - в розрізі аналізу клієнтської бази це метод класифікації клієнтів компанії за трьома групами на основі мінливості (стійкості) їх попиту [2]:

- X – 0-10% (коефіцієнт варіації) – клієнти із найстійкішим попитом. Ця група характеризується високим ступенем прогнозованості;
- Y – 10-25% – клієнти з мінливим обсягом покупок. Ця група характеризується сезонними коливаннями та середніми можливостями прогнозування;
- Z – від 25% – клієнти з випадковим попитом. Це клієнти з нерегулярним споживанням та непередбачуваними коливаннями попиту. Спрогнозувати їхній попит практично неможливо.

RFM-аналіз – є методом сегментації клієнтів за лояльністю. Його алгоритм побудований на застосуванні таких трьох показників [2]:

- R (Recency) – час з дня останньої здійсненої покупки;

- F (Frequency) – кількість здійснених клієнтом покупок за період часу;
  - M (Monetary) – витрачені кошти клієнта протягом періоду часу.
- За підсумками RFM-аналізу клієнтів вирішено розбити на 3 групи:
- R – найлояльніші клієнти;
  - F – клієнти с середньою лояльністю;
  - M – клієнти з найнижчою лояльністю.

### Математична модель комбінованого abc-xyz-rfm аналізу

Запропонований метод комбінування трьох розглядуваних методів аналізу полягає в поєднанні їх використанні. В результаті виконання кожного з аналізів клієнту присвоюється один з трьох класів, що в результаті проведення усіх видів аналізу присвоює клієнту один з 27 класів. Такий варіант класифікації дозволяє максимально вдало охарактеризувати клієнта, аби зробити висновки щодо подальшої взаємодії з ним. На рисунку 2 зображено узагальнений алгоритм комбінованого abc-xyz-rfm аналізу.

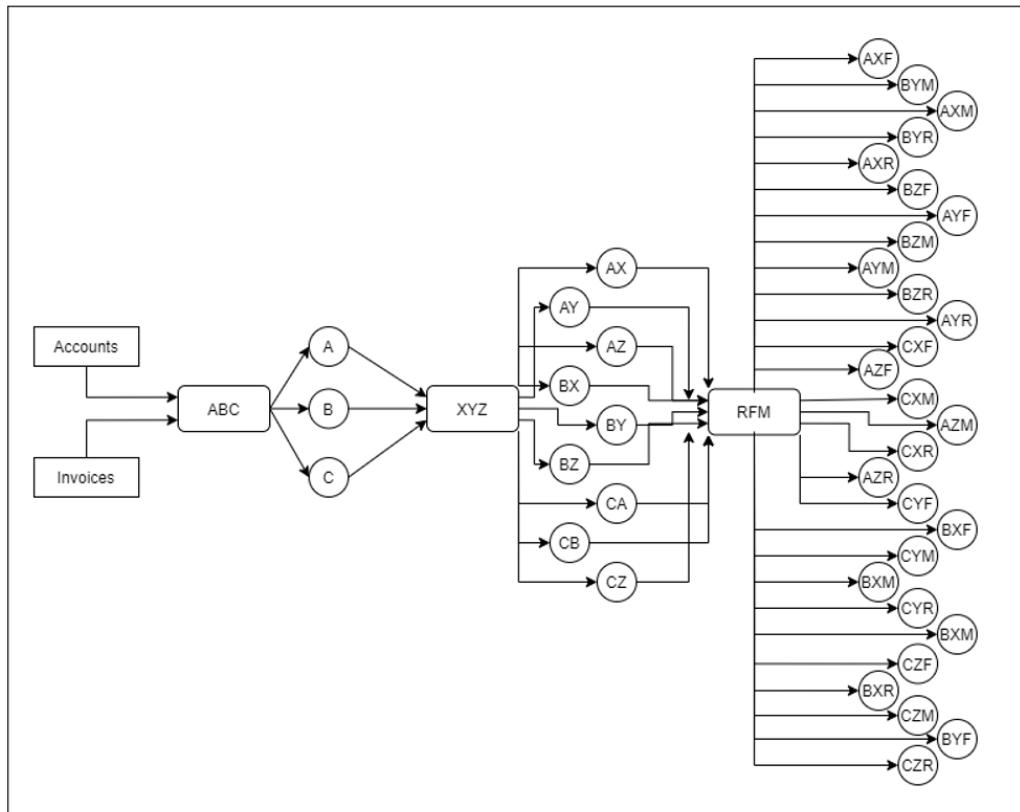


Рисунок 2 - узагальнений алгоритм комбінованого abc-xyz-rfm аналізу

Система класів зображена у таблиці 1.

Таблиця 1

## Класи поділу клієнтів для комбінованого abc-xyz-rfm аналізу

Клас	Характеристика класу
AXF	Найцінніші клієнти зі стійким попитом проте середньою лояльністю
AXM	Найцінніші клієнти зі стійким попитом але низькою лояльністю
AXR	Найцінніші клієнти зі стійким попитом і високою лояльністю
AUF	Найцінніші клієнти з мінливим попитом і середньою лояльністю
AUM	Найцінніші клієнти з мінливим попитом і низькою лояльністю
AUR	Найцінніші клієнти з мінливим попитом і високою лояльністю
AZF	Найцінніші клієнти з випадковим попитом і середньою лояльністю
AZM	Найцінніші клієнти з випадковим попитом і низькою лояльністю
AZR	Найцінніші клієнти з випадковим попитом і високою лояльністю
BXF	Клієнти середньої цінності з найстійкішим попитом і середньою лояльністю
BXM	Клієнти середньої цінності з найстійкішим попитом і низькою лояльністю
BXR	Клієнти середньої цінності з найстійкішим попитом і високою лояльністю
BUF	Клієнти середньої цінності з мінливим попитом і середньою лояльністю
BUM	Клієнти середньої цінності з мінливим попитом і низькою лояльністю
BUR	Клієнти середньої цінності з мінливим попитом і високою лояльністю
BZF	Клієнти середньої цінності з випадковим попитом і середньою лояльністю
BZM	Клієнти середньої цінності з випадковим попитом і низькою лояльністю
BZR	Клієнти середньої цінності з випадковим попитом і високою лояльністю
CXF	Найменш цінні клієнти з стійким попитом і середньою лояльністю
CXM	Найменш цінні клієнти з стійким попитом і низькою лояльністю
CXR	Найменш цінні клієнти з стійким попитом і високою лояльністю
CUF	Найменш цінні клієнти з мінливим попитом і середньою лояльністю

CYM	Найменш цінні клієнти з мінливим попитом і низькою лояльністю
CYR	Найменш цінні клієнти з мінливим попитом і середньою лояльністю
CZF	Найменш цінні клієнти з випадковим попитом і середньою лояльністю
CZM	Найменш цінні клієнти з випадковим попитом і низькою лояльністю
CZR	Найменш цінні клієнти з випадковим попитом і високою лояльністю

### **Валідація**

Найбільш точним методом валідації розробленої системи є її застосування у реальній компанії і порівняння показників ефективності цієї компанії до і після впровадження системи. Проте вже на етапі завантаження до CRM системи тестових даних, вона ідентифікувала дублікати серед записів контактів клієнтів, та, після погодження з користувачем, об'єднала їх. Також в результаті використання комбінованого abc-xyz-rfm аналізу система надала звіт, в якому знаходиться інформація про належність клієнтів до певного класу. На рисунку 3 зображено частину звіту з інформацією про клієнтів класу AXF.

### **Висновки**

Було досліджено методи розробки системи управління відносинами з клієнтами для підприємств технологічно-будівельного бізнесу і програмного та математичного забезпечення для аналізу клієнтських даних компанії, оцінки характеру відносин з клієнтами та надання рекомендацій щодо покращення взаємодії з ними.

Встановлено, що найбільш доцільно використовувати вже готову CRM систему Microsoft Dynamics CRM та налаштувати її відповідно до потреб і бізнес-процесів компанії. Також встановлено, що для аналітики найкраще використовувати комбінацію існуючих методів та на основі результатів отриманих внаслідок використання цих методів автоматизувати рекомендації по взаємодії з клієнтом.

Запропоновано компонентну модель реалізації системи управління відносинами з клієнтами та модель підсистеми аналізу супутніх даних. Система призначена для покращення показників ефективності бізнесу, а саме збільшення кількості замовлень, повторних звернень, суми замовлень та грошового обороту в цілому.

Заплановано вдосконалення системи шляхом впровадження інтеграції з сервісом зберігання даних Sharepoint, для розширення



функціональності. Так як будівельний бізнес завжди тісно пов'язаний з високим документообігом, подібний функціонал є критично важливим в системі.

В подальшому планується запровадження інтеграції з системою автоматизації калькуляції податків ССН SureTax, що значно полегшить роботу компанії та значно розширить функціонал розробленої системи.

	D	E	F
1	Name	City	CustomerID
2	Hexpol	Burton	00002226
3	Hickman Family Farms	Buckeye	00007847
4	Highland Builders	Dallas	00008503
5	Highland Park United Methodist	Dallas	00001517
6	Highpoint Urban Living	Fort Worth	00002060
7	Highridge Development Company, LLC	North Dartmouth	00002213
8	Hill & Wilkinson LTD (Parent)	Richardson	00001219
9	Hillside Christian Church   PO Box 7807	Amarillo	00006936
10	Hillwood Construction Services (Parent)	Dallas	00007247
11	Hillwood Properties (Parent)	Fort Worth	00006666
12	Hilton - Arlington	Arlington	00006204
13	Hilton (Parent)	San Antonio	00001425
14	Hilton Americas-Houston	Houston	00008202
15	HILTON DALLAS PARK CITIES	DALLAS	00006382
16	Hilton Garden Inn	Tempe	00002448
17	Hilton Granite Hotel   5805 Granite Pkwy	Plano	00001425
18	Hilton Inn   2929 West Frye	Chandler	00007791
19	Hines Interests LTD Partners (Parent)	Houston	00008139
20	HITT Contracting Inc (Parent)	Falls Church	00001494
21	Hixson Lumber Sales Inc	Dallas	00001682
22	HL Champion Holdings Company, LP (Parent)	Merrifield	00002124
23	HMG Bee Cave Healthcare, LLC	Bee Cave	00006525
24	HMH West MOB 3 & Parking Garage B	Houston	00008287
25	Hoar Construction (Parent)	Houston	00009101
26	Hobby Lobby - Billing	Oklahoma	00001598
27	Hoffbrau Steaks Headquarters	Dallas	00002011
28	Hogan Lovells US LLP   609 Main	Houston	00001841
29	H-O-H Water Technology, Inc.	PATLATINE	00006433

Рис. 3. Звіт отриманий в результаті комбінованого abc-xyz-rfm аналізу

## Література

1. Харитонов В. І. Застосування CRM-систем при ухваленні управлінських рішень в організації. Системне керування. 2016 р. - С. 55-59.

2. Лукінський В.С., Цвірінько І.А. Прикладні методи та моделі в логістиці // Логістика сьогодні. - 2004.- № 6. - С. 30-38.
3. Зенкова Ж.М. Логістичний підхід в управлінні підприємством: навчально-методичний комплекс / Ж. Н. Зенкова; Том. держ. ун-т, [Ін-т дистанційної освіти]. - Томськ: [ІДО ТГУ], 2012. URL: <http://vital.lib.tsu.ru/vital/access/manager/Repository/vtls:000460937>
4. Афанасьєв, В.Н. Інформаційні технології в управлінні підприємством / В.М. Афанасьєв, А.І. Постніков. — 2 вид., — М.: МІЕМ. — 2016. — 412 с.
5. Вільхінська О.В. Інноваційний розвиток підприємств на основі технологій електронного бізнесу / Вільхінська О.В. – 2018. – 156-166с.

УДК 519.688

Доцент Маслянюк П.П., студент Маляренко І.С.,  
асистент Ковальчук-Хіміук Л.О.

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

## МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ІДЕНТИФІКАЦІЇ ОЦІНОЧНИХ СУДЖЕНЬ У ТЕКСТАХ

### Abstract

**Pavlo Maslianko, docent; Illia Maliarenko, student;  
Kovalchuk-Khymiuk, assistant.**

*Mathematical and software systems for the identification of value judgments in texts*

*This paper concerns the task of classification in product reviews. The existed classification services and methods is studied and discussed. The new architecture for predictive classification system is proposed and described. The analysis of system efficiency and accuracy is presented. The ways for further research and upgrades are proposed as well.*

### Вступ

На сьогодні, дані, що характеризують активності та відгуки користувачів, відіграють суттєву роль у розумінні та побудові бізнес-процесів організаційної системи. Так як більша частина даних збережена у форматі тексту з використанням природньої мови, вони потребують додаткової обробки, через труднощі або неможливість чисельного представлення такого типу даних.

Задача ідентифікації оціночних суджень у відгуках користувачів є однією з важливих задач обробки природньої мови. Серед методів ідентифікації слід виділити метод класифікації природньої мови, що дозволяє визначити приналежність даних до певного класу оцінювання якості товарів та послуг. Основними методами в аналізі природньої мови для подальшої їх класифікації та передбачення є методи машинного навчання, серед яких найбільш точні результати можна отримати з використанням нейронних мереж.

У статті запропоновано рішення на основі нейронних мереж, що дозволяє суттєво підвищити точність класифікації та відрізняє його від традиційних рішень на основі методу опорних векторів.

## **Постановка задачі**

Метою дослідження є класифікація та ідентифікація суджень користувачів у відгуках на ігри з подальшою їх аналітичною оцінкою.

Предметом дослідження є математичне та програмне забезпечення системи ідентифікації оціночних суджень у відгуках на товар на основі нейронних мереж та попереднього ранжування даних.

Задачею є розробка спеціалізованої системи із використанням математичного та програмного забезпечення для подальшого аналізу, передбачення та відображення текстових даних у чисельному представленні.

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

Вимоги до системи:

- система визначає клас відгуку, у проміжку від 0 до 10 включно;
- система надає можливість автоперекладу для мови, що не підтримується;
- ідентифікація проводиться на основі відгуків на відеоігри;
- система може бути розширена для використання на відгуках щодо інших типів товарів;
- система надає більшу точність, аніж наявні аналоги;
- доступ до системи надається через web-інтерфейс;

## **Термінологія**

*Bidirectional Encoder Representations from Transformers (BERT)* – модель мови представлена працівниками компанії Google, що може застосовуватися для побудови векторного представлення слів [1].

*Модель ідентифікації оціночних суджень (МІОС)* – модель нейронної мережі побудована для класифікації текстів.

## **Компонентна модель системи ідентифікації оціночних суджень**

Архітектура системи має відповідати поставленим вимогам та мати необхідні властивості для забезпечення ідентифікації оціночних суджень у текстах. Задачею системи є отримання чисельного представлення текстів згідно передбачених класів.

Компоненти архітектури системи повинні забезпечити виконання таких основних функцій:

- групування та сортування даних зібраних в процесі роботи чи від користувача;
- завантаження нових текстових даних до сховища даних;

- класифікація нових текстових даних;
- фільтрування надлишкової інформації та попередня обробка текстів;
- визначення мови написання тексту та переклад текстів на англійську;
- тренування, ініціювання та валідацію нейронної мережі класифікатору;
- надання web-інтерфейсу для користувача.

Компонентна модель системи ідентифікації оціночних суджень, що реалізує вказані функції складається з таких компонентів (рис.1): сховище навчальних даних, обробка текстових даних, навчання МІОС, збір текстів, ідентифікатор та web-застосунок.

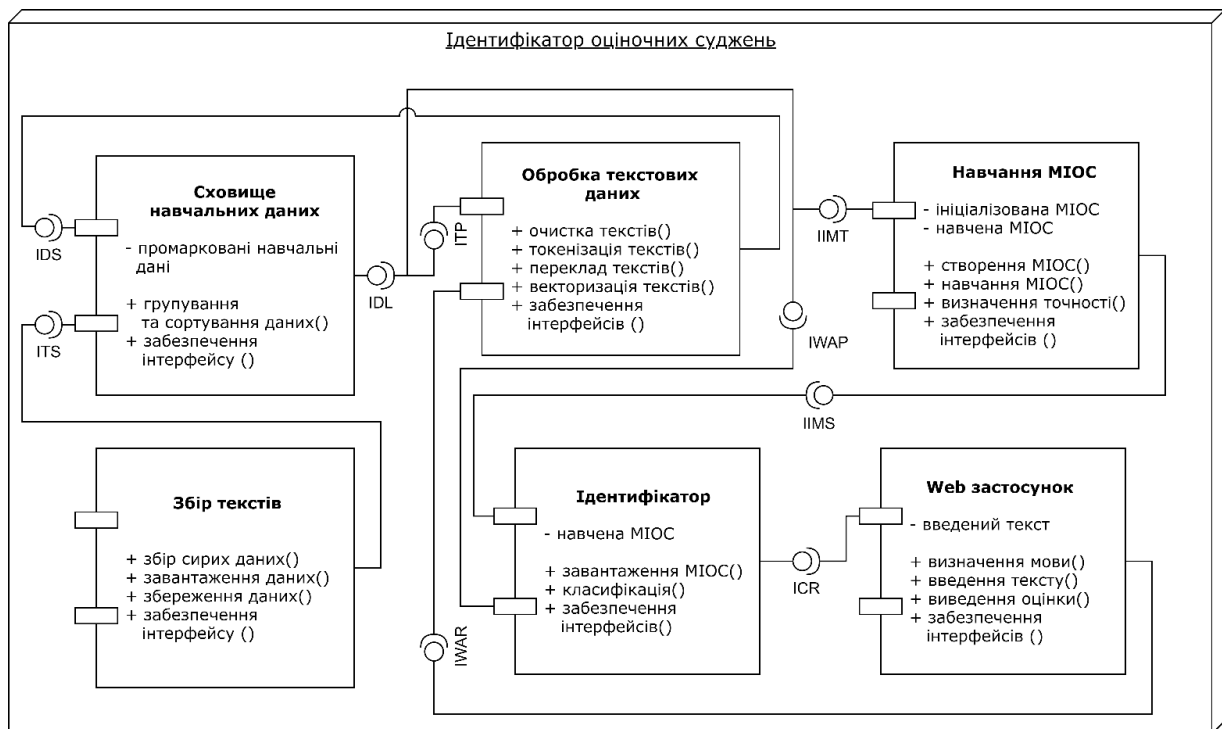


Рис. 1. Модель системи ідентифікації оціночних суджень.  
Діаграма компонентів в нотації UML

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

Перелік інтерфейсів системи:

- IDS (Interface Database Save) — інтерфейс для передачі даних у базу даних, реалізований із використанням драйверу бази даних;
- ITS (Interface Text Save) — інтерфейс завантаження даних з набору даних до бази даних, реалізований із використанням драйверу бази даних;
- IDL (Interface Database Load) — інтерфейс для завантаження даних з бази даних, реалізований із використанням драйверу бази даних;

- ITP (Interface Text Processing) — інтерфейс передачі даних у процес початкової обробки даних;
- ИИМТ (Interface Identification Model Training) — інтерфейс передачі векторизованих та оброблених даних для подальшого навчання МІОС.
- ИИМС (Interface Identification Model Save) — інтерфейс передачі навченої моделі МІОС для використання в задачах класифікації.
- ИИВАР (Interface Web Application Raw) — інтерфейс передачі текстових даних, отриманих від користувача з використанням web-інтерфейсу.
- ИИВР (Interface Classification Result) — інтерфейс передачі результату класифікації навченої МІОС на основі користувацького тексту.
- ИИВАРП (Interface Web Application Processed) — інтерфейс для передачі результатів роботи нейронної мережі класифікатору до web-інтерфейсу.

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

На рис. 2 приведена модель системи ідентифікації оціночних суджень та деталізована діаграма процесів другого рівня з уточненням кожного рівня в нотації UML.

Нижче наведено опис двох головних компонентів системи — «Навчання МІОС» та «Обробка текстових даних», оскільки решта компонентів тривіальні.

### **Математичні моделі реалізації компоненту «Навчання МІОС»**

В якості класифікатору було обрано нейронну мережу з використанням архітектури LSTM [2]. Ця архітектура була обрана на основі детального порівняння наявних теоретичних методів, із використанням літератури [3, 4] та власних досліджень.

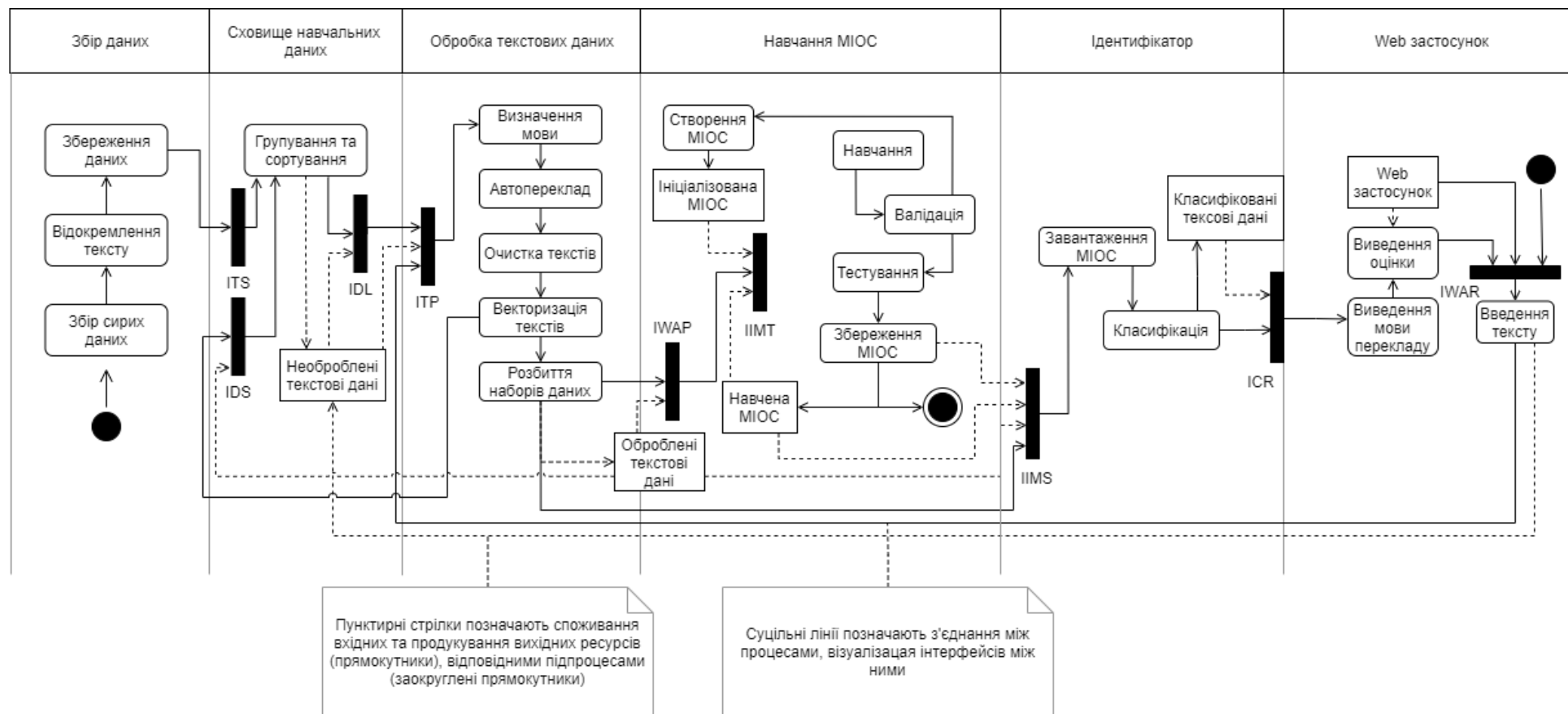


Рис. 2. Модель системи ідентифікації оціночних суджень.  
 Деталізована діаграма процесів другого рівня з уточненням кожного рівня, в нотатції UML

Ця архітектура відрізняється від інших можливістю запам'ятовувати попередні дані та використовувати їх у поточному проході через нейрон. Вона є покращенням архітектури рекурентної нейронної мережі [5].

### **Математичні моделі реалізації компоненту «Обробка текстових даних»**

Для побудови векторного представлення слів використовується BERT, що представляє собою глибоку нейронну мережу. Данна нейронна мережа представлена у декількох варіантах у залежності від розміру та кількості параметрів. В представленій архітектурі системи використовується велика версія на 24 блоків-трансформерів, та 1024 прихованих нейронів у кожному шарі з загальною кількістю параметрів 340 мільйонів [1]. Для побудови векторного представлення текстів, беруться ваги нейронів з чотирьох останніх шарів мережі, що з'єднуються в один вектор. Використання декількох шарів зумовлено тим, що кожен шар відповідає за певний набір ознак тексту, тому для збереження якомога більшої кількості даних, використовується саме цей підхід.

### **Валідація**

Для валідації побудованої моделі нейронної мережі використовуються показники точності. Навчання велося на текстах, розділених за оцінками за 10-бальною шкалою. Оскільки оцінювання та точне передбачення для текстів написаних природньою мовою неможливе, доцільно ввести допустиму похибку.

Допустима похибка — числовий показник, що визначає наскільки може відрізнятись передбачений клас від очікуваного — за абсолютною величиною — щоб вважатися за правильну відповідь.

Для кожного з класів, щоб визначити точність передбачення, було пораховано процентне співвідношення передбачених класів, що відрізняються від очікуваного на абсолютну величину меншу за допустиму похибку, до загальної кількості відгуків, що мають певний клас.

Залежність точності від допустимої похибки вказано у таблиці 1. Проілюстровано розподіл точності (рис 3) при різних значеннях похибки, та загальний розподіл передбачених класів відносно очікуваних (рис. 4).

Щільність — числовий показник, що визначає частоту певної точності серед усіх класів.

Визначено, що в середньому передбачений клас відрізняється від очікуваного на 1.475 за абсолютною величиною, а при допустимій похибці у 2 бали, точність складає 75,535%, що є гарним результатом.



Розподіл передбачування

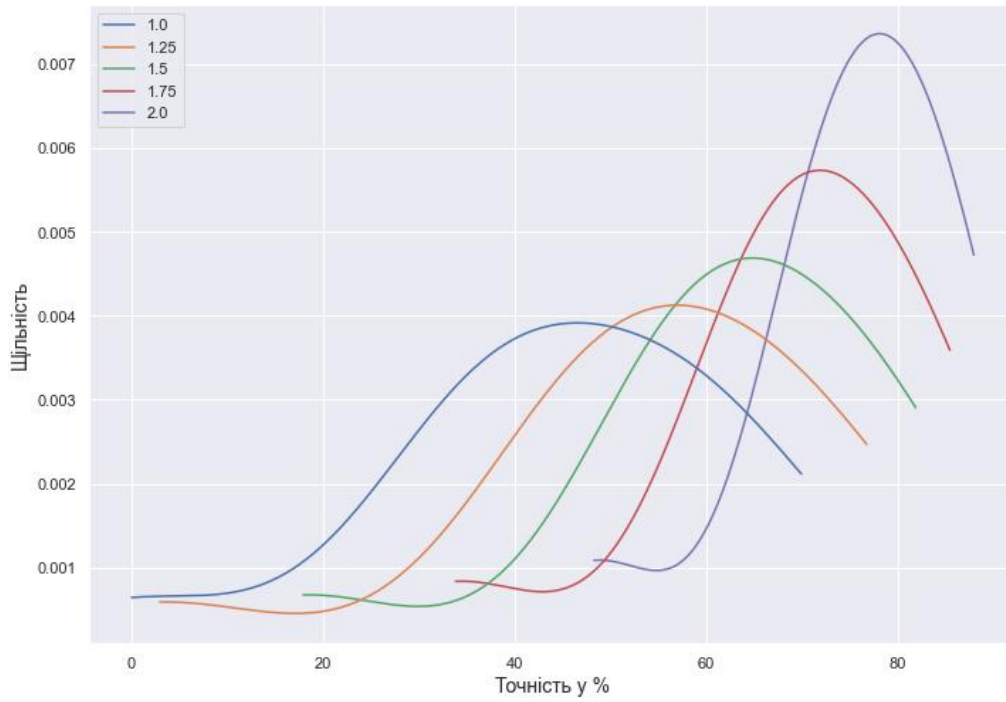


Рис. 3. Точність передбачень роботи системи, в залежності від допустимої похибки

Розподіл передбачування

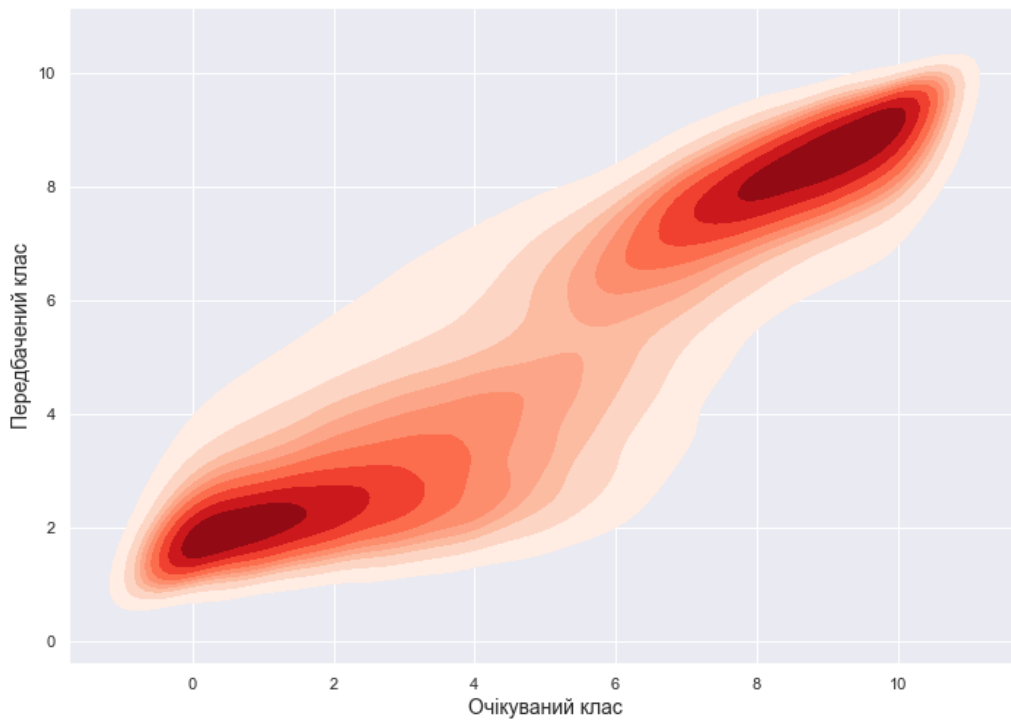


Рис. 4. Розподіл передбачень за результатами роботи системи

Таблиця 1

Точність системи в залежності від класу та похибки у відсотках

Точність (%)		Допустима похибка					
		1.0	1.25	1.5	1.75	2.0	2.25
Очікуваний клас	0	0	2,949	17,91	33,825	48,265	61,232
	1	31,697	46,734	57,913	64,796	70,758	74,923
	2	56,042	62,189	67,116	72,295	76,884	80,547
	3	54,875	66,399	74,799	80,287	83,833	86,028
	4	42,9	50,866	58,615	68,788	77,835	84,242
	5	34,076	43,076	51,464	59,589	68,021	74,705
	6	39,752	48,935	57,941	65,927	72,094	77,329
	7	55,308	65,711	72,988	77,997	82,149	85,06
	8	69,915	76,737	81,864	85,424	87,924	89,364
	9	63,832	72,124	78,765	83,037	85,956	87,775
	10	38,922	51,783	61,418	69,397	76,061	81,07

## Висновки

1. Було досліджено наявні застосунки та теоретичні методи ідентифікації оціночних суджень і встановлено, що є необхідність ввести допустиму похибку від 1 до 2 балів для передбаченого класу відгуків, тому що неможливо точно передбачити клас тексту написаного людиною.
2. Встановлено, що у сервісах оцінювання ігор найчастіше використовують позитивні відгуки, тому є необхідність у додаванні більшої кількості негативних відгуків у навчальну вибірку, та видаленні зайвих позитивних.
3. Запропоновано компонентну модель реалізації системи ідентифікації оціночних суджень з метою класифікації та ідентифікації суджень у текстах за числовою характеристикою.
4. Заплановано вдосконалення методу початкової обробки даних та нейронної мережі класифікатора, для досягнення ще більш релевантних результатів.
5. В подальшому планується використання таких метрик як: precision, recall, F1-Score, ROC-крива та показник AUC, з використанням методу перехресної крос-валідації.

## Література

1. J. Devlin, M.-W. Chang, K. Lee, і K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2019.

2. Hochreiter S., Schmidhuber J. Long Short-Term Memory. *Neural Computation*. 1997. Vol. 9. P. 1735-1780.
3. Moraes R., Valiati J., Gavião N. W. Document-level sentiment classification: An empirical comparison between SVM and ANN. *Expert Systems with Applications*. 2013. Vol. 40. P. 621-633.
4. Yin W., Kann K., Yu M. Comparative Study of CNN and RNN for Natural Language Processing. *CoRR*. 2017. 7 p.
5. Du K. L., Swamy M. N. Recurrent Neural Networks. 2014. P. 337-353.

Д-р фіз.-мат. наук, професор Норкін В.І., студент Козирєв А.Ю.

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

## ДОСЛІДЖЕННЯ ЗБІЖНОСТІ СТОХАСТИЧНИХ ГРАДІЄНТНИХ ТА КІНЦЕВЕ-РІЗНИЦЕВИХ МЕТОДІВ ОПТИМІЗАЦІЇ

### Abstract

**Volodymyr I. Norkin, Dr. Sc., prof.; Anton Y. Kozyriev, master's student**  
*Investigation of convergence of stochastic gradient and finite-difference  
optimization methods*

*Stochastic gradient algorithms have become one of the most popular numerical optimization methods due to their efficiency and formulation simplicity. In general, the convergence rate indicates the performance of the particular stochastic optimization method. The article investigates the inquired metric of stochastic finite-difference optimization methods on convex functions and highlights that it is not an entirely adequate measure of the optimization algorithm's quality in general.*

### Вступ

На сьогоднішній день стохастичні градієнтні алгоритми стали одними з найбільш популярних чисельних методів стохастичної локальної оптимізації в силу своєї ефективності та простоти формулювання. Від того наскільки ефективним є обраний алгоритм градієнтного спуску залежить обсяг витрачених ресурсів обчислювальної системи, обсяг витраченого часу для вирішення задачі, точність розв'язку або отриманої моделі тощо. Тому актуальною є тематика дослідження адаптивних стохастичних градієнтних методів, а також розробка нових та вдосконалення існуючих методів.

### Постановка задачі

Метою статті є розробка принципових схем стохастичних кінцево-різницевих адаптивних методів розв'язання задач стохастичного програмування та теоретичне і експериментальне дослідження швидкості збіжності методів. Значення швидкості збіжності для певного стохастичного методу оптимізації є показником того, наскільки алгоритм є ефективним у вирішенні задачі при порівнянні з іншими.

Задача стохастичної оптимізації має вигляд [1]:

$$\min_{x \in X} [f(x) = \mathbb{E}F(x, \xi) = \int_{\xi \in \Xi} F(x, \xi) P(d\xi)], X \subseteq \mathbb{R}^n \quad (1)$$

де  $f : X \rightarrow \mathbb{R}^n$  цільова функція задачі оптимізації,  $X \subseteq \mathbb{R}^n$  допустима множина змінних задачі,  $F : X \times \Xi \rightarrow \mathbb{R}^1$  опукла або диференційована функція, що залежить від детермінованої змінної  $x \in X$  та випадкової величини  $\xi \in \Xi$ , визначеної на деякому ймовірнісному просторі  $(\Xi, \Sigma, P)$ ,  $\mathbb{E}$  - знак математичного сподівання. Складність задачі (1) полягає у тому, що не завжди можливо обчислити інтеграл (математичне сподівання) у (1) в явному вигляді, а тим більш градієнт цього інтегралу. Ця складність долається шляхом використання так званих стохастичних градієнтних методів, в яких на кожній ітерації використовуються градієнти  $\nabla_x F(x, \xi)$  випадкової підінтегральної функції  $F(\cdot, \xi)$  або їх кінцеве-різницеві аналоги.

### Огляд стохастичних градієнтних методів

Якщо можливо обчислити значення функції  $f(x)$  та її градієнт  $\nabla f(x)$ , то задачу (1) можна розв'язувати детермінованими методами нелінійного програмування. Зокрема метод градієнтного спуску [1, 2] для розв'язання задачі (1) має вид:

$$x^{k+1} = \Pi_x(x^k - \lambda_k \nabla f(x^k)), x^0 \in X, k \in \mathbb{N}, \Pi_x(y) = \arg \min_{x \in X} \|y - x\|$$

де  $k$  позначає номер ітерації методу,  $\nabla f(x^k)$  - градієнт цільової функції  $f(x)$  в точці  $x = x^k$ ,  $\Pi_X$  - оператор проектування на допустиму множину  $X$ .

Відповідний метод стохастичного градієнтного спуску задається наступним рекурентним співвідношенням [1, 2]:

$$x^{k+1} = \Pi_x(x^k - \lambda_k u^k), x^0 \in X, k = 0, 1, \dots \quad (2)$$

де  $u^k = \nabla_x F(x^k, \xi^k)$  - градієнт по змінній  $x$  підінтегральної функції  $F(\cdot, \xi^k)$  при фіксованому значенні параметра  $\xi = \xi^k$ ,  $\lambda_k$  - невід'ємні крокові множники,  $\{\xi^k\}$  - незалежні спостереження (статистика) випадкової величини  $\xi$ . Для гарантування збіжності методу (2) крокові множники  $\lambda_k \geq 0$  повинні задовольняти умовам:

$$\sum_{k=0}^{\infty} \lambda_k = +\infty, \sum_{k=0}^{\infty} \lambda_k^2 < +\infty$$

**Визначення.** Випадковий вектор  $u^k$  називається *стохастичним градієнтом* функції  $f(x)$  в точці  $x = x^k$ , якщо для умовного стохастичного

сподівання  $\mathbb{E}\{u^k|x^k\}$  виконан  $\mathbb{E}\{u^k|x^k\} = \nabla f(x^k)$ . Таким чином, якщо  $\nabla f(x) = \mathbb{E}\nabla_x F(x, \xi)$ , то вектор  $u^k = \nabla_x F(x^k, \xi^k)$  є стохастичним градієнтом функції  $f(x)$  в точці  $x = x^k$ .

У пакетному варіанті [2, 3] стохастичного градієнтного спуску (2) використовується стохастичний градієнт виду  $u^k = \frac{1}{N_k} \sum_{i=1}^{N_k} \nabla_x F(x^k, \xi_i^k)$ , де  $\{\xi_i^k, i = 1, \dots, N_k\}$  - незалежні спостереження випадкової величини  $\xi$  на ітерації  $k$ . У якості стохастичних (квазі) градієнтів можуть використовуватися *стохастичні кінцеве-різницеві оцінки* [4, 5, 6]:

$$u^k = \frac{1}{N_k} \sum_{i=1}^{N_k} \frac{1}{2h_k} (f(x^k + h_k \tilde{y}_k^i) - f(x^k - h_k \tilde{y}_k^i)) \tilde{y}_k^i$$

градієнту цільової функції  $f(\cdot)$  в (1) по випадковим напрямом  $\tilde{y}_k^i$ , рівномірно розподіленим на одиничній сфері. Варіант методу (2) с *нормуванням стохастичних градієнтів* (або с *адаптивним регулюванням кроків* [2]) має вигляд:

$$x^{k+1} = \Pi_x(x^k - \lambda_k H^k u^k), x^0 \in X, k = 0, 1, \dots$$

де  $H^k$  - деяка квадратна діагональна матриця з невід'ємними коефіцієнтами на діагоналі, що залежать від поточної ітерації  $x^k$  або від всієї траєкторії  $\{x^0, x^1, \dots, x^k\}$ . Наприклад, діагональні елементи можуть бути  $h_{ii}^k = \frac{1}{\|u^k\| + \varepsilon}, \varepsilon > 0$ .

В методі *дзеркального спуску Неміровського-Юдіна* [1] ітераційна послідовність (2) додатково осереднюється наступним чином (див. теорему 2):

$$\bar{x}^{k+1} = \frac{\sum_{i=1}^k \lambda_i x^i}{\sum_{i=1}^k \lambda_i} = (1 - \sigma_k) \bar{x}^k + \sigma_k x^k, \sigma_k = \frac{\lambda_k}{\sum_{i=0}^k \lambda_i}$$

*Метод імпульсу* (або *важкої кулі*) для розв'язання задачі (1) з  $X \in \mathbb{R}^n$  має наступний загальний вид [2, 7]:

$$x^{k+1} = x^k + \gamma_k (x^k - x^{k-1}) - \lambda_k \nabla f(x^k) \quad (3)$$

де  $\gamma_k > 0, \lambda_k > 0$  - інерційний та кроковий параметри методу, що можуть залежать від  $x^k$  або  $\{x^0, \dots, x^k\}$ . Тут замість детермінованих градієнтів  $\nabla f(x^k)$  можуть використовуватися стохастичні градієнти  $u^k$  функції  $f(x)$  в точці  $x = x^k$ . Урахування показника імпульсу дозволяє прискорити швидкість збіжності алгоритму градієнтного спуску, проте за рахунок акумулювання даного показника алгоритм може "проскочити" точку мінімуму та може витратити додаткові ітерації для згасання значення цього показника.

Метод Нестерова (або метод яружного кроку) [7] задається наступними рекурентними співвідношеннями:

$$y^k = x^k - \rho_k \nabla f(x^k), x^{k+1} = y^k + \lambda_k (y^k - y^{k-1}), x^0 = y^0 \in \mathbb{R}^n, k = 0, 1, \dots \quad (4)$$

де  $\rho_k > 0$  крок градієнтного методу,  $\lambda_k > 0$  - яружний крок. Тут також замість детермінованих градієнтів  $\nabla f(x^k)$  можуть використовуватися стохастичні градієнти функції  $f(x)$ . Перше рівняння методу означає спуск із точки  $y^k$  на дно  $x^k$  яру функції  $f(x)$ , а друге рівняння означає крок вздовж яру від точці  $x^k$  в напрямку  $(x^k - x^{k-1})$ .

Для перелічених методів є актуальною проблема збіжності на розріджених даних (проблема зникаючих градієнтів). Рішенням є адаптація кроку градієнтного спуску  $\lambda_k$  до значень з статистичної вибірки на певній ітерації, виконуючи більші оновлення параметрів  $x^k$  для розріджених значень вхідних параметрів  $\xi^k$  та малі оновлення для частих значень вхідних параметрів [9]. Нехай  $G_k$  - лінійна комбінація зі значень складової на попередній ітерації та поелементного добутку градієнта на поточній ітерації  $G_k = G_{k-1} + \|\nabla F(x^k, \xi^k)\|^2$ , а  $\varepsilon > 0$  - згладжуючий доданок. Введемо значення адаптивного кроку  $\bar{\lambda}_k = \frac{\lambda_k}{\sqrt{G_k + \varepsilon}}$  у співвідношення (2) та отримаємо ітераційну послідовність методу адаптивного градієнту (AdaGrad) [8]:

$$x^{k+1} = x^k - \frac{\lambda_k}{\sqrt{G_k + \varepsilon}} \circ \nabla_x F(x^k, \xi^k), x^0 = y^0 \in \mathbb{R}^n, k = 0, 1, \dots \quad (5)$$

Метод має кращу збіжність на розрідженій вибірці, проте сума квадратів значень градієнтів призводить до поступового зменшення значення градієнтного кроку  $\lim_{k \rightarrow \infty} \|\bar{\lambda}_k\| = 0$ . Вирішенням недоліку неконтрольованого зменшення  $\bar{\lambda}_k$  є застосування стохастичної апроксимації значення градієнтної складової  $G_k$  значенням  $\mathbb{E}G_k$ . Позначивши  $\beta_k$  як коефіцієнти урахування імпульсу, отримаємо ітераційну послідовність методу середньоквадратичного адаптивного кроку (RMSProp) [8]:

$$\bar{G}_k = (1 - \beta_k) \bar{G}_{k-1} + \beta_k \|\nabla F(x^k, \xi^k)\|^2, \beta_k \rightarrow 0 \quad (6)$$

$$x^{k+1} = x^k - \frac{\lambda_k}{\sqrt{\bar{G}_k + \varepsilon}} \circ \nabla_x F(x^k, \xi^k), x^0 = y^0 \in \mathbb{R}^n, k = 0, 1, \dots$$

Дана модифікація алгоритму робить адаптивним значення математичного сподівання (момент першого порядку) стохастичної величини  $\nabla_x F(x^k, \xi^k)$ , проте має місце застосування стохастичної апроксимації для значення

незміщеної дисперсії  $v_i$  (момент другого порядку). Застосуємо модель стохастичної апроксимації для моментів першого та другого порядків [9]:

$$\begin{aligned} m_k &= (1 - \alpha_k)m_{k-1} + \alpha_k \nabla_x F(x^k, \xi^k), \alpha_k \rightarrow 0 \\ \nu_k &= (1 - \beta_k)\nu_{k-1} + \beta_k \|\nabla_x F(x^k, \xi^k)\|^2, \beta_k \rightarrow 0 \end{aligned}$$

Значення моментів статистичної вибірки можуть бути зміщеними величинами, тому підлягають додатковому коректуванню:

$$\bar{m}_k = \frac{m_k}{1 - \alpha_k}, \bar{\nu}_k = \frac{\nu_k}{1 - \beta_k}$$

Величини  $\bar{m}_k$  та  $\bar{\nu}_k$  є поточними статистичними оцінками градієнтів  $\nabla f(x^k)$  та норм  $\|\nabla f(x^k)\|^2$  для цільової функції задачі (1). Таким чином, отримуємо ітераційну послідовність *методу адаптивної оцінки моментів (ADAM)* [9]:

$$x^{k+1} = x^k - \frac{\lambda_k}{\sqrt{\bar{\nu}_k + \varepsilon}} \circ \bar{m}_k, x^0 \in \mathbb{R}^n, k = 0, 1, \dots \quad (7)$$

### Швидкість збіжності стохастичного кінцеве-різницевого методу

Розглянемо так звані згладжені (або усереднені) функції наступного вигляду:

$$F_h(x) = \frac{1}{\nu_n h^n} \int_{B_h(x)} F(y) dy, B_h(x) = \{y \in \mathbb{R}^n, \|y - x\|_2 \leq h\}$$

де  $\|\cdot\|_2$  - Евклідова норма в просторі  $\mathbb{R}^n$ . Нехай  $\nu_n$  та  $S_n$  - одинична куля та її поверхня у  $\mathbb{R}^n$ ,  $S_n$  та  $\nu_n = nS_n$  - площа поверхні та об'єм одиничної кулі у  $\mathbb{R}^n$ ,  $N(y)$  - зовнішня одинична нормаль до одиничної кулі у  $\mathbb{R}^n$  в точці  $y$ .

Якщо функція  $F(\cdot)$  ліпшицева з субдиференціалом  $\partial F(\cdot)$ , то її градієнт можна представити у вигляді:

$$\nabla F_h(x) = \frac{1}{\nu_n h^n} \int_{B_h(x)} \partial F(y) dy$$

Градієнт  $F_h(x)$ , коли  $F(x)$  неперервна, обчислюється поверхневими інтегралами:

$$\begin{aligned} \nabla F_h(x) &= \frac{n}{h S_n} \int_{S_n} F(x + hy) N(y) dS = \\ &= \frac{1}{2h S_n} \int_{S_n} (F(x + hy) - F(x - hy)) N(y) dS \end{aligned}$$

Якщо позначити  $\tilde{y}$  випадковий вектор, рівномірно розподілений на одиничній сфері  $S_1(0) = \{y \in \mathbb{R}^n : \|y\| = 1\}$ , то градієнт  $\nabla F_h(x)$  можна представити у вигляді математичних очікувань:



$$\begin{aligned}\nabla F_h(x) &= \frac{n}{h} \mathbb{E}_{\tilde{y}}(F(x + h\tilde{y}) - F(x))\tilde{y} = \\ &= \frac{n}{2h} \mathbb{E}_{\tilde{y}}(F(x + h\tilde{y}) - F(x - h\tilde{y}))\tilde{y} = \\ \mathbb{E}_{\tilde{y}} \left\| \frac{n}{2h} (F(x + h\tilde{y}) - F(x - h\tilde{y}))\tilde{y} \right\|^2 &\leq nL^2\end{aligned}$$

На основі подання градієнтів згладжених функцій у вигляді математичних очікувань центральних і симетричних різниць значень функції по випадковим напрямкам побудовані і обґрунтовані стохастичні методи оптимізації негладких ліпшицевих функцій [6].

**Теорема 1** [6]. Нехай  $F(x)$  - ліпшицева функція в околиці кулі  $\{x \in \mathbb{R}^n : \|x\| \leq r\}$  з субградієнтними множинами Кларка  $\partial F(x)$ , і вся множина  $F^* = \{x : \|x\| \leq r, 0 \in \partial F(x)\}$  не є щільною, а є, наприклад, скінченною. Нехай відома точка  $\bar{x}_0$ ,  $\|\bar{x}_0\| < r$ , така, що  $F(\bar{x}_0) < \min_{\|x\|=r} F(x)$ . Розглянемо ітераційний процес:

$$x_{k+1} = \begin{cases} x_k - \rho_k S_k, & \|x_k\| \leq r, \\ x_0, & \|x_k\| > r \end{cases}, x_0 = \bar{x}_0, k = 0, 1, \dots$$

де  $S_k = \frac{1}{2h_k}(F(x_k + h_k\tilde{y}_k) - F(x_k - h_k\tilde{y}_k))$ ; випадкові вектори  $\{\tilde{y}_k\}$  незалежні і рівномірно розподілені на поверхні одиничної сфери з центром у початку координат. Нехай виконані умови:

$$\begin{aligned}\rho_k > 0, h_k \geq h_{k+1} > 0 \\ \lim_k h_k = \lim_k \frac{\rho_k}{h_k} = \lim_k \frac{h_k - h_{k+1}}{\rho_k} = 0 \\ \sum_{k=0}^{\infty} \rho_k^2 < \infty, \sum_{k=0}^{\infty} \rho_k = +\infty\end{aligned}$$

Тоді всі граничні точки випадкової послідовності  $\{x_k\}$  майже напевно належать множині  $X^* = \{x : \|x\| \leq r, 0 \in \partial F(x)\}$  і числова послідовність  $\{F(x_k)\}$  майже напевно має границю.

Має місце наступний результат про швидкість збіжності стохастичного кінцеве-різницевого методу з усередненням траєкторії на опуклих функціях.

**Теорема 2.** Припустимо, що функція  $F(\cdot)$  опукла і ліпшицева з деякою константою  $L$  в Евклідовій нормі на опуклій обмеженій множині  $X \subseteq \mathbb{R}^n$  і справедлива оцінка  $\|X\| = \sup_{x \in X} \|x\| \leq D$ . Послідовність  $\{x_t\}$  будується рекурентно за наступним правилом:

$$x_{t+1} = \pi_x(x_t - \rho_t \eta_t), t = 1, \dots, T$$

де  $\eta_t = \frac{1}{K} \sum_{k=1}^K \frac{1}{2h_t}(F(x_t + h_t\tilde{y}_t^k) - F(x_t - h_t\tilde{y}_t^k))\tilde{y}_t^k$  - випадкові напрямки спуску;  $\rho_t$  - детерміновані крокові множники;  $h_t$  - параметри згладжування; випадкові вектори  $\{\tilde{y}_t^k\}$  незалежні і рівномірно розподілені

на поверхні одиничної сфери;  $\pi_x$  - оператор проектування на опуклу множину  $X$ .

Тоді при постійному кроці  $\rho = \frac{D}{L} \sqrt{\frac{K}{T}}$  точка  $\bar{x}_T = \frac{1}{T} \sum_{t=1}^T x_t$  задовольняє співвідношенню:

$$\begin{cases} \mathbb{E}[F_h(\bar{x}_T) - \min_{x \in X} F_h(x)] \leq \frac{nLD}{\sqrt{KT}}, & h > \frac{nD}{\sqrt{KT}} \\ \mathbb{E}[F(\bar{x}_T) - \min_{x \in X} F(x)] \leq \frac{3nLD}{\sqrt{KT}}, & h \leq \frac{nD}{\sqrt{KT}} \end{cases}$$

При змінному кроці  $\rho_t = \frac{D\sqrt{K}}{L\sqrt{t}}$  для любого  $t$  і  $\bar{x}_t = \frac{1}{t} \sum_{k=1}^t x_k$  виконано наступну умову:  $\mathbb{E}F_h(\bar{x}_t) - F_h(x^*) \leq \frac{nLD}{2\sqrt{K}} \times \frac{1+\ln t}{\sqrt{t-1}}$ .

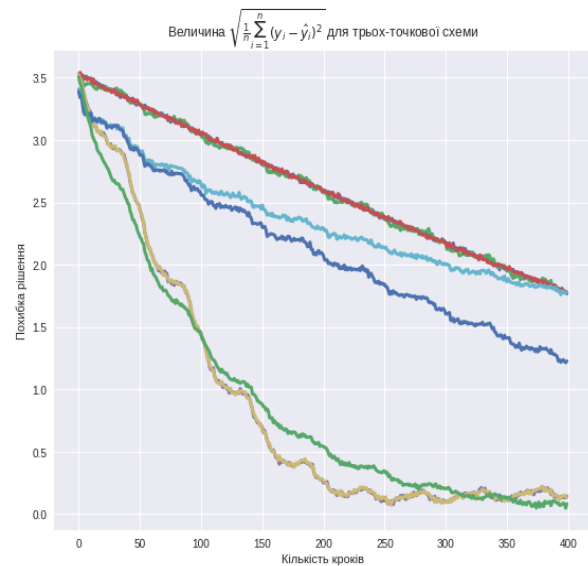
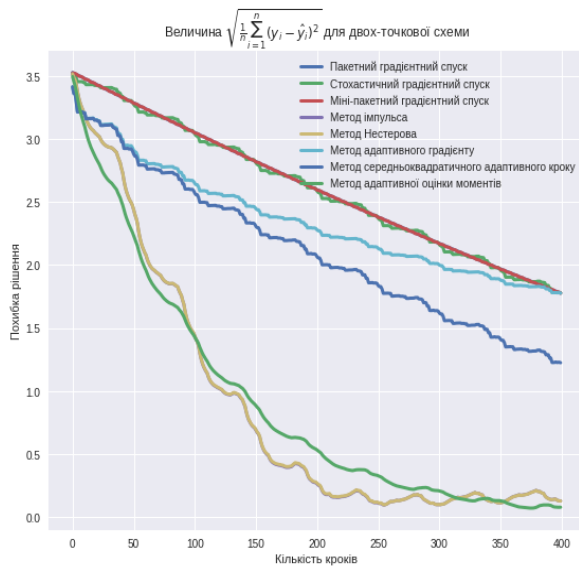
**Зауваження.** Величина  $M = 2KT$  дорівнює числу обчислень функції за  $T$  кроків методу, тому при постійному кроці  $\rho = \frac{D}{L} \sqrt{\frac{K}{T}} = \frac{D\sqrt{M}}{\sqrt{2T}}$  методу справедлива оцінка:

$$\mathbb{E}[F_h(\bar{x}_T) - \min_{x \in X} F_h(x)] \leq \frac{nLD}{\sqrt{KT}} = \frac{\sqrt{2nLD}}{\sqrt{M}}$$

Таким чином, точність методу залежить від числа обчислень функції і не залежить від числа  $k$  кінцевих різниць для оцінки градієнту на кожній ітерації. Але алгоритм з усередненням градієнтів може бути розпаралеленим, а без усереднення – ні. З отриманої оцінки видно, що вона може бути покращена як за рахунок збільшення  $k$  так і  $t$ , наприклад, тільки за рахунок збільшення  $k$  при  $T=1$ . Але з виду оцінки видно, що середнє значення похибки є недостатньою мірою точності алгоритму, потрібно також враховувати дисперсію похибки.

### Експериментальні дослідження

Для експериментального порівняння швидкості збіжності стохастичних градієнтних методів, використаємо градієнтні методи (2) - (7) з однаково заданими постійними гіперпараметрами  $\rho_k \equiv \rho = 0.01, \alpha_k \equiv \alpha = 0.1, \beta_k \equiv \beta = 0.01, \gamma_k \equiv \gamma = 0.8$  для задачі оптимізації логістичної регресії, де значення статистичної вибірки  $\{\xi_k\}$  є нормально розподіленими згідно закону  $\mathcal{N}(0, 1)$ . Розв'язок задачі оптимізації виконано для двох-точкових та трьох-точкових кінцевих різниць, значення точності кожного з градієнтних методів в залежності від ітерації зображено на рисунку. Відповідно до отриманих результатів можна зробити висновки, що точність методу залежить від числа обчислень функції і не залежить від числа  $k$  кінцевих різниць для оцінки градієнту на кожній ітерації, що власне є експериментальним доведенням теоретичних властивостей.



## Висновки

1. Розглянуті класичні стохастичні градієнтні методи та методи з адаптивним регулюванням їх параметрів з детальним описом алгоритмів та порівняльною характеристикою.

2. Встановлені умови збіжності в середньому стохастичного кінцево-різницевого методу оптимізації опуклих функцій на опуклій допустимій множині.

3. Показано, що значення швидкості збіжності в середньому для певного стохастичного методу оптимізації не є повністю адекватною мірою якості відповідного алгоритму. Показано, що оцінка точності стохастичного кінцево-різницевого методу залежить від числа обчислень функції і не залежить від числа кінцевих різниць для оцінки градієнту на кожній ітерації.

4. Експериментально порівняно швидкість збіжності різних адаптивних стохастичних методів оптимізації і продемонстрована перевага адаптивних методів над не адаптивними.

**Подяка.** Роботу виконано в рамках проекту Національного фонду досліджень України № 2020.02/0121 “Analytical methods and machine learning in control theory and decision-making in conditions of conflict and uncertainty”.

## Література

1. Nemirovsky A. and D. Yudin. Informational Complexity and Efficient Methods for Solution of Convex Extremal Problems, J. Wiley & Sons, New York, 1983.
2. Bottou, L., Curtis, F.E., Nocedal, J. (2018). Optimization Methods for Large-Scale Machine Learning. SIAM Review. Vol. 60(2), pp. 223–311. DOI:10.1137/16m1080173
3. The Impact of the Mini-batch Size on the Variance of Gradients in Stochastic Gradient Descent [Text] / Xin Qian, Diego Klabjan. – arXiv Preprint. – Optimization and Control. – 2020. – arXiv:2004.13146.
4. Duchi J., Jordan M., Wainwright M., Wibisono A. Optimal rates for zero-order optimization: the power of two function evaluations. [Text] // IEEE Transactions on Information Theory.-2015.-Vol. 61(5).-pp. 2788-2806.
5. Polyak, B.T. (1987). Introduction to Optimization. Optimization Software.
6. Норкин В.И. Два алгоритма случайного поиска для минимизации недифференцируемых функций. Математические методы исследования операций и теории надежности. [Text] / Ред. Ю.М. Ермольев, И.Н. Коваленко. Киев: Институт кибернетики. – 1978. – pp. 36–40.
7. Нестеров Ю.Е. Метод решения задачи выпуклого программирования со скоростью сходимости  $O(1/k^2)$ . Доклады АН СССР. 1983. Том 269. № 3. С.543-547.
8. Duchi J., Hazan E., Singer Y. Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research, 12(Jul.):2121–2159, 2011.
9. Kingma D., Ba J. Adam: a method for stochastic optimization. ICLR, 2015. arXiv:1412.6980v9 [cs.LG] 30 Jan 2017.

**К.т.н., доцент Олефір О.С., студент Андрійчук В.В.**

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

## **ПРОЕКТУВАННЯ МОДЕЛІ УНІВЕРСАЛЬНОГО ЗАГАЛЬНОПЛАНЕТАРНОГО ЧАСУ**

### **Abstract**

**Olefir Olexandr, assoc. prof., PhD; Victor Andriychuk, student**  
*Designing of the universal planetary time model*

*This paper concerns the task of optimal triangulation construction. The classical Delaunay triangulation algorithm is studied and discussed. The modified iterative algorithm with the data structure “triangles, edges and vertices” 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**

У багатьох командних та індивідуальних проектах на сучасному ринку розробник стикається з проблемою часу, що власне залежить від сучасної моделі часових поясів. На сьогоднішній день існує багато моделей часу, проте кожна з них надзвичайно віддалена від досконалості [1].

У даній статті пропонується модель єдиного загальнопланетарного часу, що допоможе вирішити більшу частину економічних, комунікаційних, технічних та інших проблем шляхом її впровадження на практиці.

### **Постановка задачі**

Задача полягає в модифікації існуючої системи часових поясів, що є всебічно базована на політичних рішеннях взаємодії окремих альянсів держав, застосуванні відміни понять літнього часу, поглибленої орієнтації на єдиний (можливо гринвіцький часовий пояс) та запровадженні єдиного графіку роботи для всього населення планети без врахування місцевого добового часу.

Мета дослідження – підвищення ефективності використання загальнопланетарної моделі часових поясів.

Об'єкт дослідження – процес проектування моделі універсального загальнопланетарного часу.

Предмет дослідження – математичні моделі різних часів, програмні технології кодування різних видів часів та їх універсалізації.

Методи вирішення задачі: обробка результатів спостережень, дослідження залежностей, регресійний аналіз, абстрагування, узагальнення.

### **Термінологія**

*Часовий пояс* – частина поверхні земної кулі, на якій ухвалено один стандартний час, який ще часто називають місцевим часом. Для зручності відліку поточного часу вся територія земної кулі поділена на 24 часові пояси шириною в середньому  $15^\circ$ . В межах кожного такого поясу встановлюється свій єдиний поясний час.

*Літній час* (відомий також як Daylight Saving Time (DST), згідно з його назвою в американському варіанті англійської мови) — місцевий час, який встановлюється на певній території на літній період року. Зазвичай є більшим на годину від стандартного часу, який іноді некоректно називають «зимовим», прийнятого на цій території [2].

*Всесвітній координований час* (англ. Coordinated Universal Time, фр. Temps universel coordonné, UTC, Time Zulu — «час зулу», Z) — стандарт, за яким суспільство регулює годинники й час. Приблизно відповідає сонячному часу на Гринвіцькому меридіані. Запроваджений 1961 року. Відрізняється на цілу кількість секунд від суворо рівномірного атомного часу (TAI) та не більш як на  $\pm 0,8$  секунди від квазірівномірного всесвітнього часу (UT1) [3].

### **Опис алгоритму**

В основу сучасної системи часових поясів покладено всесвітній координований час, від якого залежить час усіх поясів. Щоб не вводити місцевий сонячний час для кожного значення довготи, поверхня Землі умовно поділена на 24 часові пояси, місцевий час на межах яких змінюється рівно на 1 годину. Географічні часові пояси обмежуються меридіанами, що проходять на  $7,5^\circ$  на схід і захід від середнього меридіана кожного поясу, причому в зоні меридіану гринвіцького діє всесвітній час. Однак насправді, для збереження єдиного часу в межах однієї адміністративної території чи групи територій, межі поясів не збігаються з теоретичними граничними меридіанами.

Реальна кількість часових поясів більша за 24, оскільки в низці країн правило цілої різниці в годинах від всесвітнього часу порушується — місцевий час кратний півгодини або чверті години. Крім того, поблизу лінії зміни дати в Тихому океані є території, які використовують час додаткових поясів: +13 і навіть +14 годин.

Подекуди деякі часові пояси пропадають — час цих поясів не використовується, що характерно для малонаселених регіонів, що знаходяться вище за широту приблизно  $60^\circ$ , наприклад: Аляска, Гренландія, північні регіони Росії. На Північному та Південному полюсах меридіани сходяться в одній точці, тому там поняття часових поясів та місцевого сонячного часу втрачають сенс. Вважається, що на полюсах має використовуватися всесвітній час, однак, наприклад, на станції Амундсен-Скотт (Південний полюс) діє час Нової Зеландії.

До введення системи часових поясів у кожному населеному пункті використовувався свій місцевий сонячний час, який визначається географічною довготою конкретного населеного пункту або найближчого міста. Система стандартного часу (або, як було прийнято називати в Росії, поясного часу) з'явилася наприкінці XIX століття як спроба покінчити з такою плутаниною. Необхідність запровадження такого стандарту стала особливо актуальною з розвитком мережі залізниць — якщо графіки руху поїздів склалися за місцевим сонячним часом кожного міста, це могло викликати не тільки незручності та плутанину, а й аварії. Перші проекти стандартизації часу з'явилися та були реалізовані у Великій Британії.

Як так вийшло, що люди ускладнили собі життя і начебто б спочатку правильну ідею довели в деяких країнах до абсурду?

Звичайно, уніфікація часу по всій планеті – це правильно. Планета з розрізнених держав стає космополітичним осередком. Так, ще збереглися національні держави, але сама економіка, міграція населення стали глобальними. Однак, важливо розібратися, чи справді саме існуюче зараз на планеті рішення щодо уніфікації часу є найкращим рішенням.

Як і раніше, у людей по всьому світу виникають проблеми — від життєво-побутових і робочих до технічних. Як і раніше, плутанина з часом приносить складнощі для людей, які подорожують між різними часовими поясами, ведуть бізнес з різними часовими поясами. Та й підтримка інфраструктури щодо обслуговування поточної моделі уніфікованого часу ускладнена наявністю різних поясів, переходів між ними, а також переходом на літній та зимовий час.

Серед головних недоліків часових поясів можна виділити політизованість їх розподілу, незручність окремих поясів на кшталт місця, де наявна лінія зміни дати, переходи на літній і зимовий час, зміщення окремих поясів за недостатньо зрозумілих причин, складність визначення назв часів (а усього їх 200) [4].

Як часто буває в таких випадках, досить просто вийти за межі звичного сприйняття світу, щоб рішення знайшлося — чому б не піти далі в уніфікації часу по всій планеті та скасувати часові пояси? Наприклад, розробник із Москви легко домовиться з командою з Силіконової Долини

про дзвінок — 15 годин за Грінвічем. Припустимо, цей приклад не виразний, оскільки розробники, які працюють у різних часових поясах, звикли переводити час. Але для самої економіки набагато вигідніше уніфікований час, що поведе за собою скорочення трудових та енергетичних ресурсів.

## **Висновки**

У даній роботі оглянуто теоретичний базис понять часу та часових поясів, проведено розрахунки та виявлено закономірність щодо впровадження алгоритму універсального часу та спрощення роботи у багатьох сферах життя, також економії трудових ресурсів. Усі часи та необхідність їх розділення з'являлись на всіх етапах розвитку людства, зокрема багато рішень мали політичне або соціальне підґрунтя.

Слід зазначити, що проектування моделі єдиного часу стверджує запровадження строго однакового часу незалежно від місцевого, середньодобового часу чи інших видів часів – дана модель базується на уже існуючому гринвіцькому поясі, проте всі підприємства по всій планеті працюють за різним добовим часом і однаковим фактичним.

У подальшому було б доцільно розробити специфічний додаток, що дозволяє привести до єдиного стандарту не лише всі часові пояси, а й усі види часу як такого.

## **Література**

1. *Сидоренков Н. С.* Часы, время и неравномерность вращения Земли//Земля и Вселенная. 1971. —№ 3. — С. 26–31.
2. *Михайлов А.* Часовые пояса и «летнее» время//Наука и жизнь. — 1981. —№ 3. —С. 33–34.
3. *Рыхлова Л. В.* Что такое «координированное время»? // Земля и Вселенная. — 1979. — № 3. — С. 34–39.
4. *Пушкин С. Б.* Новый Государственный эталон времени//Земля и Вселенная. — 1984. — № 5. — С. 51–55.



Д.т.н., професор Ориняк І.В., студент Кольцов Д.Р.

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

## ЗАСТОСУВАННЯ ТЕОРІЇ БАЛОК ДЛЯ ПОБУДОВИ ДВІЧІ ДИФЕРЕНЦІЙОВАНИХ ПЛОСКИХ КРИВИХ ПО НЕТОЧНИМ ДИСКРЕТНИМ ДАНИМ

### Abstract

Igor V. Orynyak, prof., PhD; Koltsov R. Dmytro, student

*Employment of the beam theory for twice differentiable curve fitting of discrete points with noise*

*The smoothing of measured noisy positions of discrete points has a big significance in various industries and computer graphic application. The most common approach is a compilation of nonlinear functional based on least square method with subsequent iteration process of its realization. The idea of work consists in employment of the beam theory technique for smoothing of the geometrical noise data. Three important improvements in relation to the classical beam approximation are made. First, the characteristic curve points are initially connected by the straight line segments, where the local coordinates system are established. Second, the initial angles between them are accounted for in the conjugation equations, which are to provide the angular continuity. Third, the measured points are considered as the placement of elastic supports, which compliances are iteratively increased to provide the required level of smoothness. Several examples of the reconstruction of the closed denoised curve from an unstructured and highly noisy 2D point cloud are presented.*

### Вступ

Відтворення замкнутих ліній із набору неточно визначених точок має велике значення в обчислювальній геометрії. Більше того, ця задача має застосування в аналізі зображень, комп'ютерному зору та для обернених фізичних задач. Як приклад розглянемо екстракцію силуетів із нечітких глибинних зображень. Існуючі методи в основному базуються на згладжуванні з допомогою ймовірнісної функції Гауса, яка дає гладкі форми [1]. Проте її застосування може приводити до втрати корисної інформації. Більше того, недоліком подібних методів є те, що не враховується реальний шум (середньо квадратичне відхилення), навіть якщо він відомий [2]. Дуже

часто відтворення геометрії відбувається під певні геометричні криві (коло, еліпс, гвинтова лінія) з невідомими параметрами. В цьому випадку методом найменших квадратів складається функціонал, який ітераційно мінімізується [3]. Проте історично першими сплайнами були рішення, що базувалися на здобутках теорії балок і фактично використовували відоме рівняння трьох моментів [4]. Згодом ця теорія була доповнена можливістю застосування пружних опор, що моделювали вимірні положення з похибкою, а збільшення податливості пружинки опор приводило до більшого згладжування лінії пружної балки [5]. Ці ідеї були розвинуті в роботі авторів [6], де початкові прямолінійні ділянки визначали локальний координатний базис, а між собою з'єднувалися під кутами, і власне завданням деформування було згладжування цих кутів [6]. В цій роботі балкові пружинні сплайни застосовуються ітераційно з врахуванням великих переміщень і зміни локальних векторів нормалі до кожної ділянки [7].

### Постановка задачі і опис моделі

Розглянемо математичну модель прямолінійної балки на згин, її параметри та рівняння. Балка характеризується вектором стану  $\vec{Y}(x)$  в кожній точці, що характеризується поточною координатою довжини  $x$ :

$$\vec{Y}(x) = \text{column}\{W(x); \theta(x); M(x); Q(x)\} \quad (1a)$$

Де  $W(x)$  це переміщення кожної точки,  $\theta(x)$  – кут повороту (нахилу деформованої лінії відносно початкового напрямку),  $M(x)$  згинальні моменти,  $Q(x)$  -поперечна сила. Ці параметри зв'язані диференціальними залежностями, що можуть трохи різну форму і рішення для різних варіантів навантаження і поведінки балки. Тут розглядається найпростіша статична балка, так звана балка Ейлера-Бернуллі, що описана в усіх підручниках по опору матеріалів, і яка не є навантаженою на жодній ділянці. Тоді рішення для балки є наступним:

$$\begin{pmatrix} \vec{Y}(x) \end{pmatrix} = \begin{bmatrix} 1 & x & \frac{x^2}{2EI} & \frac{x^3}{6EI} \\ 0 & 1 & \frac{x}{EI} & \frac{x^2}{2EI} \\ 0 & 0 & 1 & x \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} \vec{Y}_0 \end{pmatrix} = [A(x)](\vec{Y}_0) \quad (16)$$

Де -  $\vec{Y}_0 = \vec{Y}(x = 0)$  – вектор стану в початковій точці розглядуваної ділянки.

На кожній ітерації геометрія досліджуваної фігури представляється як набір прямих відрізків, що з'єднуються під певним кутом. Тобто, початки і кінці кожного відрізка  $i$  на кожній ітерації  $j$  мають певне положення, і вони

співпадають з положеннями сусідніх відрізків, Мал 1. З кожним з цих відрізків ми пов'язуємо локальний дотичний вектор  $\vec{t}_i$  та нормальний вектор  $\vec{n}_i$ . Напрямок локального нормального вектору вибирається так, щоб найкоротший поворот від дотичного до нормального вектору відбувався за годинниковою стрілкою, тобто якщо вектор  $\vec{t}_i$  заданий як

$$\vec{t}_i(x) = a_i \vec{i} + b_i \vec{j} \quad (1в)$$

То вектор  $\vec{n}_i$  знаходиться по формулі:

$$\vec{n}_i(x) = c_i \vec{i} + d_i \vec{j} \quad (1г)$$

де

$$\begin{pmatrix} c_n \\ d_n \end{pmatrix} = \begin{pmatrix} \cos(-\pi/2) & -\sin(-\pi/2) \\ \sin(-\pi/2) & \cos(-\pi/2) \end{pmatrix} \begin{pmatrix} a_n \\ b_n \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} a_n \\ b_n \end{pmatrix} \quad (1д)$$

Важливим моментом моделі є врахування кута між двома сусідніми відрізками, який має спрямлятися на кожній наступній ітерації, цей кут позначається як  $\psi_{i-1,i}$ , Рис 1, та рахується по формулі

$$\sin(\psi_{i-1,i}) = \vec{t}_{i-1} * \vec{t}_i \quad (1е)$$

Іншою особливістю розглядуваного підходу є розгляд початкових вхідних неточних точок як фіксованих точок опор, де приєднані пружні опори, в яких виникає додаткова сила, що пропорційна відхиленню даної точки контура (початок відрізка) від положення опори. Це відхилення визначається як різниця початкового положення точки  $\Pi_i$  перед початком розрахунку і додатковим переміщенням точки  $W_{i,1}$  на даній ітерації розрахунку, Рис 1.

Таким чином, для кожного відрізка сформуємо два набори невідомих по чотири основних параметрів – на початку і в кінці кожної ділянки, які будемо позначати відповідно нижніми індексами “0” та “1”. Таким чином, з врахуванням прийнятих позначень і вказаних вище умов на границі між елементами запишемо чотири рівняння *спряження*:

$$W_{i+1,0} = W_{i,1} \quad (2а)$$

$$\theta_{i+1,0} = \theta_{i,1} - \psi_{i,i+1} \quad (2б)$$

$$M_{i+1,0} = M_{i,1} \quad (2в)$$

$$Q_{i+1,0} = Q_{i,1} - C_i(W_{i,1} - \Pi_i) \quad (2г)$$

Де  $C_i$  – це податливість опори ( характеристика пружинки). Рівняння спряження доповнюються рівняннями зв'язку, що отримуються із рівняння (1б), яке записується для точок кінця і початку ділянки. Запишемо його скорочено в векторній формі:

$$(\vec{Y}_{i,1}) = \vec{Y}_{i,0}(s_i) = [A_{i,k}(s_i)](\vec{Y}_{i,0}) \quad (3)$$

Тут  $s_i$  – це довжина ділянки. Таким чином, для кожної ділянки маємо 8 невідомих та по 4 рівнянь зв'язку та спряження.

### Основна ідея алгоритму

Основна ідея алгоритму полягає в побудови двічі диференційованих плоских кривих по неточним дискретним даним. Ці точки розглядаються як точки положення пружних опор і звичайно залишаються незмінними в процесі рішення. На відміну від них положення пов'язаних з ними точки початків ділянок змінюються від ітерації до ітерації, враховуючи розрахункові значення переміщень  $W$  отриманих на попередній ітерації. Перед першою ітерацією вважаємо, що положення початків відрізків і точок опор співпадають, тобто  $P_i^{j=0} = 0$ .

Після розрахунку нових  $W$  для всіх відрізків уточнюється положення всіх інших точок, включаючи першу і останню. Вважається що  $W$  направлена по нормалі до контура, яка залежить не тільки від початкової нормалі  $\vec{n}_i$ , але і сама нормаль в процесі розрахунку відхиляється на кут  $\theta(x)$ , який розраховується по другій із формул (1б). Таким чином, рівняння уточненої нормалі приймає вигляд:

$$\begin{pmatrix} c_n(x) \\ d_n(x) \end{pmatrix} = \begin{pmatrix} \cos(-\pi/2 - \theta(x)) & -\sin(-\pi/2 - \theta(x)) \\ \sin(-\pi/2 - \theta(x)) & \cos(-\pi/2 - \theta(x)) \end{pmatrix} \begin{pmatrix} a_n \\ b_n \end{pmatrix} \quad (4a)$$

А саме положення  $\vec{D}_j(x)$  кожної точки  $x$  перебудовується по формулі:

$$\vec{D}_j(x) = \vec{B}_j(x) + W(x)\vec{n}_n(x) \quad (4б)$$

Де  $\vec{B}_j(x)$  – початкове положення точки на прямому відрізку, а  $W(x)$  розраховане положення на кожній ітерації відносно початкового прямого відрізка і координати кожної точки.

Потім відбувається уточнення геометрії граничних точок відрізків, знаходиться нова відстань від опор до цих відрізків з врахуванням знаків, уточняється нове значення податливості опори. В ітераційному процесі ми починаємо з досить жорстких опор і поступово зменшуємо їх податливість, поки не буде задоволений суб'єктивний чи об'єктивний критерій достатності згладжування.

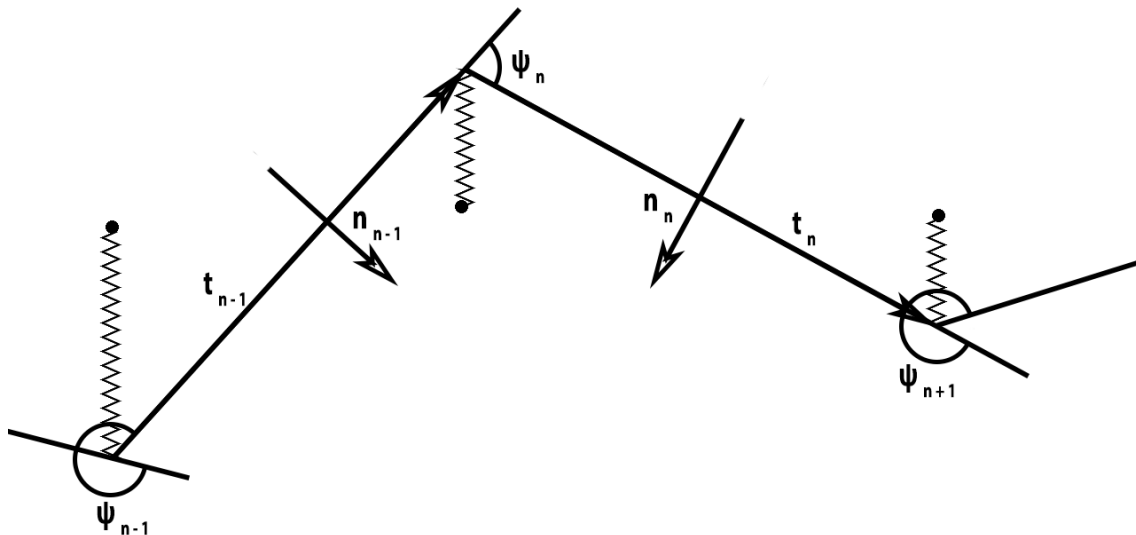
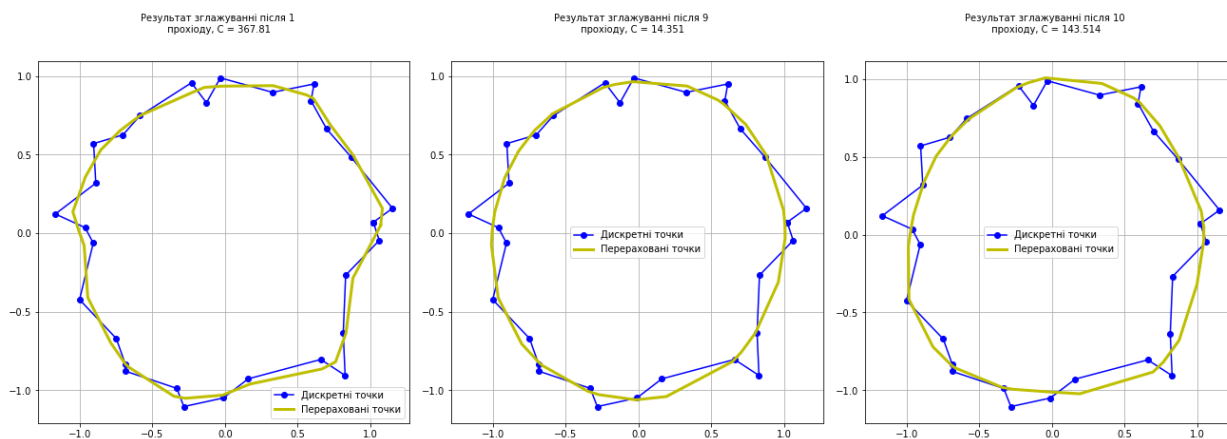


Рис 1. Схема застосування дискретних балок на пружних опорах

### Результати розрахунку

Розглянемо спочатку повне коло. Для даного прикладу було згенеровано 30 точок ідеального кола, потім до цих точок було додано випадкове число в діапазоні від  $[-1/5; 1/5]$ . Для згладжування цього контуру було зроблено 10 ітерацій.



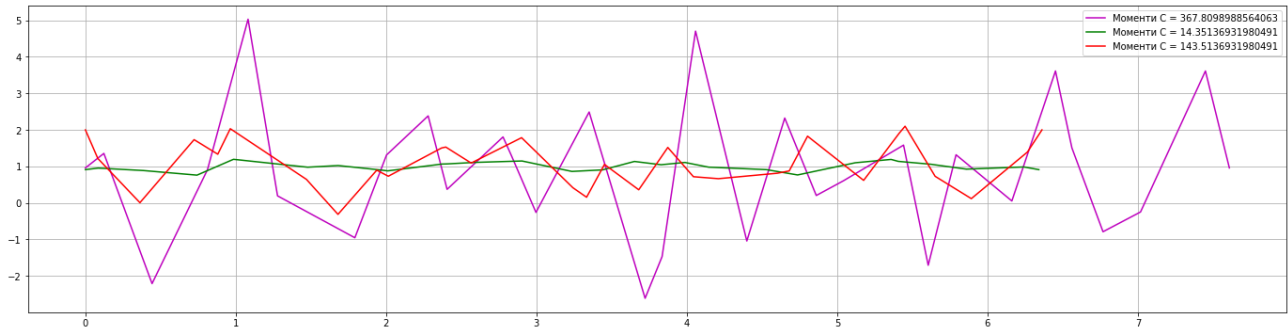
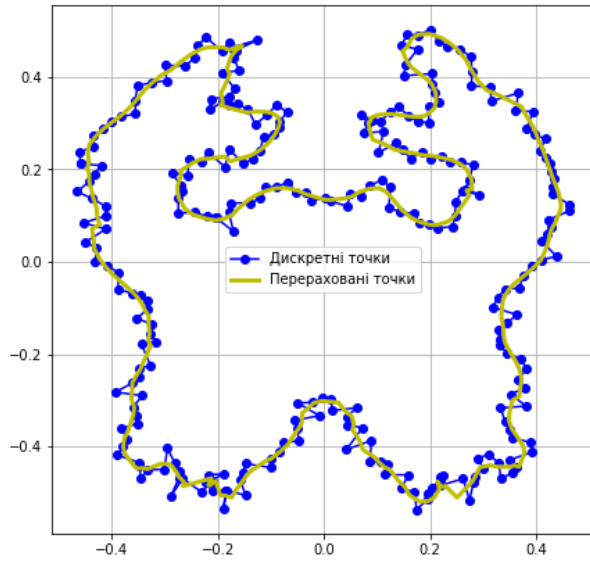


Рис 2. Еволюція зміни контуру від ітерацій, та розрахункові кривизни

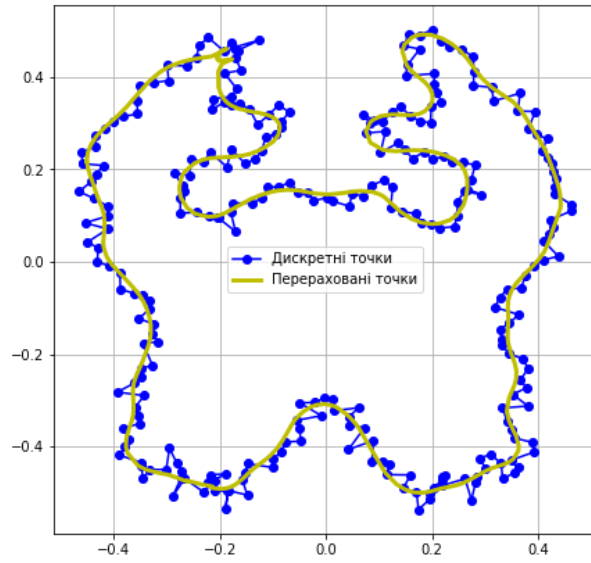
Як бачимо, на останній із проведених ітерацій розрахований контур не тільки візуально нагадує коло, але і аналітично визначені кривизни (другі похідні від положення), досить близькі до значення 1, тобто відтворюють коло.

Розглянемо силуети фігур приведених в [2]. В якості силуетів було обрано 4 фігури, що задавались набором послідовних точок зі згенерованим шумом [2]. Деякі з них мали точки, що перетинаються. Цей алгоритм дозволяє вирішити цю проблему і впорядкувати точки в правильному порядку, приклад наведено нижче. Для всіх силуетів, окрім яблука, було зроблено по 10 ітерацій, для яблука було достатньо 4. На кожній ітерації ми зменшуючи коефіцієнт жорсткості в 1.5 рази, а на останній для перевірки правильності процедури підіймали його в декілька разів. Через таку реалізацію прями, що перетинаються ставали на свої місця і при останньому збільшенні коефіцієнт податливості зсувались до опорних точок при цьому не утворюючи перетину.

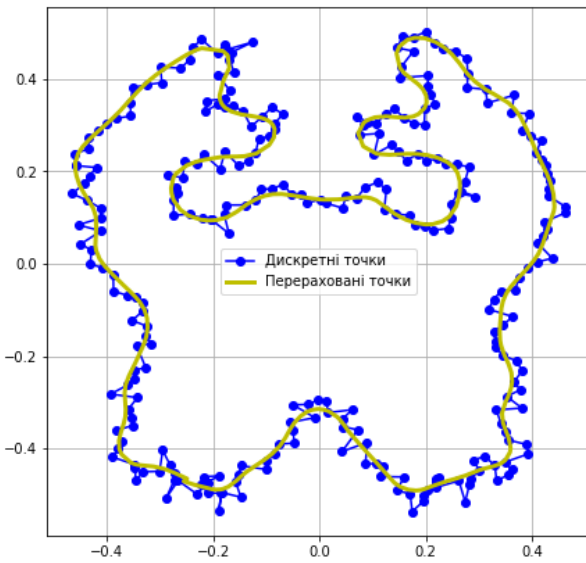
Результат згладжуванні після 1  
проходу,  $C = 28435.921$



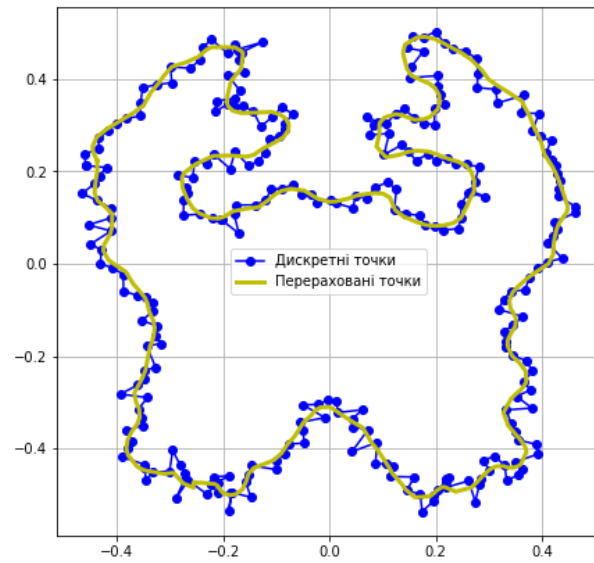
Результат згладжуванні після 4  
проходу,  $C = 27599.625$



Результат згладжуванні після 9  
проходу,  $C = 26260.096$



Результат згладжуванні після 10  
проходу,  $C = 262600.96$



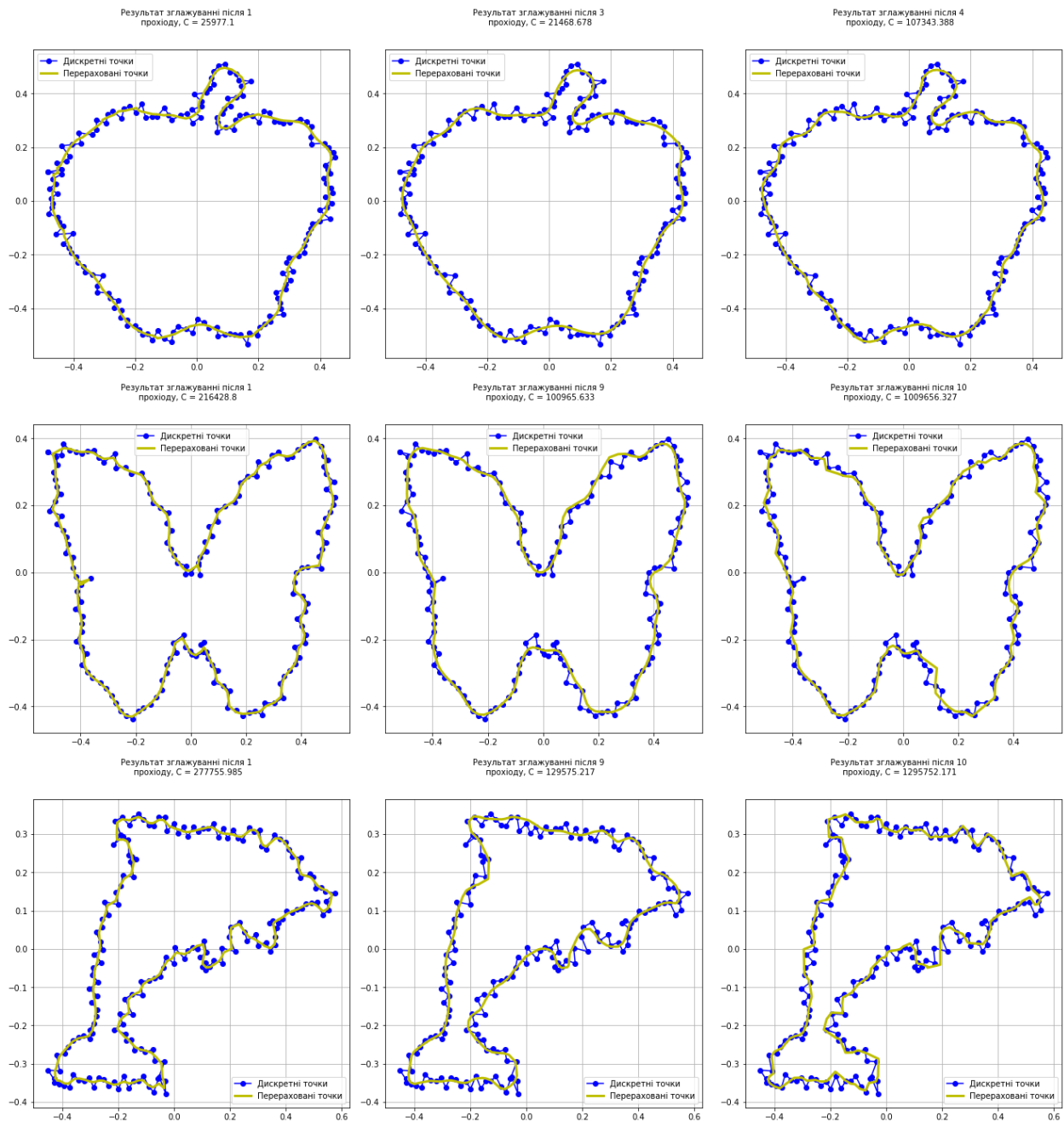


Рис 3. Деякі силуети фігур (сині жирні точки), та згладжені контури на різних ітераціях розрахунку.

## Висновки

В роботі на основі попереднього досвіду аналізу геометрії інженерних конструкцій створено більш універсальний підхід до згладжування



довільного масиву геометричних точок, визначених з похибкою, для отримання неперервних замкнутих контурів. Отримані наступні результати:

1. На основі теорії прямолінійних балок, що розміщені на дискретних пружних опорах, сформульовані загальні рівняння, що включають чотири рівняння зв'язку та чотири рівняння спряження. Останні мають на меті, зокрема, вирівняти кут неспіввісності, що виникає при прямолінійному початковому розміщенні ділянок фігури.

2. Сформовано алгоритм уточнення положення точок фігури, що враховує зміну нормалі до ділянки як результат розрахунку нового деформованого положення. Розроблено механізм уточнення положення точок ділянок шляхом послідовного зменшення жорсткості (збільшення податливостей опор) в процесі розрахунку.

3. Проведені тестування даних з літературних джерел показали ефективність і перспективність розробленого методу згладжування.

## Література

1. *Mark S. Nixon, Alberto S. Aguado.* Feature Extraction & Image Processing for Computer Vision. Academic Press. 2013, 609p. doi. 10.1016/C2011-0-06935-1
2. *Stefan Ohrhallinger, Michael Wimmer.* StretchDenoise: Parametric Curve Reconstruction with Guarantees by Separating Connectivity from Residual Uncertainty of Samples. PG '18: Proceedings of the 26th Pacific Conference on Computer Graphics and Applications: Short Papers, October 2018 Pages 1–4. <https://doi.org/10.2312/pg.20181266>
3. *Sung Joon Ahn.* Least Squares Orthogonal Distance Fitting of Curves and Surfaces in Space. Springer, Berlin, 2004, 125 p. doi: 10.1007/b104017
4. *Алберг Дж., Нильсон Э., Уолш Дж.* Теория сплайнов и ее приложения. Пер с англ. М.: Мир. 1972 318 с.
5. *P. H. Wagner, X. Luo, and K. A. Stelson.* Smoothing curvature and torsion with spring splines, Computer-Aided Design, 27, No. 8, 615–626 (1995)
6. *Orynyak I., Lokhman I, Bohdan A.* The Spring Splines Procedure with Prescribed Accuracy for Determination of the Global (Pipe Centerline) as well as the Local (Dent) Curvatures/ Proceedings of IPC 2012 9th International Pipeline Conference, September 24 – 28 Calgary, Alberta, Canada, IPC2012-90127.
7. *Orynyak I., Mazuryk R., Orynyak A.* (2020). “Basic (discontinuous) and smoothing up (conjugated) solutions in transfer matrix method for static geometrically nonlinear beam and cable in plane”. Journal of Engineering Mechanics, V46, 5. doi: 10.1061/(ASCE)EM.1943-7889.0001753

УДК 519.688

Д.т.н., професор Ориняк І.В., студент Піць В.В.

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

## ПОРІВНЯННЯ МЕТОДІВ ГАЛЬОРКІНА, НАВ'Є ТА КІНЦЕВИХ РІЗНИЦЬ НА ПРИКЛАДІ СТАТИЧНОГО ДЕФОРМУВАННЯ ПРЯМОКУТНОЇ ПЛАСТИНИ

### Abstract

Igor V. Orynyak, prof., PhD; Pits Vadym, student  
*Comparison of Galerkin, Navier and finite difference methods on example of static deformation of rectangular plate*

*The main goal of paper is justification of application of exponential functions in Galerkin method for analysis of deformation of the rectangular plate. The detailed procedure of constructions of the trial functions which satisfy to the boundary conditions is described, and the general algorithm of method application is elaborated and numerically implemented. To show the efficiency of method it is compared with accurate results obtained by Navier method for simply supported plate. To extend the possibilities of comparison at other boundary conditions the finite difference approach is also realized, and its efficiency is analyzed, too.*

### Вступ

Прямокутні пластини широко застосовуються в корабельній, авіаційній техніці, в будівельній галузі, тощо. Тонка пластина, приварена по краю до жорсткої рами і піддана нормальному тиску, є найпоширенішим елементом в інженерній практиці, і вимагає правильного розрахунку її товщини [1]. Ця задача історично була однією із перших математичних задач в частинних похідних. Розробка методів її рішення привело до появи численних методів рішення задач математичної фізики [2], і до цього часу аналіз пластин є популярною темою в математичних і інженерних журналах. Математичні моделі деформування пластин застосовуються для інтерполяції просторових даних та обробки зображень в медицині та техніці [3]. Нещодавно для рішення подібних задач було запропоновано використовувати метод Гальоркіна, де базисні функції будуються на основі експоненціальних функцій [4], МГЕФ. Розширення застосування цих методів вимагає їх

верифікації на основі більш трудоміжких але точних методів, до яких відноситься метод кінцевих різниць, МКР. Тому мета даної роботи полягає як в модернізації методу Гальоркіна і верифікації його з допомогою точного методу Нав'є, так і альтернативної реалізації МКР для тестування точності в випадку відносно простих граничних умов, до яких відноситься прямокутна пластина з шарнірними умовами закріплення.

### Постановка задачі

Розглянемо прямокутну пластину, Рис 1, загальною довжиною  $2a$  та шириною  $2b$ . Центр координат  $(x, y)$  пов'яжемо з її серединою. Відповідно до загальної постановки теорії пластин [1] задача зводиться до біквадратного диференційного рівняння відносно прогину кожної точки  $w(x, y)$ :

$$\Delta(\Delta w) = \frac{d^4 w}{dx^4} + 2 \frac{d^4 w}{dy^2 dx^2} + \frac{d^4 w}{dy^4} = q(x, y) \quad (1a)$$

Де  $q(x, y)$  – навантаження по локальному прямокутнику зі сторонами  $2u$  та  $2v$  центрованого в точці  $(\eta, \xi)$ , Рис1. Граничні умови для шарнірно опертої пластини, яка розглядається в роботі:

$$w(x = \pm a, y) = 0, \quad M_x(x = \pm a, y) = 0 \Rightarrow \frac{\partial^2 w}{\partial x^2}(x = \pm a, y) = 0 \quad (1б)$$

$$w(x, y = \pm b) = 0, \quad M_y(x, y = \pm b) = 0 \Rightarrow \frac{\partial^2 w}{\partial y^2}(x, y = \pm b) = 0 \quad (1в)$$

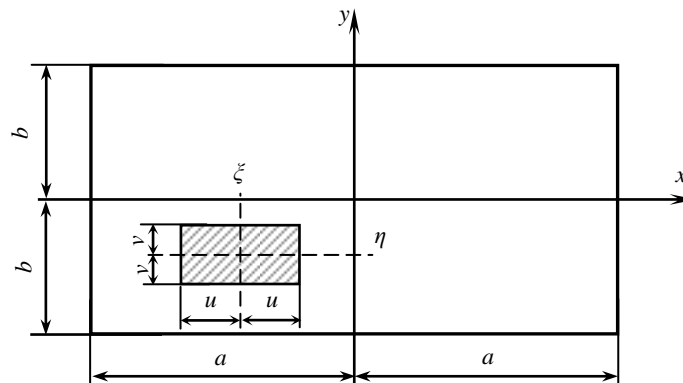


Рис. 1. Частково навантажена пластина

### Теоретичні основи розглядуваних методів

*Метод Нав'є* [1]. Це точний метод, тобто при необмеженій кількості членів розкладу він дає точне рішення. Проте цей метод є застосованим лише для шарнірно опертої пластинки. Тому для верифікації більш універсального запропонованого МГЕФ ми розглядаємо саме таку спрощену задачу.

Представимо функцію навантаження  $q(x, y)$  у вигляді подвійного тригонометричного ряду:

$$f(x, y) = \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} a_{mn} \sin \frac{m\pi x}{a} \sin \frac{n\pi y}{b} \quad (2a)$$

Тоді шукаючи функцію прогину в вигляді подібного тригонометричного ряду отримаємо наступне рішення:

$$w = \frac{1}{\pi^4 D} \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \frac{a_{mn}}{\left(\frac{m^2}{a^2} + \frac{n^2}{b^2}\right)^2} \sin \frac{m\pi x}{a} \sin \frac{n\pi y}{b} \quad (2б)$$

*Метод Гальоркіна, а саме МГЕФ.* Загальна методологія МГ описана в книзі [5], а в роботі [4] дається техніка застосування його при використанні експоненційних функцій для створення базисних (пробних) функцій з допомогою яких наближено рішення диференційне рівняння (1a) при довільних граничних умовах. Основні кроки застосування МГЕФ такі:

1. Вводяться набори ЕФ:

$$\Gamma_k(x, L_x) = \exp\left(\frac{kx}{L_x}\right); \Gamma_m(y, L_y) = \exp\left(\frac{my}{L_y}\right), \quad (3a)$$

де  $k, m$  є цілими числами та  $L_x$  та  $L_y$  - параметри масштабування, значення яких є співставними з характерними розмірами пластини.

2. Наступним кроком є побудова наборів пробних функцій, позначених нижче як  $\Phi_k(x)$  та  $\Phi_m(y)$ . Оскільки кожна пробна функція повинна задовольняти 4 граничним умовам (по дві з кожної сторони), то вони беруться як сума п'яти послідовних експоненціальних функцій  $\Gamma_k(x, L_x)$  та  $\Gamma_m(y, L_y)$ . Наприклад, для  $\Phi_k(x)$  напишемо таке представлення:

$$\Phi_k(x) = \sum_{i=0}^4 \alpha_{k,i} \Gamma_{2+k-i}(x, L_x), \quad K_1 \leq k \leq K_2. \quad (3б)$$

де перші коефіцієнти  $\alpha_{k,0}$  вважаються рівними 1, а всі інші коефіцієнти  $\alpha_{k,i}$  обчислюються за граничними умовами.

3. Вводиться набір двовимірних базових функцій  $\Psi_{k,m}(x, y)$ , що є добутком одновимірних пробних функцій:

$$\Psi_{k,m}(x, y) = \Phi_k(x, L_x) \cdot \Phi_m(y, L_y) \quad (3в)$$

Тоді невідома функція переміщень представляється як сума функцій  $\Psi_{k,m}(x, y)$  помножених на невідомі коефіцієнти  $\beta_{k,m}$ . Підставляючи її в оператор (1a), по чергово множачи функції  $\Psi_{k,m}(x, y)$ , і інтегруючи по всій області пластинки отримаємо систему лінійних алгебраїчних рівнянь відносно невідомих  $\beta_{k,m}$ .

*Метод кінцевих різниць.* Ідея методу полягає в розбивці області на дискретні точки, та заміні диференціальних рівнянь алгебраїчними комбінаціями значень функції в близьких точках до розглядуваної точки.

Таким чином, ми маємо замінити диференціальне рівняння (1а) та граничні умови виду (1б-в) на кінцево різницеve представлення. Використовуючи звичайні К.-Р. схеми, і розглядаючи окіл точки  $x = n\Delta x$ ,  $y = m\Delta y$ , де  $\Delta x$  – крок розбивки по осі  $x$ ,  $\Delta y$  – крок розбивки по осі  $y$  представимо диференційні оператори для граничних умов наступним чином:

$$\left. \frac{d^2 w}{dx^2} \right|_n = \frac{1}{dx} \left( \left. \frac{dw}{dx} \right|_{n+1/2} - \left. \frac{dw}{dx} \right|_{n-1/2} \right) = \frac{w_{n+1}^m - 2w_n^m + w_{n-1}^m}{x^2} \quad (4a)$$

$$\left. \frac{d^2 w}{dy^2} \right|_m = \frac{1}{dy} \left( \left. \frac{dw}{dy} \right|_{m+1/2} - \left. \frac{dw}{dy} \right|_{m-1/2} \right) = \frac{w_n^{m+1} - 2w_n^m + w_n^{m-1}}{y^2} \quad (4б)$$

Тут і нижче для економії запису в К.-Р. схемах ми, по перше, замість  $\Delta x$  записуємо  $x$ , а замість  $\Delta y$  -  $y$ , а по-друге, вводимо позначення для значень переміщень в відповідних точках розбивки як  $w_n^m$ , де верхній індекс відноситься до положення точки по осі  $y$ , а нижній індекс - по  $x$ . Складнішим є представлення диференційного рівняння (1а), так для оператора  $O$  (ліва частина) маємо:

$$O = \frac{w_n^{m+2}}{x^4} + \frac{w_{n+1}^{m+1}}{y^2 x^2} - \frac{2w_n^{m+1}}{y^2 x^2} - \frac{4w_n^{m+1}}{y^4} + \frac{w_{n-1}^{m+1}}{y^2 x^2} + \frac{w_{n+2}^m}{x^4} - \frac{4w_{n+1}^m}{x^4} + \frac{2w_{n+1}^m}{y^2 x^2} + \frac{6w_n^m}{x^4} + \frac{6w_n^m}{y^4} - \frac{4w_n^m}{y^2 x^2} - \frac{4w_{n-1}^m}{x^4} + \frac{2w_{n-1}^m}{y^2 x^2} + \frac{w_{n-2}^m}{x^4} + \frac{w_{n+1}^{m-1}}{y^2 x^2} - \frac{2w_n^{m-1}}{y^2 x^2} - \frac{4w_n^{m-1}}{y^4} + \frac{w_{n-1}^{m-1}}{y^2 x^2} + \frac{w_n^{m-2}}{y^4} \quad (4в)$$

Оскільки техніка реалізації методу Нав'є і МГЕФ описані в роботі [4], то нижче ми даємо основні кроки по реалізації МКР.

### Алгоритм реалізації МСР

Розбиваємо вісь  $x$   $N + 1$  точками  $x_0, x_1, \dots, x_n, \dots, x_{N+1}$  і проводимо через них вертикальні лінії, що звичайно розміщені на відстані  $\Delta_x = x = a/N$ . Розбиваємо вісь  $y$   $M + 1$  точками  $y_0, y_1, \dots, y_m, \dots, y_{M+1}$  і проводимо через них вертикальні лінії, що розміщені між собою на відстані  $\Delta_y = y = b/M$ . Через специфіку формулювання алгоритму і граничних умов, вводимо два горизонтальні ряди фіктивних точок, тобто на лінії

$$y = -\Delta_y \quad \text{та} \quad y = b + \Delta_y \quad (5a)$$

Та два вертикальні ряди точок

$$x = -\Delta_x \quad \text{та} \quad x = a + \Delta_x \quad (5б)$$

Таким чином, ми маємо прямокутник довжиною  $a + 2\Delta_x$  з  $N + 3$ , точками, тобто вертикальними лініями та висотою  $b + 2\Delta_y$  з  $M + 3$ , точками, тобто горизонтальними лініями. Тобто розглядаючи доповнений прямокутник бачимо, що ми всього маємо  $Z = (N + 3)(M + 3)$  вузлових точок, які нумеруються як  $Z_i$ , де:

$$i = (n + 2) + (m + 1)(N + 3) \quad (5в)$$

Всі точки можна поділити на 5 груп:

**Група А.** Абсолютно непотрібні точки, що вводяться лише для зручності нумерації всіх інших точок. Ці точки розміщені в кутках – це точки з координатними номерами  $(x = -1, y = -1)$ ,  $(x = N + 1, y = -1)$ ,  $(x = -1, y = M + 1)$ ,  $(x = N + 1, y = M + 1)$ . Для кожної з них вводимо фіктивне рівняння  $Z_i = 0$ . Всього для групи А є 4 точки і 4 рівняння.

**Група Б.** Граничні кутові точки – їх всього 4. В даному випадку вони особливі тим, що одночасно належать і одній і другій стороні. В цих точках групи Б, ми формуємо три г.у. – тобто рівняння для переміщень – одне, та по одному рівнянню для моментів в кожному напрямку. Всього 12 рівнянь.

**Група В** – це точки, що лежать на границях реальної пластини за виключенням кутових. В цих точках ми записуємо по два рівняння або (1б) або (1в). Тому всього маємо для них  $4(N - 1) + 4(M - 1)$  рівнянь

**Група Г** – це всі внутрішні точки, яких є  $(N - 1)(M - 1)$ . Для кожної такої точки складаємо рівняння вигляду (4в).

**Група Д** – це інші фіктивні точки, яких є  $2(N + M + 2)$ . Рівнянь немає.

Проходячи по черзі по точках пластини, складаємо рівняння для кожної точки відповідно до типу цієї точки. Коефіцієнти при доданках в цих рівняннях записуємо в матрицю розмірності  $(N + 3)(M + 3) \times (N + 3)(M + 3)$ , де номер рядка відповідає рівнянню, а номер стовпчика відповідає порядковому номеру точки, що входить в конкретне рівняння типу (4в) чи умови (4а) та (4в).

Розглянемо спочатку квадратну пластину навантажену рівномірним навантаженням

### **Результати розрахунку**

У зв'язку з методологічним характером цієї роботи нижче ми розглянемо лише квадратні пластини, кожна сторона яких дорівнює 1, тобто  $a = b = 1$ . Також завжди беремо  $L_x = L_y = L$ . Буде досліджено вибір цього досить довільного параметра для прогнозованих результатів. Для статичного навантаження результати представлені лише для переміщень.

Рівномірно розподілене навантаження — це дуже тривіальний випадок, який вимагає невеликої кількості доданків. Отже, розглянемо більш концентровані випадки навантаження. Почнемо з навантаження на 1/4 площі плити, тобто візьмемо, що  $u/a = 0.5$ . Рис. 2, а показує точне переміщення Нав'є ( $N_1 = 400$ ) вздовж осі  $x = y$ . Рис. 2, б та Мал. 2, в показують відхилення наближених результатів для зміщення від точного,  $\delta(w_{\alpha, N}(x, y))$  на кількість

доданків у МГЕФ, де число членів розкладу по кожній координаті  $N$  дорівнює 4 і 8. Ці відхилення обчислюються за такими виразами:

$$\delta(w_{\alpha,N}(x,y)) = \frac{w_{exact}(x,y) - w_{\alpha,N}(x,y)}{w_{exact}^{max}} \quad (6)$$

де  $w_{exact}^{max}$  є максимально «точними» (for  $N_1 = 400$ ) значеннями Нав'є, тут досягнуто в точці  $(x = 0, y = 0)$ . Індекси “ $\alpha$ ” ( $\alpha = \{\Psi, \Theta, \Omega\}$ ) відносяться до наших досліджень найкращого варіанту МНЕФ: а саме для методу Гальоркіна  $\alpha = \Psi$ , для методу моментів  $\alpha = \Theta$  і для методу найменших квадратів  $\alpha = \Omega$ .

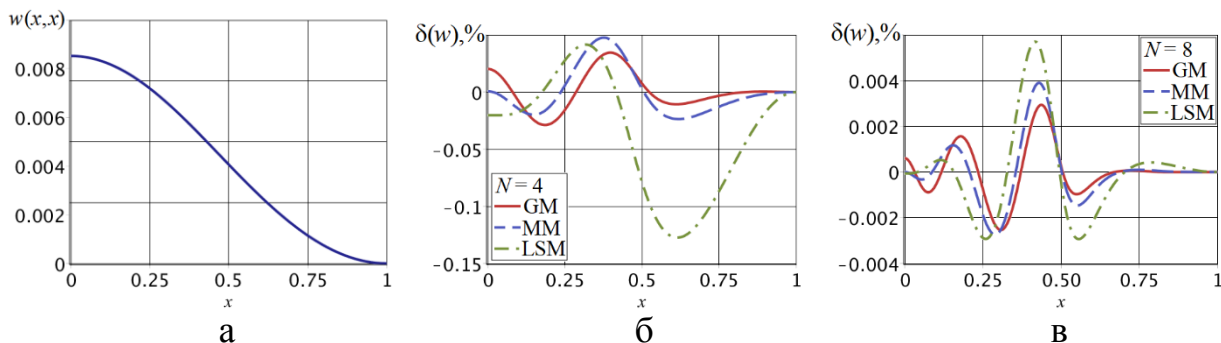


Рис. 2. Точні значення переміщень (а) вздовж осі  $x=y$  (а), та їхні відхилення відповідно до методу МГЕФ обчислені для  $N=4$  (б) та  $N=8$  (в) при  $u = v = 0.5$

Очевидно, що всі варіанти МГЕФ демонструють дуже вражаючу точність, уже при малій кількості членів. Що стосується МКР, то подібна точність досягається при використанні розбитті кожної сторони 40 точками, тобто при використанні 1600 невідомих. Очевидно, що аналітичні методи мають перевагу перед чисельними.

## Висновки

В роботі як перший крок до обґрунтування застосування та демонстрації ефективності МГЕФ розглянута лише прямокутна пластина, для якої проводиться порівняння з точним методом Нав'є. Для того, щоб порівняння було можливим і при інших граничних умовах створений алгоритм рішення подібних задач МКР, для якого описана техніка детальної розбивки, нумерації, алгоритм.

Отримані наступні результати:

1. Для статичного аналізу досліджували лише пластину SSSS, для якої існує точне рішення. Загалом запропонований метод вимагає значно меншої кількості членів, ніж метод Нав'є. Чим більше сконцентровано зовнішнє навантаження, тим менше доданків потрібно в запропонованому методі

порівняно з рішенням Нав'є для досягнення подібної точності. Наприклад, для центрального рівномірного навантаження на квадрат зі стороною 0,04 запропоноване 8-членне рішення порівняно з 80-членним рішенням Нав'є.

2. Запропоновано зручний алгоритм реалізації МКР, що дозволяє автоматично складати рівняння і рішати задачу при великій довільній кількості точок. Для даної задачі метод дає невисоку точність при середній кількості точок. Він буде важливим при аналізі задач зі складними границями, де застосування аналітичних методів є проблематичним.

## Література

1. Тимошенко С.П. Пластинки и оболочки: Пер. с англ./ С.П. Тимошенко, С. Войновский-Кригер. – М.: Наука, 1966. – 636 с.
2. Мелешко В. В. Бигармоническая задача для прямоугольника: история и современность// Мат. методы и физ.-мех. поля, 2004. - Т. 47, No 3. С. 45–63.
3. Bookstein F. L. Principal warps: thin-plate splines and the decomposition of deformations, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, V. 11, N. 6, pp. 567-585, June 1989, doi: 10.1109/34.24792.
4. Orynyak I. V., Bai Yu. P. Application of exponential functions in weighted residuals method in structural mechanics. Part II: static and vibration analysis of rectangular plate / *Mechanics and Advanced Technologies* June 2021, 5(1).
5. Флетчер К. Численные методы на основе метода Галёркина: Пер. с англ. / К. Флетчер. – М.: Мир, 1988. – 352 с.



**Д.ф.-м.н., професор Пашко А.О., студент Канєвський В.В.**

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

## **АЛГОРИТМ ГЕНЕРАЦІЇ ДВІЙКОВИХ ВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ В ЗАДАЧАХ ЗАХИСТУ ІНФОРМАЦІЇ**

### **Abstract**

**Anatolii O. Pashko, prof.; Vladyslav V. Kanievskiy, student**

***Algorithm for generating binary random sequences in information protection tasks***

*This paper investigates an algorithm for constructing binary sequences for cryptographic protection algorithms. The algorithm uses models of implementations of Wiener process. And the models themselves are constructed with specified accuracy and reliability in space norms.*

*To implement the proposed algorithm, a pseudorandom number generator based on the method of residuals is used. The specified generator has a large period and the possibility of paralleling.*

### **Вступ**

На сьогодні захист інформації є одною з найважливіших задач сучасних інформаційних технологій. За останнє десятиліття розвиток обчислювальних технологій досяг тих висот, які певний час здавалися неможливими. Повсякденне їх застосування пояснюється розширенням сфери використання, що ще підвищує роль захисту інформації.

Генерування випадкових послідовностей із заданими ймовірнісними законом та перевірка їх адекватності – одна з найважливіших проблем сучасної криптографії.

Попит на генератори випадкових послідовностей із заданими ймовірнісними розподілами, а також на самі випадкові послідовності, настільки зріс, що за кордоном з'явилися науково-виробничі фірми, які займаються виробництвом і продажем великих масивів випадкових чисел, наприклад, лабораторія NIST.

Генератор випадкових чисел (від англ. Random number generator, скорочено RNG) - алгоритм, що генерує послідовність випадкових чисел,

елементи якої практично незалежні один від одного і підпорядковуються вказаному розподілу.

Існують різні методи для створення послідовностей випадкових чисел: апаратні, табличні та програмні методи генерування послідовностей. Найчастіше використовують програмні генератори, які реалізують послідовності випадкових чисел за рекурентними формулами. Числа, що генеровані даним методом, насправді є псевдовипадковими, тобто, при належній підготовці є вірогідність «зламати» послідовність.

В роботі досліджується вдосконалений алгоритм побудови випадкової двійкової послідовності.

### Використання стохастичних моделей в задачах захисту інформації

Випадкові та псевдовипадкові послідовності і генератори таких послідовностей є невід’ємною складовою будь-якої системи криптографічного захисту інформації. Кожен потоковий шифр можна розглядати як генератор псевдовипадкових послідовностей, від якості якого залежить стійкість алгоритму шифрування. Оцінювання якості псевдовипадкових послідовностей та можливостей їх використання є одним із найактуальніших питань захисту інформації.

В якості генератора для моделювання випадкової послідовності запропоновано алгоритм, що використовує модель вінерівського процесу. Двійкова послідовність будується за формулою

$$b_i = \begin{cases} 1, & W((i+1)dt) - W(idt) \geq 0 \\ 0, & W((i+1)dt) - W(idt) < 0 \end{cases}, \quad (1)$$

де  $i = 0, 1, \dots, N$ ,  $dt = \frac{T}{N}$ ,

$W(t)$  - вінерівський процес.

Нехай  $(\Omega, \mathcal{B}, P)$ - стандартний ймовірнісний простір. Для моделювання гауссових випадкових процесів використовується

представлення процесу у вигляді стохастичного ряду  $\xi(t) = \sum_{k=1}^{\infty} f_k(t) X_k$ , де

$\{X_k\}$  - послідовність незалежних гауссових випадкових величин з нульовим середнім та  $EX_k^2 = \sigma_k^2$ . Модель будується у вигляді

$$S_M(t) = \sum_{k=1}^M f_k(t) X_k \quad [1-3].$$

При реальному моделюванні послідовності  $\{X_k\}$  за рахунок алгоритмів моделювання та точності представлення чисел в

обчислювальних системах отримуємо, як правило, строго субгауссові випадкові величини.

Простір субгауссових випадкових величин є банаховим. Властивості субгауссових величин досліджувались в роботі [4].

Серед гауссівських процесів велике значення має вінерівський та узагальнений вінерівський процеси.

*Означення 1.* Узагальненим вінерівським процесом (дробовим броунівським рухом) з індексом Хюрста  $\alpha \in (0,1)$  називається гауссівський процес  $w_\alpha(t)$ ,  $t \in [0,1]$  такий, що  $w_\alpha(0) = 0$ ,  $EW_\alpha(t) = 0$  та кореляційною

функцією  $R_\alpha(t, s) = \frac{1}{2}(|t|^{2\alpha} + |s|^{2\alpha} - |t-s|^{2\alpha})$ .

При  $\alpha = \frac{1}{2}$  маємо стандартний вінерівський процес.

Будемо розглядати вінерівський процес на відрізку  $T = [0, T]$ . Вінерівський процес можна представити у вигляді випадкового ряду. Так, розклад за власними функціями кореляційного оператора броунівського мосту має вигляд

$$\xi_1(t) = t\eta_0 + \sqrt{2} \sum_{i=1}^{\infty} \frac{\sin(i\pi t)}{i\pi} \eta_i, \quad (2)$$

де  $\{\eta_0, \eta_1, \eta_2, \dots\}$  - незалежні стандартні гауссові випадкові величини,  $\lambda_i = i\pi$  - власні числа кореляційного оператора [5]. Розклад у ряд Фур'є на  $t \in [0,1]$  має вигляд [5].

$$\xi_2(t) = t\eta_0 + \sqrt{2} \sum_{i=1}^{\infty} \left( \eta_{1i} \frac{\sin(2\pi i t)}{2\pi i} + \eta_{2i} \frac{1 - \cos(2\pi i t)}{2\pi i} \right), \quad (3)$$

де  $\{\eta_{1i}, \eta_{2i}\}$  - незалежні стандартні гауссові випадкові величини.

Оскільки  $\sum_{i=1}^{\infty} \frac{1}{(i\pi)^2} < \infty$ , то якщо в представленнях  $\xi_1(t)$  та  $\xi_2(t)$

використати строго субгауссові випадкові величини, тоді процеси  $\xi_1(t)$  та  $\xi_2(t)$  будуть строго субгауссовими випадковими процесами [6]. При моделюванні вінерівських процесів в якості моделі, в залежності від представлення, можна використовувати або модель

$$S_1(t, M) = t\eta_0 + \sqrt{2} \sum_{i=1}^M \frac{\sin(i\pi t)}{i\pi} \eta_i, \quad (4)$$

або модель

$$S_2(t, M) = t\eta_0 + \sqrt{2} \sum_{i=1}^M \left( \eta_{1i} \frac{\sin(2\pi i t)}{2\pi i} + \eta_{2i} \frac{1 - \cos(2\pi i t)}{2\pi i} \right). \quad (5)$$

**Твердження 1[1].** Модель  $s_1(t, M)$  наближає процес  $\xi_1(t)$  з точністю  $\delta > 0$  та надійністю  $1 - \varepsilon$ ,  $0 < \varepsilon < 1$  в  $L_2([0,1])$ , якщо виконуються нерівності  $\delta^2 > J1_{M+1}$

$$\exp\left\{\frac{1}{2}\right\} \frac{\delta}{\sqrt{J1_{M+1}}} \exp\left\{-\frac{\delta^2}{2J1_{M+1}}\right\} \leq \varepsilon$$

$$\text{де } J1_{M+1} = \sum_{i=M+1}^{\infty} (i\pi)^{-2}.$$

**Твердження 2[1].** Модель  $s_2(t, M)$  наближає процес  $\xi_2(t)$  з точністю  $\delta > 0$  та надійністю  $1 - \varepsilon$ ,  $0 < \varepsilon < 1$  в  $L_2([0,1])$ , якщо виконуються нерівності  $\delta^2 > J2_{M+1}$

$$\exp\left\{\frac{1}{2}\right\} \frac{\delta}{\sqrt{J2_{M+1}}} \exp\left\{-\frac{\delta^2}{2J2_{M+1}}\right\} \leq \varepsilon$$

$$\text{де } J2_{M+1} = 2 \sum_{i=M+1}^{\infty} (i\pi)^{-2}.$$

В таблиці 1 наведено кількість доданків для моделей (4) та (5) для різних точності і надійності.

Таблиця 1

Значення  $M$  для різних точності та надійності

№ п/п	Точність, $\delta$	Надійність, $1 - \varepsilon$	Модель 1	Модель 2
1	0.1	0.95	150	400
2	0.1	0.99	270	450
3	0.05	0.95	800	1500
4	0.01	0.95	20000	37000

### Використання паралельних алгоритмів при моделюванні

Розвиток інформаційних технологій, можливості паралельного виконання різних процесів, не оминули і алгоритми статистичного моделювання. Так, в роботі [6] описаний паралельний генератор псевдовипадкових чисел на базі методу залишків. Послідовність чисел рівномірно розподілених в інтервалі  $[0,1]$  будується за формулою

$$u_{i+1} = u_i A \pmod{2^{128}}, \alpha_i = u_i 2^{-128}$$

$$\text{де } u_0 = 1, A = 50059.$$

Для розподілу псевдовипадкових чисел між різними процесорами (чи процесами) використовується наступний алгоритм. Нехай для отримання однієї реалізації випадкового процесу необхідно  $N$  псевдовипадкових чисел, і в наявності є  $L$  процесорів. Алгоритм побудови на кожному процесорі має вигляд

$$u_{i+1,j} = u_{i,j} A(N) \pmod{2^{128}}, \quad j = 1, \dots, L$$

$$\alpha_{i,j} = u_{i,j} 2^{-128}, \quad A(N) = A^N \pmod{2^{128}}.$$

Необхідно для кожного процесору визначити початкові значення:

$$u_{0,1} = 1, \quad u_{0,j+1} = u_{0,j} A(N) \pmod{2^{128}}.$$

Як правило, в якості  $N$  вибирається число рівне кількості псевдовипадкових чисел необхідних для побудови однієї реалізації. В таблиці 1 представлена кількість строго субгауссових величин необхідна для побудови однієї реалізації.

Послідовність строго субгауссових випадкових величин будемо за формулою

$$X_i = \frac{\sqrt{12}}{\sqrt{m}} \sum_{j=1}^m \left( \alpha_j - \frac{1}{2} \right),$$

де  $\alpha_j$  - послідовність псевдовипадкових чисел, рівномірно розподілених в  $[0,1]$ .

Початкові значення для  $L = 10$  представлені в таблиці 2.

Таблиця 2

Початкові значення для паралельного алгоритму

п/п	$u_{0,j} \times 10^{38} (N1)$	$u_{0,j} \times 10^{38} (N2)$
	1,0	1,0
	1,698763363451242	1,968893780093321
	2,803919176383495	0,538211675643232
	3,332248939262583	1,641149765541650
	1,162284001628347	1,748607720979625
	3,198013804400205	0,734308882357589
	3,250532030016155	1,874800259285361
	0,753566783084389	1,640981522953988
	2,464892424987986	3,309399682973288
	0,347975724023908	3,350954410534310
0	0,495977043682777	1,639369046037485

Вибір методу залишків обумовлений тим, що при паралельній реалізації псевдовипадкових чисел отримуємо великий об'єм вибірки. І доцільно використовувати алгоритм, що забезпечує великий період послідовності. Так, описаний метод залишків має період  $Pl = 2^{126} \approx 10^{38}$  [6]. Цей метод пройшов тестування і апробацію, результати приведені в [1, 6].

### Практичні результати

Так як прирости вінерівського процесу є незалежними гауссовими випадковими величинами, то в (1) замість реалізації вінерівського процесу можна використати послідовність незалежних стандартних гауссівських величин.

На рис. 1. представлена реалізація вінерівського процесу, на рис. 2 реалізація двійкової послідовності за вдосконаленим алгоритмом.

Для перевірки якості отриманої послідовності використовувались статистичні методи – перевірка гіпотези про закон розподілу, оцінка кореляційної функції для послідовності (рис. 3). Спектральна щільність (перетворення Фур'є) двійкової послідовності зображена на рис. 4. при  $N = 1000$ . Для цієї ж послідовності були перевірені деякі тести системи NIST.

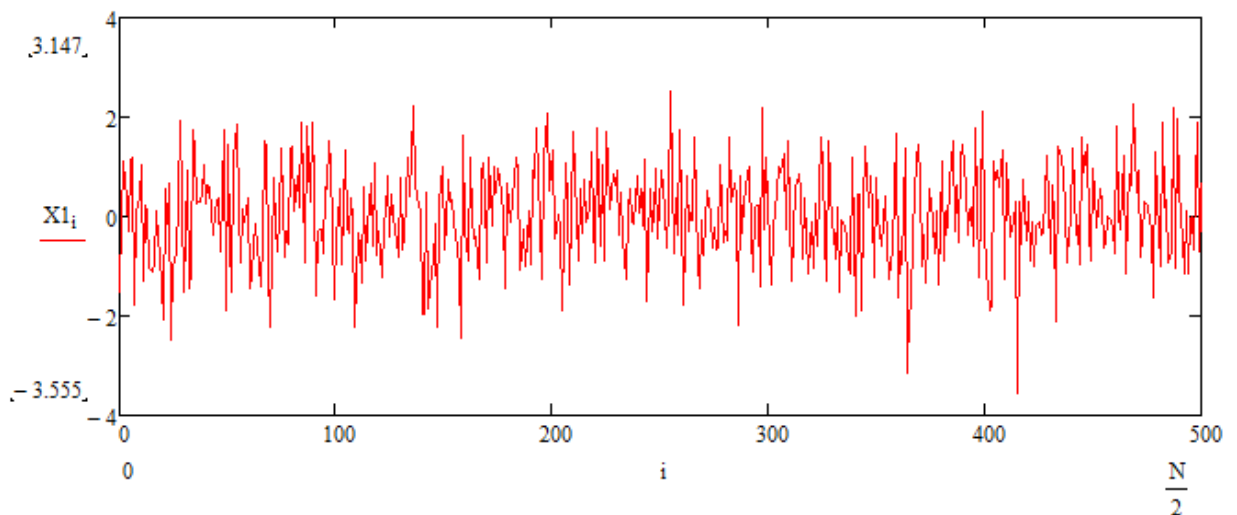


Рис.1. Реалізація вінерівського процесу

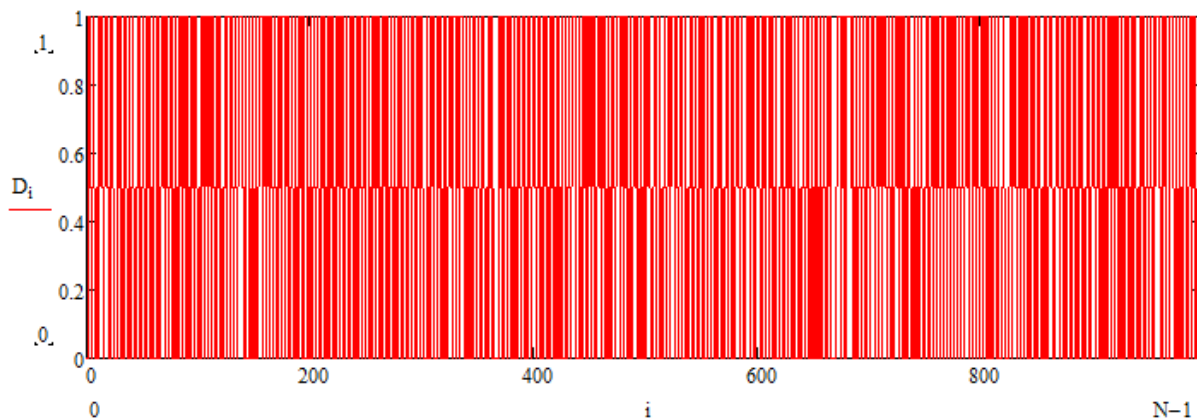


Рис. 2. Реалізація двійкової послідовності

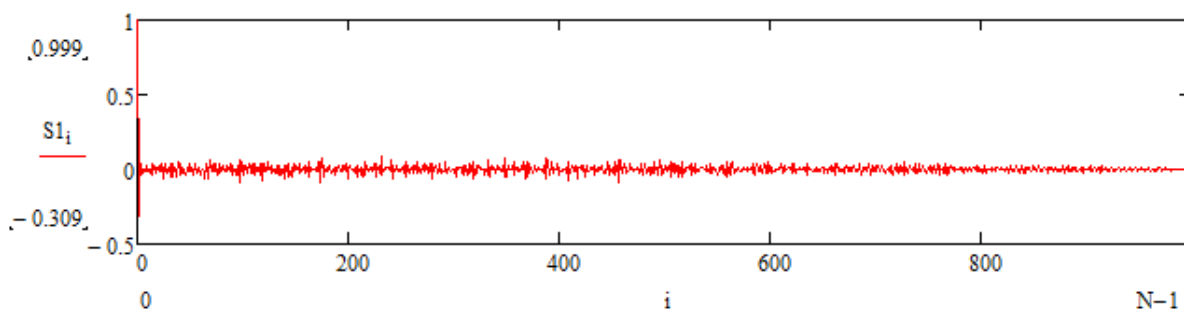


Рис. 3. Кореляційна функція двійкової послідовності

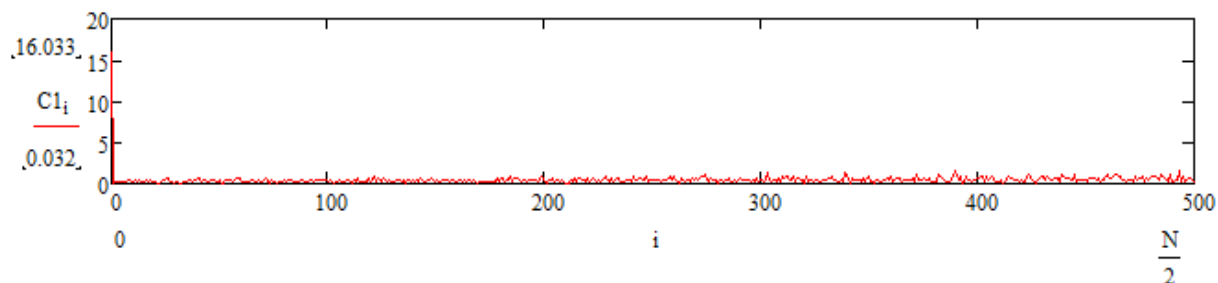


Рис. 4. Спектральна щільність двійкової послідовності

Перевірка статистичної гіпотези про закон розподілу, оцінки кореляційної функції та спектральної щільності підтвердили якість алгоритму побудови двійкових послідовностей, незалежність її значень.

## Висновки

В роботі досліджено і вдосконалено алгоритм побудови двійкових послідовностей для алгоритмів криптографічного захисту. Алгоритм використовує моделі реалізацій вінерівського процесу. А самі моделі будуються з заданими точністю і надійністю в нормах простору  $L_2(T)$ .

Для реалізації розробленого алгоритму запропоновано використання генераторів псевдовипадкових чисел на базі методу залишків. Вказаний метод має великий період і можливість розпаралелення.

Отримані результати покращують існуючі.

## Література

1. Пашко А.О. Моделювання субгауссових випадкових процесів в інформаційних системах / А.О. Пашко // Науковий вісник Чернівецького університету. Серія: Комп'ютерні системи та компоненти. - 2014. - Т. 5, Вип. 1. - С. 48 – 54.
2. Козаченко Ю.В. Точність і надійність моделювання випадкових процесів та полів в рівномірній метриці: монографія /Ю.В. Козаченко, А.О. Пашко. – Київ, ТОВ СІК ГРУП Україна, 2016. - 216с.
3. Козаченко Ю.В. Моделювання випадкових процесів і полів /Ю.В. Козаченко, А.О. Пашко, І.В. Розора. - К.: Задруга, 2007. - 230 с.
4. Булдыгин В.В. Метрические характеристики случайных величин и процессов / В.В. Булдыгин, Ю.В. Козаченко. – К.: ТВиМС, 1998. – 289 с.
5. Гихман И.И. Введение в теорию случайных процессов / И.И. Гихман, А.В. Скороход. – М.: Наука, 1977. – 570 с.
6. Mikhailov G.A. Parallel realization of statistical simulation and random number generators./ G. A. Mikhailov, M. A. Marchenko// - Russ. J. Numer. Anal. Math. Modelling. 2002. V.17.№1. P.113-124.



**К.т.н., доцент Сирота С.В., студент Бандурін В.Ю.**

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

**ІНТЕРАКТИВНИЙ СЕРВІС ПОРІВНЯЛЬНОГО АНАЛІЗУ  
УНІВЕРСИТЕТІВ УКРАЇНИ НА ОСНОВІ ДАНИХ ВСТУПНОЇ  
КАМПАНІЇ**

**Abstract**

**Sergiy Syrota, assoc. prof., PhD; Vladyslav Bandurin, student**  
*An interactive service for comparing Ukrainian universities based on the data of the introductory company.*

*Every year in Ukraine more than 240,000 entrants enter higher education institutions. And they all face a very difficult choice of university and specialty for which they will apply for admission. At the same time, there are a limited number of information resources that can help applicants compare universities, faculties and specialties with each other. Most of the existing tools are not informative enough for comparison or have limited flexibility in use, which does not allow entrants to conveniently make comparisons on different combinations of criteria.*

*Poor awareness of graduates of general secondary education institutions and the lack of a web resource that would help entrants to easily compare universities, faculties and specialties, leads to the fact that entrants with high external evaluation scores do not realize the existence of alternative opportunities for higher education. . Society loses from this, because the best entrants can not but get the best education and because of this may not realize to the fullest their full potential for the benefit of the state.*

**Вступ**

Щороку в Україні до закладів вищої освіти вступають близько 240 000 абітурієнтів. І всі вони стикаються з дуже складним вибором університету та спеціальності, на який вони будуть подавати заявки для вступу. У той же час, існує обмежена кількість інформаційних ресурсів, які здатні допомогти абітурієнтам у порівнянні університетів, факультетів та спеціальностей між собою. Більшість з існуючих інструментів недостатньо інформативні для порівняння або ж мають обмежену гнучкість у використанні, що не дозволяє абітурієнтам зручно робити порівняння за різними комбінаціями критеріїв.

За даними дослідження Cedos, близько 70% абітурієнтів зверталось до сайтів університетів за інформацією про ЗВО та спеціальність, але більшість зазначили, що це джерело було не завжди інформативне для них,

зокрема через застарілість змісту, незручну навігацію, застарілий дизайн тощо [1]. Погана поінформованість випускників закладів повної загальної середньої освіти призводить до того, що абітурієнти з високими балами ЗНО не здогадуються про існування альтернативних можливостей для вступу з більш високим рівнем якості освіти.

### **Постановка задачі**

Існують багато рейтингів університетів за галузями, які показують лише топ 10-100 університетів за різними показниками. Але вони не дають можливості порівняти університети для конкретних спеціальностей та галузей.

Ця стаття має на меті розробку інтерактивного сервісу, який дозволить порівнювати університети за середньозваженим балом ЗНО вступників на обрані спеціальності. У статті описується дослідження факторів, які впливають на вибір ЗВО абітурієнтами, що дає змогу виділити функції сервісу, які були б найбільш корисні для користувачів. Зручний у використанні інтерфейс сервісу дозволить спростити абітурієнтам процес вибору майбутнього університету.

### **Витоки**

За даними аналітичного центру Cedos через конкретну спеціальність до свого ЗВО вступили 40%, через його престиж та репутацію — 37% і через високу якість навчання — 28%, частина респондентів також звужували вибір ЗВО до найближчих до свого місця проживання (самого міста або регіону) або орієнтувалися на «перевірені» — тобто ті, в яких уже навчалися або навчаються знайомі. Для майже половини (46%) була важливою комбінація університет-спеціальність, натомість для 24% — саме конкретна спеціальність і лише для 15% — конкретний університет[1].

Місто, у якому знаходиться університет, також впливає на вибір місця навчання серед абітурієнтів, а подекуди може бути і ключовим фактором. Зокрема, для 5% першокурсників важливим у виборі місця навчання було саме місто, де вони житимуть наступні мінімум чотири роки, а лише потім — навчальний заклад та спеціальність. Однією з причин здобувати вищу освіту для 10% першокурсників став переїзд в інше місто під час навчання. Та водночас для мешканців сільської місцевості і невеликих міст внутрішня міграція була обов'язковою, якщо вони хотіли вступити в університет: 39% першокурсників не мали у своєму населеному пункті ЗВО[2]. Не менш складним є вибір факультету. У багатьох університетах спеціальність може бути представлена на декількох факультетах, це ускладнює задачу абітурієнтам, оскільки з інформації,

представленої на сайтах університетів не завжди можна зрозуміти відмінність між факультетами та дізнатись де навчають краще.

Використовуючи результати цього дослідження, можна зробити висновок, що до факторів, які впливають на рішення абітурієнтів щодо вступу, належать:

- Престижність університету
- Наявність рівень якості навчання на конкретних спеціальностях у конкретних університетах
- Місто, де знаходиться університет та близькість до місця поточного проживання

Враховуючи ці фактори можна виділити функціональні критерії, яким повинен задовольняти сервіс, щоб бути корисним та зручним для його кінцевих користувачів – абітурієнтів. До цих функціональних критеріїв належить:

1. Можливість порівнювати університети за рейтингом серед абітурієнтів;
2. Можливість порівнювати університети в рамках конкретних спеціальностей;
3. Можливість відфільтрувати університети за певними регіонами, які цікавлять абітурієнтів.

### **Опис алгоритму**

Рейтинг університету рахується як середній конкурсний бал на спеціальність, «зважений» на частку абітурієнтів, які були зараховані на неї у виші.

Алгоритм описується наступним чином:

1. Користувач обирає параметри, за якими подивитися рейтинг;
2. Заявки вступників фільтруються за обраними параметрами, з них обираються лише заявки зі статусом «До наказу»;
3. Використовуючи отриману множину заявок для кожного університету рахується середній бал абітурієнтів, зарахованих до нього.

Цей рейтинг може бути інтерпретований абітурієнтами як «Престижність», оскільки він відображає рівень підготовки вступників, які були зараховані у цей навчальний заклад. З цього можна зробити припущення, що чим кращий рівень підготовки студентів, які там навчаються, тим більш навчальне середовище буде сприяти якісному навчанню.

### **Опис реалізації алгоритму**

Для вирішення описаної проблеми був розроблений інтерактивний

веб-сервіс, який будує рейтинг університетів на основі середньозваженого балу абітурієнтів, які були зараховані до нього на обрані спеціальності. Даний сервіс представлений на сайті StudSearch.org[3].

Данні про вступну компанію 2020 року були отримані з ЄДЕБО [2].

Функціонал ресурсу дозволяє в інтерактивному форматі обрати цікаві для абітурієнта параметри для порівняння університетів, такі як:

- Галузь – можна обрати одну чи декілька галузей;
- Спеціальність - можна обрати одну чи декілька спеціальностей;
- Регіон університету- можна обрати один чи декілька регіонів;
- Бюджет чи контракт – порівняння університетів за вступниками на бюджет чи на контракт.

Інтерфейс сервісу можна побачити на Рис 1. На прикладі представлений рейтинг університетів Києва, Львівської та Харківської областей по спеціальності 122 Комп'ютерні науки за середнім балом абітурієнтів, які поступили на контракт. Можливим висновком з цього прикладу може стати те, що найбільш престижним серед обраних університетів на 122 спеціальність на контракт є Український Католицький Університет. Незважаючи на високу вартість навчання, туди поступають абітурієнти із середнім балом 197.7, що значно вище за інші університети.

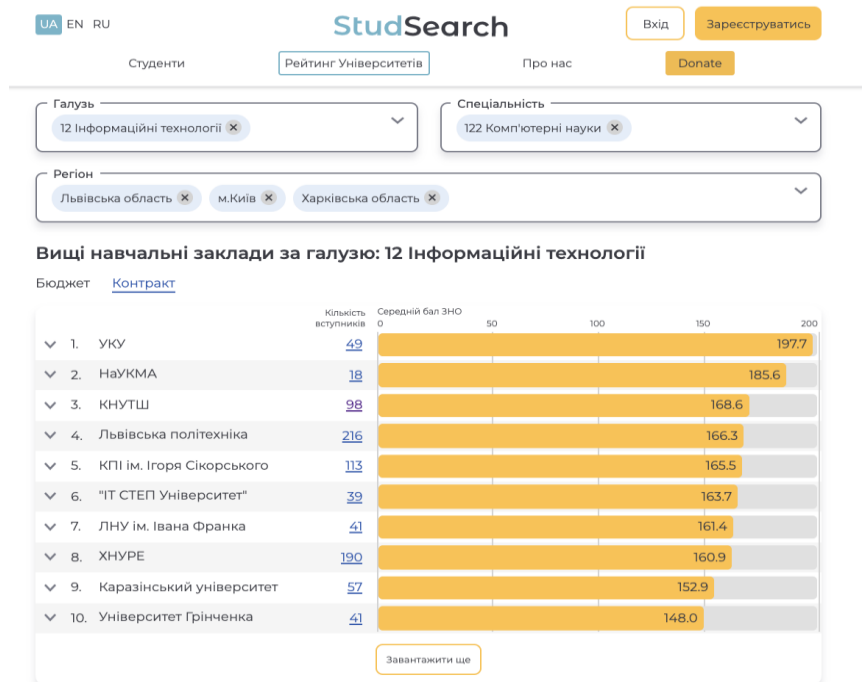


Рис. 1. Інтерфейс порівняння університетів за рейтингом на основі середньозваженого балу вступників, зарахованих на навчання у ЗВО. Інтерфейс представлений на сайті StudSearch.org.

## **Висновки**

Проаналізовано дослідження Cedos щодо факторів, які найбільш впливають на вибір ЗВО абітурієнтами. Виділені функції сервісу порівняння, які є найбільш корисні для користувачів.

Запропоновано алгоритм для порівняння закладів вищої освіти, який дозволяє порівняти ЗВО за різними галузями та спеціальностями.

Алгоритм реалізовано у вигляді інтерактивного веб-сервісу.

Представлений сервіс для порівняння університетів значно спрощує процес вибору абітурієнтам своєї майбутньої вищої освіти. За допомогою нього можна зручно знайти найкращі університети за галузями, спеціальностями та у заданих регіонах.

На базі даного сервісу у подальшому буде розроблена рекомендаційна система ЗВО.

## **Література**

1. Cedos. Вибір вищої освіти після школи: звіт за результатами дослідження [Електронний ресурс] / Марія Куделя, Юлія Назаренко, Тетяна Жерьобкіна – 2020. – Режим доступу до ресурсу: <https://cedos.org.ua/researches/vybir-vyshchoi-osvity-pislia-shkoly-zvit-za-rezultatamy-doslidzhennia/>.
2. Єдина державна електронна база з питань освіти [Електронний ресурс]. Режим доступу до ресурсу: <https://info.edbo.gov.ua/>.
3. StudSearch.org. Рейтинг університетів [Електронний ресурс]. Режим доступу до ресурсу: <https://studsearch.org/rating>.

**К.т.н., доцент Сирота С.В., студент Фельдман М.Г.**

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

## **ОГЛЯД МЕТОДІВ ОБРОБКИ ТЕКСТІВ ПРИРОДНОЇ МОВИ ДЛЯ АВТОМАТИЗОВАНОГО ОЦІНЮВАННЯ СТИЛІСТИЧНОЇ ЯКОСТІ ЕСЕ**

### **Abstract**

**Sergiy Syrota, assoc. prof., PhD; Feldman Mykhailo student**

*An overview of NLP Models for the Essay Stylistic Quality Automated Grading.*

*This paper presents the description of an algorithms used to solve the problem of Automated Essay Scoring, using natural-language processing (NLP). Multi-label classification methods are discussed. The most accurate methods for solution of this problem are proposed. The ways for further research are mentioned as well.*

### **Вступ**

Розгорнута відповідь у вигляді невеликого письмового твору завжди була й залишається важливою частиною академічного процесу. Особливої важливості така форма відповіді набуває коли йдеться про такі предмети як рідна або іноземна мови та література. В обох випадках на остаточну оцінку впливає як стиль твору (об'єм написаного, різноманіття використаної лексики, граматики, орфографія), так і його змістове наповнення. Сьогодні, перевірку розгорнутих письмових відповідей з мови та літератури виконують викладачі, що займає велику частину їх робочого, а подекуди й вільного часу. Звісно, автоматизація цього процесу дуже б їм допомогла.

### **Постановка задачі**

Задача полягає у порівняльному аналізі методів та моделей, які можуть бути використані для створення системи автоматичного оцінювання невеликих прозових творів. Така система повинна мати можливість автоматично оцінити стилістичну якість текстового документа, який не був присутній у навчальній вибірці (даних, які були надані системі для її налаштування під час розробки). Відповідь системи (оцінка) повинна належати до множини дискретних категорій, наприклад, до множини чисел від 1 до 5.

## Термінологія

*Bidirectional Encoder Representations from Transformers (BERT)* – модель мови представлена працівниками компанії Google, що може застосовуватися для побудови векторного представлення слів [1].

*TF-IDF* (від англ. *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[2].

*Word2vec* — це одна з методик обробки природної мови. Алгоритм *word2vec* використовує нейронну мережну модель для навчання пов'язаностей слів із великого корпусу тексту. Як випливає з її назви, *word2vec* представляє кожне окреме слово певним переліком чисел, званим вектором[3].

*LSTM* (від англ. *long short-term memory*, *LSTM*) — це архітектура рекурентних нейронних мереж (РНМ, штучна нейронна мережа), запропонована 1997 року Зеппом Хохрайтером та Юргеном Шмідгубером[4].

## Дослідження існуючих рішень

Можна розглянути процес оцінювання текстових документів з точки зору розподілення великого набору текстових сутностей на невелику кількість дискретних категорій, що відповідають можливим оцінкам. Іншими словами, цей процес можна вважати проблемою статистичної класифікації [5]. Для вирішення такої проблеми можна використовувати методи машинного навчання [6]. Щоб використовувати такі методи необхідно перетворити текстові дані у векторну форму. Існує багато способів зробити таке перетворення. Найвідомішими є так звана модель "мішку слів" (Bag of Words BoW) [7] та TF-IDF. Потім можна використати будь-яку модель машинного навчання, що здатна робити класифікацію. Наприклад, логістичну регресію. Справа в тому, що такий метод не бере до уваги порядок слів, а тільки їх значення, це значить, якщо взяти непогане есе, умовно оцінене на високий бал, потім перемішати всі слова випадковим чином для цього метода це будуть два однакові есе. Тому застосування моделей BoW для оцінки стилістичної якості для прозових творів, навіть таких маленьких як есе, не підходить. Для вирішення цієї проблеми в даній роботі запропоновано використовувати моделі на основі LSTM нейронних мереж та мовні моделі на кшталт BERT. Саме про останні піде мова у цій статті.

## Опис алгоритму

Для того, щоб подати текст на вхід нейронної мережі, необхідно представити його в чисельному вигляді. Найбільш тривіальний спосіб — це створити словник з усіма словами, що зустрічаються у корпусі, і подати мережі номер слова в цьому списку. Наприклад, якщо слово "квітка" стоїть в цьому словнику на місці 15, то на вхід нейронної мережі для цього слова подаємо число 15. Однак, в такому підході існує дуже суттєвий недолік, повністю втрачаються ті асоціації, які приходять на думку будь-якій живій людині, коли вона чує або пише слово "квітка" ("красива", "пахуча", тощо). Існує просте рішення й цієї проблеми: розташувати схожі за змістом слова поруч у словнику. На практиці, кожному слову призначають не одне число, а вектор, скажімо, з 32 чисел. Надалі, якщо потрібно дізнатися наскільки схожі два слова, ми можемо виміряти відстань між їх векторами в 32-мірному просторі. Доступні й арифметичні операції, щоб проілюструвати їх роботу, можна навести декілька класичних прикладів. Найвідомішим є такий: король – чоловік + жінка = королева [5]. Цей приклад доводить нам, що така модель здатна вловити поняття гендеру. Інший приклад: Париж – Франція + Польща = Варшава [5]. У цьому випадку векторна різниця між Парижем і Францією фіксує поняття столиці. Слова у вигляді таких векторів отримали назву Embeddings.

Описаний вище метод ми можемо кожен раз втілювати в життя з нуля, тобто кожен раз створювати векторний простір зі слів, що наявні в нашому корпусі даних. Інструменти для цього, можна знайти в будь-якій сучасній бібліотеці для глибокого навчання. Однак, швидко стало зрозуміло, що набагато швидше та вигідніше заздалегідь навчити подібне векторне представлення слів на якомусь величезному корпусі текстів, наприклад на всій Wikipedia, а в конкретних нейронних мережах використовувати вже готові вектори слів. Існує багато готових, заздалегідь навчених моделей представлень слів векторами, наприклад: word2vec, GloVe, Elmo.

Дуже довгий час в науці та бізнесі для векторизації слів використовувалися саме моделі, що були описані вище. Однак, влітку 2018 року OpenAI (некомерційна дослідницька компанія зі США, що в свою чергу належить компанії Google) довела, що заздалегідь навчена на великих об'ємах даних нейронна мережа на архітектурі Transformer, із великим відривом, показує набагато кращі результати ніж звична комбінація з word2vec векторизації та LSTM нейронної мережі на множині різних завдань по обробці природної мови.

Процес навчання BERT виглядає так: на вхід нейронної мережі подаються фрази, в яких деякі слова маскуються, далі нейронна мережа навчається передбачати замасковані фрагменти. Наприклад, якщо подаємо



на вхід нейронної мережі фразу “the man went to the [MASK1], he bought a [MASK2] of milk”, вона повинна на виході показати слова “store” і “gallon” [6]. Цей етап навчання дозволяє моделі набагато краще розуміти значення слова, беручи до уваги контекст в якому воно вжито. Найточніше розуміння контексту виділило BERT серед конкурентів, які були частково або повністю контекстно незалежні (context-free).

Щоб ще краще розуміти контекст, модель повинна гарно відчувати зв'язок між реченнями, для цього був розроблений ще один етап навчання. На вхід моделі подаються два речення А та Б з одного корпусу даних, задача моделі сказати чи речення Б йде після А чи ні [6].

Треба також зазначити, що ні в першому, ні в другому етапі нам абсолютно не потрібна розмітка даних людиною, інакше кажучи цей спосіб навчання є повністю unsupervised. Саме ця перевага дозволила використати в навчанні величезні корпуси текстових даних.

### **Використання**

Щоб використати модель BERT в нашому завданні (мультикласова класифікація), пропонується поверх неї додати декілька шарів простої повнозв'язної мережі (останнім шаром будуть 5 нейронів з сигмоїд активаціями, які й будуть віддавати нам вірогідності від 0 до 1 для кожного з 5 класів), і донавчати тільки їх, не чіпаючи основну мережу BERT. Така техніка називається Fine Tuning та дуже широко використовується в задачах пов'язаних з зображеннями, наприклад, в класифікації зображень.

Таким чином ми зможемо значно покращити рівень роботи моделі саме для нашого завдання. На офіційній сторінці BERT є багато прикладів, якими ми можемо скористатися, щоб правильно організувати донавчання моделі.

### **Висновки**

У даній роботі було розглянуто принцип роботи новітньої мовної моделі BERT. Були обґрунтовані переваги саме цієї моделі над класичними методами текстової класифікації. Запропоновано способи її використання для вирішення завдання автоматизованого оцінювання стилістичної якості невеликих прозових творів.

### **Література**

1. BERT [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://github.com/google-research/bert>.

2. TF-IDF [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://uk.wikipedia.org/wiki/TF-IDF>.
3. Word2vec [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://en.wikipedia.org/wiki/Word2vec>.
4. LSTM [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://habr.com/ru/company/wunderfund/blog/331310/>
5. Word2vec [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://www.technologyreview.com/2015/09/17/166211/king-man-woman-queen-the-marvelous-mathematics-of-computational-linguistics/>
6. What is BERT? [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://github.com/google-research/bert>

Доцент Соловйов С.О., ст. викладач Копичко С.М.,  
студент Рябоконт З.А.

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

## СТОХАСТИЧНЕ МОДЕЛЮВАННЯ СТРАТЕГІЙ КОНТРОЛЮ ПАНДЕМІЇ COVID-19

### Abstract

**Serhii Soloviov, assoc.prof., PhD, DrSc; Sergei Kopychko, senior lecturer; Zakhar Riabokon, master student**

*Stochastic modeling of COVID-19 pandemic control strategies.*

*This paper presents the description of mathematical modeling of the balance of COVID-19 pandemic control strategies. Here we will consider general information about pandemic, different variations of health care models and our model which we developed based on these studies.*

### Вступ

Сьогодні увесь світ бореться з новою інфекційною хворобою COVID-19, спричиненою коронавірусом SARS-CoV-2. Вперше захворювання було виявлено в Ухані, Хубей, Китай, у грудні 2019 року. Дві величезні проблеми з інфекційними захворюваннями вже виникали раніше через коронавіруси – коронавірусні інфекції SARS у 2003 році та MERS у 2012 році [4].

Кількість інфікованих осіб з кожним днем збільшується. За даними Всесвітньої організації охорони здоров'я (ВООЗ), станом на 20 жовтня 2021 року загальна кількість підтверджених випадків – 245 504 979 та 4 983 516 смертей. Наразі кількість підтверджених випадків в Україні становить 2 803 159 та 64 936 смертей повідомлених 27 жовтня 2021 р.

Щоб зрозуміти поведінку інфекційних захворювань, а також для прогнозування динаміки стратегій контролю захворюваності часто застосовуються математичні моделі. Так, математичні моделі використовувались для прогнозування поширення таких інфекцій, як ВІЛ-інфекція, туберкульоз, лихоманки Ебола, Денге, Зіка, MERS та SARS [7].

Оскільки пандемія COVID-19 є світовою проблемою, вже запропоновано багато математичних моделей для прогнозування поведінки передачі COVID-19. У цій роботі ми пропонуємо новий підхід, який

застований на моделі Маркова і аналізується за допомогою мікросимуляції Монте-Карло [6].

### **Постановка задачі**

Основна задача полягає у розробці моделі, за допомогою якої можливо вивчати та досліджувати ефективність вакцинації чи карантину, та обирати відповідну оптимальну стратегію з урахуванням стохастичності епідемічного процесу COVID-19 та інших факторів. В запропонованій моделі враховано вплив стратегій вакцинопрофілактики та карантину на епідемічний процес COVID-19. На основі цієї моделі буде досліджено ефективність двох основних профілактичних заходів:

- вакцинація сприйнятливих осіб;
- ізоляція інфікованих осіб або інші карантинні заходи.

### **Огляд математичної моделі**

Запропоновано інтегровану модель, яка припускає, що застосування вакцини відбуватиметься відповідно до існуючих наказів та протоколів МОЗ. Нижній рівень моделі передбачає опис подій, які відбуватимуться з певною особою протягом визначеного дискретного проміжку часу (рис. 1). Спочатку населення розподіляється на групи, вакцинованих і не вакцинованих осіб. Для вакцинованих осіб залежно від кількості отриманих доз, двох чи однієї, відбувається подальший розподіл з певною ймовірністю захворювання на COVID-19, або ж особа завершує цикл здоровою. При вчасному виявленні COVID-19 особа ізолюється і перестає впливати на розповсюдження хвороби або ж далі залишається в суспільстві і передає інфекцію серед сприйнятливого населення. З огляду на подальший перебіг захворювання, інфіковану особу або госпіталізують або лікують амбулаторно, після чого ця особа завершує цикл або в стані одужання, або летальним випадком. У випадку не вакцинованих осіб особи відразу діляться на інфікованих COVID-19 і здорових. Ті хто інфікувались, можуть бути госпіталізовані чи лікуватись амбулаторно. У випадку госпіталізації особа залежно від наявності ускладнень завершує цикл моделі одужавшою, або ж з летальним випадком. Також важливо зазначити, що передбачається стохастичність моделі [3]. Кожна можлива випадкова подія в ній пов'язана з ймовірністю, заснованою на доказах, а точна послідовність і час подій, які пережила дана особа, були результатами розіграшу випадкових чисел, що відбувався на кожному етапі моделі.

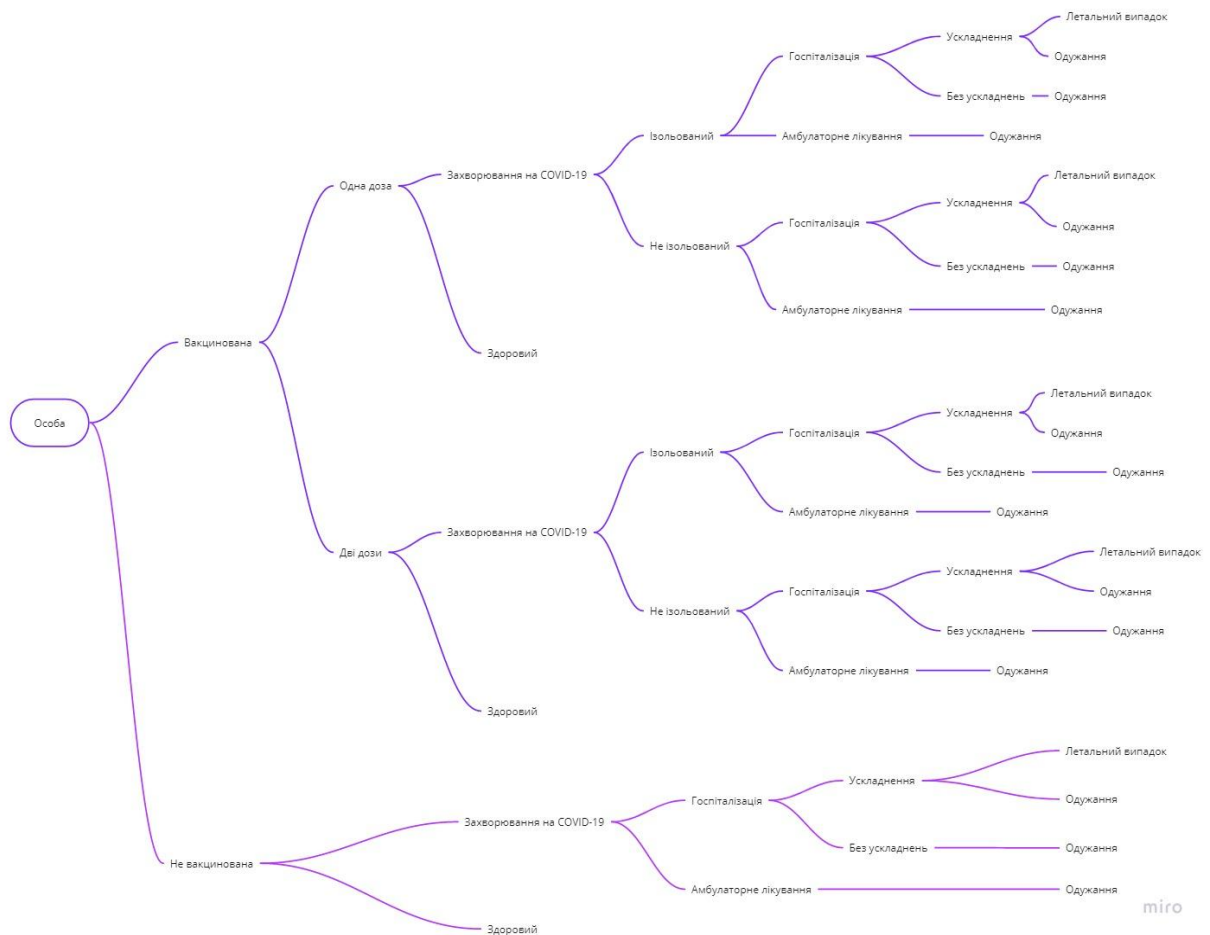


Рис. 1. Індивідуалізована модель захворювання на COVID-19

Модель верхнього (популяційного рівня) передбачає опис динаміки епідемічного процесу в популяції в цілому (рис. 2). Усе населення подається сукупністю часток різних епідеміологічних груп:

1. Сприйнятливі особи: кожна людина сприйнятлива до захворювання від народження, оскільки імунітет не є спадковим;
2. Особи інфіковані на COVID-19: сприйнятливі особи які були інфіковані внаслідок контакту з хворими особами;
3. Населення з набутим поствакцинальним імунітетом: частка сприйнятливого населення яка набула поствакцинального імунітету від вакцинації або від інфікування та наступного розвитку гострого вірусного захворювання;
4. Ізольовані особи: особи, у яких підтверджено, що інфіковані захворюванням і ізольовані від контакту з сприйнятливими особами.

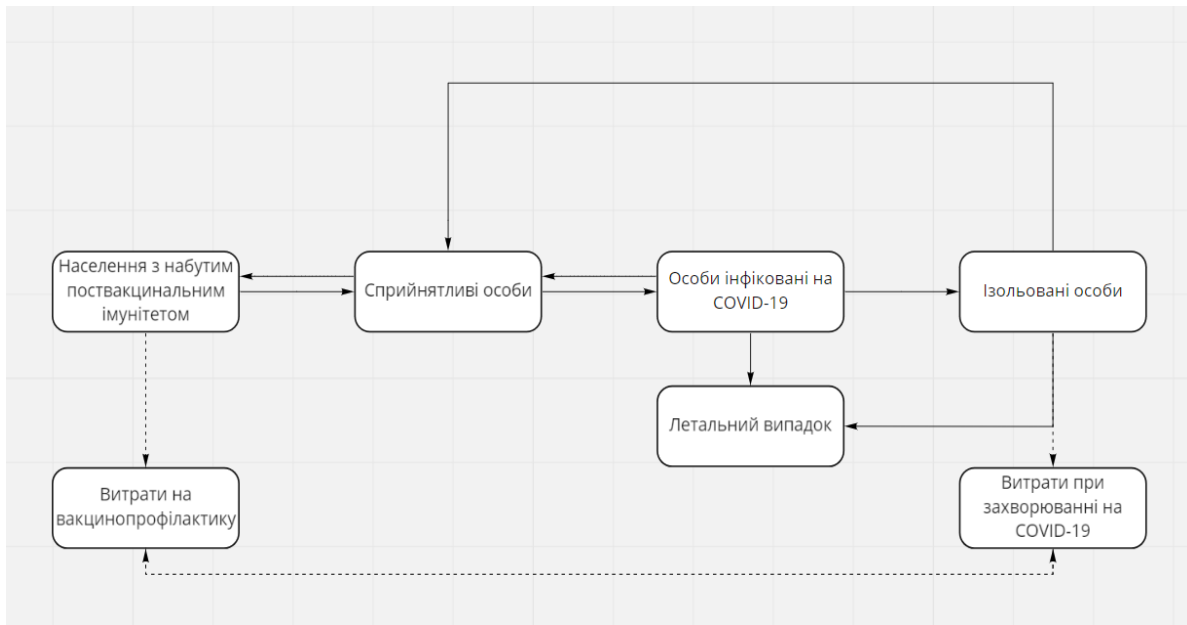


Рис. 2. Популяційна модель епідемічного процесу COVID-19

## Висновки

В цій статті було запропоновано інтегровану модель, яка складається з двох рівнів. Нижній рівень моделі передбачає опис подій, які відбуватимуться з певною особою протягом визначеного дискретного проміжку часу. Модель верхнього (популяційного рівня) передбачає опис динаміки епідемічного процесу в популяції в цілому. Також, слід зазначити про стохастичність цієї моделі, що дає змогу слідувати не одному сценарію при вивченні та дослідженні ефективності вакцинації чи карантину, а дає ширший спектр для досліджень, а в подальшому і для вибору оптимальної стратегії під час пандемії. Слід зазначити, що запропонована модель враховує вплив стратегій вакцинопрофілактики та карантину на епідемічний процес COVID-19.

## Література

1. Sawilowsky SS, Fahoome GC. Statistics via Monte Carlo Simulation with Fortran. Rochester Hills, MI: JMASM; 2003.
2. Kurtz TG. Limit Theorems for sequences of jump Markov processes approximating ordinary differential processes. JAppl Probab. 1971;8:344–56.
3. Whitt W. Stochastic-Process limits: An introduction to stochastic-process limits and their application to queues. New York, NY: Springer; 2002. Available from: <http://www.columbia.edu/~ww2040/preflongno.pdf>.

4. Tezuka S. Uniform Random Numbers: Theory and Practice. Boston, MA: Springer US; 1995. doi:10.1007/978-1-4615-2317-8.
5. Zhao S, Chen H. Modeling the epidemic dynamics and control of COVID-19 outbreak in China. Quantitative biology (Beijing, China). 2020; p. 1–9.
6. Althaus C.L., (2020) Real-time modeling and projections of the COVID-19 epidemic in Switzerland, Institute of Social and Preventive Medicine, University of Bern, Switzerland 20 April 2020 <https://ispmbern.github.io/covid-19/swiss-epidemic-model>.
7. Kiss I.Z., Miller J.C., Simon P.L. (2017) Mathematics of Epidemics on Networks. Springer International Publishing AG 2017 DOI 10.1007/978-3-319-50806-1.

**К. ф.-м. н., доцент Третиник В. В., студент Коринь Й. В.**

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

## **МОДЕЛІ СТАТИСТИЧНОГО АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ПОВЕДІНКИ РИНКУ НЕРУХОМОСТІ**

### **Abstract**

**Violeta V. Tretynyk, assoc. prof., PhD; Yosyp V. Korin, student**  
*Models of statistical analysis and forecasting of real estate market behavior*

*In this paper the algorithm of construction of the system of the analysis of behavior of the real estate market is considered. It is proposed to use different approaches to market research to achieve better accuracy. It is established that to obtain the forecast of the required accuracy, it is necessary to combine different approaches of modeling and forecasting, to validate the accuracy and adequacy of the forecast. The model of time series which will consider a trend, stochastic, cyclic and seasonal components is chosen.*

### **Вступ**

Відколи почався капіталізм, всі особи що займалися торгівлею хотіли й хочуть купувати товар по меншій ціні та продавати по більшій – проста та дієва стратегія торгівлі. З розвитком цивілізації та економіки в цілому почалися появлятися ринки товарів, послуг, акцій, криптовалют та інших цінних речей якими можна торгувати та отримувати прибуток. Аналітики, трейдери та інвестори з часом побачили що ринки циклічні, тобто можна визначити певні закономірності руху ціни активу чи прослідкувати певний тренд. Інструменти аналізу ринку, зокрема ціни активу, дали можливості підвищити ефективність торгів та збільшити прибуток, зазвичай дослідження проводилися на основі статистичних методів аналізу, які можна постійно удосконалювати.

### **Постановка задачі**

Задача полягає у розгляді та описі моделей статистичного аналізу та прогнозування поведінки ринку, на основі історичних даних, циклів ринку та трендів для розробки системи аналізу ринку нерухомості.

Дослідник формулює модель на основі економічної теорії або емпіричних очікувань тенденцій ринку. У моделі присутня змінна відсотка



ціни та абсолютних значень ціни на нерухомість як функція можливих пояснювальних змінних. Модель буде в кращому випадку наближеною, оскільки малоімовірно, що вона охопить кожен відповідну інформацію, яка впливає на ціни на нерухомість. Якщо модель занадто складна, вона буде важкою і практично неактуальною, оскільки її стане важко реалізувати. З іншого боку, якщо модель занадто проста, вона навряд чи зможе охопити та пояснити різницю цін на нерухомість. Прийнятна модель повинна давати гарне наближення, практичне для реалізації і в той же час не ігнорувати важливу інформацію.

### **Виклад основного матеріалу дослідження.**

Розглянемо два типи даних: однофакторний часовий ряд і багатовимірний (причинний) часовий ряд. Типи даних є інтервальними, тобто для них дозволені всі типи обчислень. Наприклад, ціни на нерухомість є реальними даними, які дозволяють виконувати всі види математичних маніпуляцій. Інші відповідні змінні, які можуть пояснити різницю цін на нерухомість, такі як прибуток домогосподарств і міський податок, також є реальними змінними. Дані одновимірного часового ряду – це дані про одну змінну, яку можна побудувати на діаграмі відносно моментів часу, наприклад, щомісячні ціни на нерухомість будинку за рік. Моделі часових рядів зосереджуються на історії змінної, намагаючись сформувати модель, яка може пояснити її майбутнє. Багатоваріантні (причинно-наслідкові) дані часових рядів – це дані про безліч змінних у часі, наприклад, річні дані про ціни на нерухомість, доходи та податки.

В цій статті розглянемо індекс цін на житло міського типу нерухомості. Інші відповідні змінні, які можуть пояснити зміну цін на нерухомість, такі як прибуток і податки, також є реальними змінними, що можуть впливати на цільовий показник. Дані відобразатимуть квартальні зміни цін на типи у межах міста. Результати аналізу даних ринку нерухомості будуть представлені поквартально у відсотковій зміні динаміки вартості активу, що розраховується, як середня зміна прибутковості (у відсотках) по ринку; враховуватиметься вплив динаміки річного прибутку (у відсотках), який розраховується шляхом множення середнього значення ціни активу на кількість періодів, тобто чотири квартали; враховуватиметься також квартальний ризик (у відсотках), що представляє собою рівень оподаткування прибутків, який дорівнює чотири, помноженому на квартальний ризик [1]. Коефіцієнт варіації ( $CV$ ) — це стандартний показник у фінансах, який вимірює співвідношення відсоткового рівня ризику до відсоткового рівня прибутку.

Це корисний показник для порівняння різних інвестицій, оскільки вони ранжуються від найвищого  $CV$ , що вказує на найвищий ризик на

одиницю прибутку, до найнижчого  $CV$ , що вказує на найнижчий ризик на одиницю прибутку.

Методика Бокса-Дженкінса [2] прийнята для визначення лінійної моделі *ARIMA* (Autoregressive Integrated Moving Average), яка може допомогти в оцінці, а потім і в прогнозуванні цін на нерухомість. В джерелі [3] зазначено, що лінійна модель *ARIMA* відображає краще прогнозування цін на житло поза вибіркою. Вибір підходу Бокса-Дженкінса [2], в порівнянні з методами автоматичної ідентифікації моделі, такими як інформаційний критерій Акайке (*AIC*), остаточна помилка прогнозу Акайке (*FPE*) і інформаційний критерій Байєса (*BIC*), полягає в тому, що підхід Бокса-Дженкінса є більш інформативним, оскільки допомагає досліднику зрозуміти дані. *AIC*, *FPE* і *BIC* розглядають часові ряди як чорний ящик, оскільки вони пропонують найкращу модель, не розкриваючи особливостей процесу генерування даних. Крім того, ці критерії часто призводять до надмірної відповідності моделі вибірки [4]. Аналіз багатоваріантних часових рядів зосереджується на зв'язках між залежною змінною (ціни на нерухомість) і пояснювальними змінними, які можуть пояснити динаміку залежної змінної.

Часовий ряд – це порівняння спостережень добре структурованих даних, отриманих шляхом повторних вимірювань протягом певного часу. Дослідження канадського індексу цін на нерухомість, де порівнювались результати *ARIMA* з більш економетричними передовими моделями, такими як моделі авторегресії вектора, моделі спільної інтеграції та виправлення помилок показало, що прості моделі, такі як *ARIMA*, більш ефективні порівняно зі складнішими моделями [5]. Проводилося дослідження з використанням *ARIMA*, *GARCH* і однофакторних моделей перемикавання режимів для прогнозування ринку нерухомості в різних частинах США. Використовувалися дані повторних транзакцій на рівні штату для Каліфорнії, Флориди, Массачусетса, Огайо та Техасу. Річні темпи зростання щоквартально розраховувалися з кожного з цих індексів з 1979 (1 квартал) до 2001 (4 квартал). Дослідження показало, що моделі *ARIMA*, як правило, більше підходять для прогнозів поза вибіркою [3]. В Ірландії, коли проводили дослідження ринку нерухомості застосували моделі звичайних найменших квадратів (*OLS*), *ARIMA* та Vector Autoregressive (*VAR*), використовували щоквартальні дані з 1978 по 2003 рік. Виявилось, що модель *ARIMA* має кращі можливості прогнозування, ніж модель за період 1998–2001 рр. [6].

Однією з головних цілей аналізу часових рядів є прогнозування майбутніх значень ряду. Повільне і регулярне зростання на рівні серії називається трендом. Циклічні закономірності важливі, особливо в контексті управління нерухомістю. Дослідження ринків показують, що

існують чіткі докази існування коротких циклів серед 117 передмість Брісбена в Австралії протягом 1957-99 років. Спочатку треба видалити тенденцію в даних, перш ніж проаналізувати будь-які циклічні цінові моделі за допомогою спектрального аналізу даних кожного передмістя. Було виявлено, що середній рік циклу для всіх 117 передмість становить приблизно 9,95 року з відповідним стандартним відхиленням 1,97 року [7]. Кількість короткотермінових циклів становила 56 із 117 передмість, а діапазон років циклу – 7-14 років. Найпростішим класом моделей часових рядів, які можна розглянути, є процес ковзного середнього [4]. Нехай  $u_{t(t=1,2,\dots,n)}$  буде процесом білого шуму з математичним сподіванням  $E(u_t) = 0$  і дисперсією  $D(u_t) = \sigma^2$ . Тоді

$$y_t = \mu + u_t + \theta_1 u_{t-1} + \theta_2 u_{t-2} + \dots + \theta_q u_{t-q} \quad (1)$$

– модель ковзного середнього порядку  $q$ , позначена як  $MA(q)$ , і  $\theta$  оцінка параметрів для кожного відставання до  $q$ .

Модель ковзного середнього - це така лінійна комбінація процесів білого шуму, що  $y_t$  залежить від поточних та попередніх значень терміну порушення білого шуму. Авторегресивна модель - це модель, у якій поточне значення змінної,  $y_t$ , залежить лише від значень, які змінна приймала в попередні періоди, плюс помилка. Авторегресивна модель порядку  $p$  [2], позначена  $AR(p)$ , може бути подана як

$$y_t = \mu + u_t + \tau_1 u_{t-1} + \tau_2 u_{t-2} + \dots + \tau_p u_{t-p}, \quad (2)$$

де  $\mu$  – константа,  $\tau_1$  – оцінка відставання першого параметра із  $y_t$  та  $\tau_2$  – оцінка відставання другого параметра із  $y_t$ ,  $u_t$  параметр збудження білого шуму. Поєднавши  $AR(p)$  і  $MA(q)$  моделі одержуємо модель  $ARMA(p, q)$ . Така модель стверджує, що поточне значення деякого ряду лінійно залежить від його власних минулих значень плюс комбінація поточних і минулих значень доданків помилки білого шуму.

Ідентифікація порядку  $p$  і  $q$   $ARMA(p, q)$  може бути виконана одним із двох методів; автоматичний вибір або методологія Бокса-Дженкінса [2]. Для автоматичного вибору використовують інформаційний критерій Акаїке ( $AIC$ ) та байєсівський інформаційний критерій ( $BIC$ ). Критерії  $AIC$  і  $BIC$  будуються на основі інформаційної ентропії, тобто вони пропонують відносну оцінку втраченої інформації, коли дана модель використовується для представлення поведінки. Оскільки можна збільшити ймовірність, додаючи параметри, і  $BIC$ , і  $AIC$  вирішують цю проблему, вводячи штрафний термін для кількості параметрів у моделі. Однак цей термін

більший для  $BIC$ , ніж для  $AIC$ . Проблема з критеріями  $AIC$  і  $BIC$  полягає в тому, що вони обробляють дані як чорний ящик і не пропонують жодного уявлення про процес генерування даних.

Альтернативою критеріям відбору моделі є методика Бокса-Дженкінса, яка базується на моделі і ідентифікації шляхом дослідження автокореляційних функцій  $ACF$  і часткових автокореляційних функцій ( $PACF$ ). Автокореляція випадкового процесу описує кореляцію між значеннями процесу в різні моменти часу, як функцію відставання в часі. Нехай  $X$  представляє ціну нерухомості в момент часу  $t$ . Тоді  $X_t$  - це значення, отримане в момент часу  $t$ . Припустимо, що процес має визначені значення для середнього  $\mu$  і дисперсії  $\sigma^2$ . Тоді визначення автокореляції між часами  $t$  і  $t - s$ , де  $s$  є часовим відставанням, виглядає наступним чином [2]:

$$R(t, t - s) = \frac{E[(X_t - \mu)(X_{t-s} - \mu)]}{\sigma^2}, \quad (3)$$

де  $E$  - оператор математичного сподівання. Якщо функція  $R$  чітко визначена, її значення має лежати в діапазоні  $[-1, 1]$ , причому 1 означає ідеальну позитивну кореляцію, а  $-1$  — ідеальну негативну кореляцію.

Функція часткової автокореляції ( $PACF$  позначається  $\tau_{kk}$ ) вимірює кореляцію між спостереженням  $k$  періодів тому і поточним спостереженням після регулювання спостережень із проміжними лагами (тобто всіма лагами менше  $k$ ). Наприклад,  $PACF$  для затримки 3 вимірює кореляцію між  $y_t$  і  $y_{t-3}$  після контролю за ефектами  $y_{t-1}$  і  $y_{t-2}$ .

Методологія Бокса-Дженкінса, показує, як різні моделі функцій  $ACF$  і  $PACF$  можуть допомогти у визначенні початкової моделі  $ARMA(p, q)$ . Потім параметри моделі оцінюються з використанням максимальної ймовірності за допомогою алгоритму Берндта-Холла-Холла-Хаусмана. Алгоритм Берндта-Холла-Холла-Хаусмана є алгоритмом чисельної оптимізації, подібним до алгоритму Ньютона-Рафсона, він замінює спостережувану негативну матрицю Гессена зовнішнім добутком градієнта. Ця апроксимація заснована на рівності інформаційної матриці і тому дійсна лише при максимізації функції правдоподібності.

Загальна значущість моделі перевіряється за допомогою  $F$ -критерію Фішера. Потім залишки моделі перевіряються на відсутність шаблонів у їхніх  $ACF$  та  $PACF$ . Крім того,  $Q$ -тест Лjunga-Бокса застосовується для перевірки значущості накопичення відставання  $ACF$  до певної граничної точки [8]. Після того, як модель була проаналізована щодо відсутності шаблонів  $ACF$  і  $PACF$ , до залишків застосовуються нормальні імовірнісні тести, такі як  $W$ -тест Шапіро-Уїлка (1965) і  $Q - Q$  тест Себера (1977). Якщо

модель показує серйозні ознаки невідповідності таким припущенням, як відсутність *ACF* і *PACF* в залишках, а також нормальність залишків, весь процес ідентифікації моделі повторюється знову, поки не буде досягнута задовільна оптимальність моделі.

## Висновки

У даній роботі розглянуто підходи для побудови системи аналізу поведінки ринку нерухомості, запропоновано використати різні підходи для дослідження ринку, щоб досягнути кращої точності. Встановлено, що для отримання прогнозу потрібної точності, потрібно комбінувати різні підходи моделювання та прогнозування, для валідації точності та адекватності прогнозу. Метою майбутніх досліджень є проведення експериментів, завдяки яким запропоновані моделі будуть уточнюватися та удосконалюватися.

## Література

1. Brooks, C., and Tsolacos, S. (1999), "The impact of economic and financial factors on UK property performance", *Journal of Property Research* 16(2), 139-52 p.
2. Box, George and Jenkins, Gwilym (1970), "Time series analysis: Forecasting and control".
3. Crawford, G. and Fratantoni, M. (2003), "Assessing the forecasting performance of regime-switching, ARIMA and GARCH models of house prices", *Real Estate Economics*, 31, 223-43 p.
4. Brooks, C., and Tsolacos, S. (2010), "Real Estate Modelling and Forecasting"
5. Vishwakarma, V. K. (2013), "Forecasting Real Estate Business: Empirical Evidence From The Canadian Market", *Global Journal of Business Research*, Vol.7, No.3.
6. Stevenson, Simon. (2007), "A comparison of the forecasting ability of ARIMA models", *Journal of Property Investment & Finance*, 25(3), 223-240.
7. Reed, Richard (2002), "The Cyclical Nature of Residential Housing Markets", Paper presented in the Eighth Annual Conference of Pacific Rim Real Estate Society, Christchurch, 21-23 January.
8. Ljung, G. M. and Box, G. E. P. (1978), "On a measure of lack of fit in time series models". *Biometrika* 65 (2): 297-303
9. Воронін, В. О. Аналітика ринку нерухомості: методологія та принципи сучасної оцінки: монографія / В. О. Воронін, Е. В.

- Лянце, М. М. Мамчин . – Львів: видавництво «Магнолія 2006», 2014. – 304 с.
10. Тенденції ринку нерухомості України. Реалії та прогнози. Криза 2008-20?? / [В. О. Воронін , В. В. Галасюк , А. С. Гусельников та ін.]. – К.: АртТек, 2009. – 300 с.
  11. Тенденції ринку нерухомості України: реалії та прогнози. 2007-2013: [монографія] / [В. М. Шалаєв , О. І. Драпіковський, І. Б. Іванова та ін.]; [за ред. О. І. Драпіковського, І. Б. Іванової]; УВЕКОН. – К.: Арт Економі, 2012. – 240 с.
  12. Тарасевич, Е. И. Современные принципы анализа рынка недвижимости / Е. И. Тарасевич// Вопросы оценки. – 1999. – № 3. – С. 27-35.

УДК 62-791.4:004.912: 004.855.5

Д.т.н., проф. Чертов О. Р., студент Братун А. Ю.

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

## СИСТЕМА ВИЯВЛЕННЯ ТА РОЗПІЗНАВАННЯ НОМЕРНИХ ЗНАКІВ ВАГОНІВ ПОТЯГУ НА ВІДЕОПОТОЦІ ЗА ДОПОМОГОЮ НЕЙРОННИХ МЕРЕЖ

### Abstract

**Oleg Chertov, Prof., ScD; Andrii Bratun, student**

*The article describes a system for detecting and classifying objects based on convolutional neural networks that use layers of convolutional operations to search for objects. Data markup formats for creating object recognition systems are discussed. The most accurate methods of solving this problem are proposed. Ways of further research are also indicated.*

### Вступ

Вантажні залізничні перевезення є більш рентабельні, ніж автомобільні й не так залежать від природних факторів, як водні (наявність річки чи каналу, відсутність криги тощо). Раціональна організація залізничних перевезень потребує розв'язання ряду специфічних задач, до яких можна віднести відстежування вагонів та їх маршрутів. Особливо важливою ця задача є для таких підприємств, як вугільнопереробні, деревообробні, логістичні та виробничі [1].

Автоматизування перевірки номерів, підрахунок вагонів у складі потягу, виявлення вагонів, які проїжджають неправильним маршрутом, порівняння загальної маси складу з допустимою та корегування логістичних маршрутів значно полегшить та прискорить рутинну роботу людини.

Також компанії, які орендують вагони та склади, матимуть змогу отримати чітку аналітику рухів орендованих вагонів та мінімізувати час простою. В даному дослідженні розглядаються методи побудови нейронних мереж, які дозволяють виявляти та класифікувати об'єкти на відеопотоці для автоматизованого контролю залізничного транспорту [2].

## Постановка задачі

Задача полягає у розробці системи, яка б виконувала автоматизований контроль залізничного транспорту для кількісного та номерного обліку вагонів та вагового контролю вантажу, що перевозиться ними.

Система повинна отримувати на вхід дані з відеокамер, знаходити положення номеру на вагоні, розпізнавати цифри, звіряти знайдені номери з номерами, які знаходяться у базі даних.

## Огляд існуючих методів

Були розглянуті такі методи виявлення (detection) та класифікації об'єктів відеопотоку.

*Метод виявлення за допомогою боксів (bounding box detection model)* [3]. Даний метод використовується як базова модель, яка є згортковою нейронною мережею, що має щільний вихідний шар, який прогнозує координати знайденого боксу (див. Рис. 1). Зазначений метод використовує формат розмітки, який називається darknet. Побудова розмітки виконується таким шляхом: для кожного фрейму створюється окремий текстовий файл, який містить в собі рядки, де показано що за об'єкт на фреймі та його координати. Зазвичай, координати використовуються у відносних величинах, таких як центр по осі  $x$ , центр по осі  $y$ , висота та ширина боксу, але також можуть бути використані координати нижнього лівого та верхнього правого пікселя.

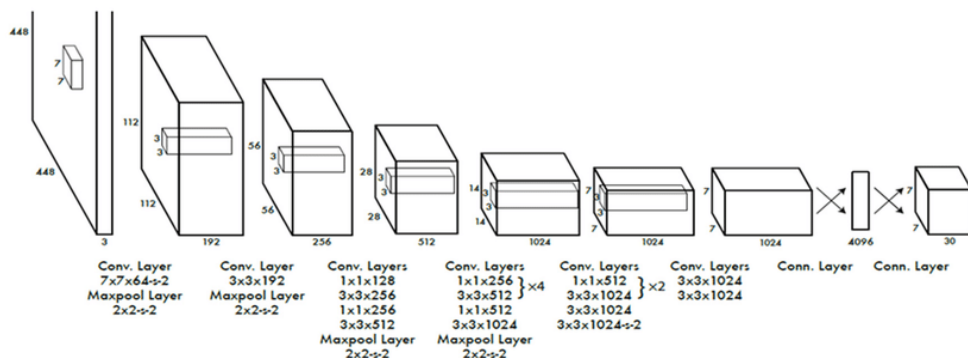


Рис. 1. Метод виявлення за допомогою боксів [4]

*Метод виявлення за допомогою сегментації (Segmentation model)* [5]. Даний метод також є згортковою нейронною мережею, яка має щільний вихідний шар, але він прогнозує належність кожного пікселю до якогось відповідного класу (див. Рис. 2). Зазначений метод використовує формат розмітки yaml. Побудова розмітки виконується таким шляхом: для кожного фрейму створюється окремий yaml-файл, який містить в собі



послідовності пікселів, що належать до окремого класу. Послідовності вказуються початковим та кінцевим пікселем, рахунок починається зліва направо та зверху вниз. Таким чином, модель сегментації надає кожному пікселю якийсь окремий клас. Під час навчання пікселі, які не відмічені на розмітці, вважаються класом фону.

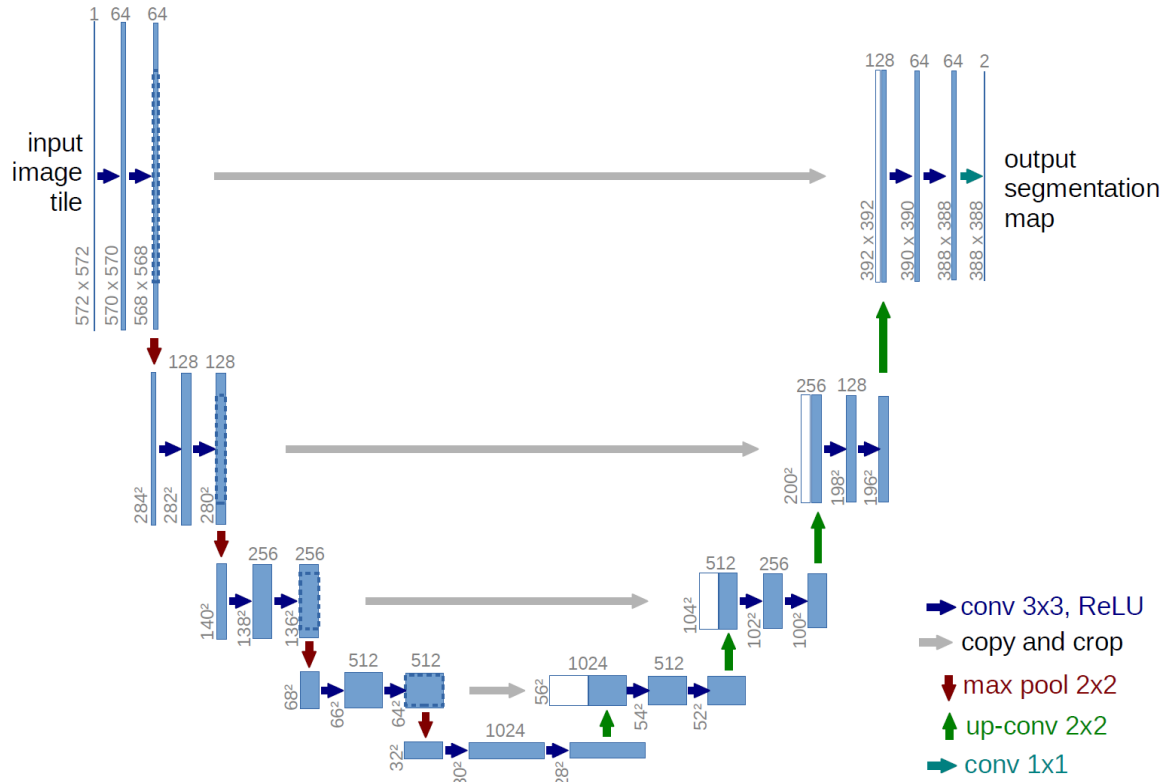


Рис. 2. Метод виявлення за допомогою сегментації [6]

*Inception v3* [7]. Цей метод використовується для класифікації об'єктів на фреймі. Його основою є згорткові шари, за якими йдуть шари зменшення (див. Рис. 3). Кінцевим шаром у випадку багатокласової класифікації є щільний з кількістю нейронів, яка дорівнює кількості класів, та функцією активації softmax. У випадку бінарної класифікації використовують один нейрон з функцією активації sigmoid, який прогнозує ймовірність належності класу до позитивного.

*Resnet v3* [9]. Даний метод належить до групи методів, що на відміну від попереднього, використовує зв'язки для пропуску декількох шарів (див. Рис. 4). Такий підхід значно покращує швидкість мінімізації функції втрат та прискорює навчання моделі.

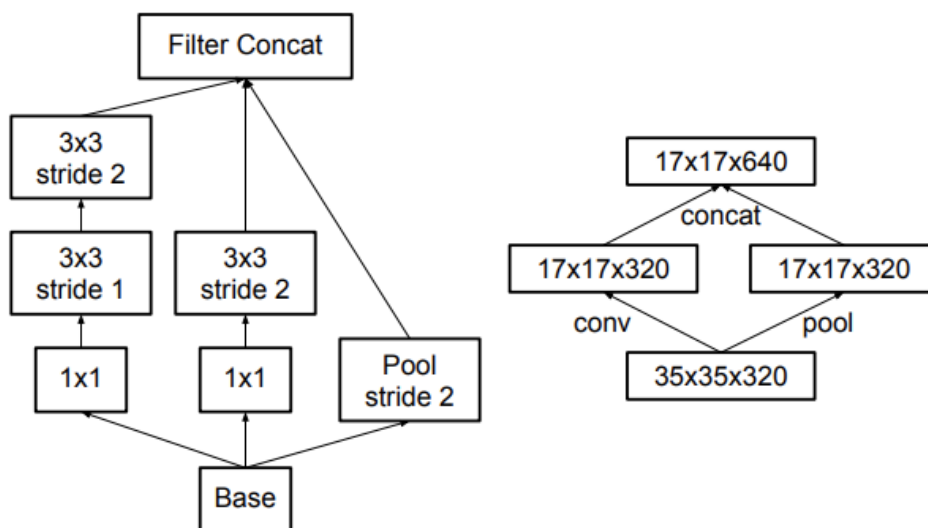


Рис. 3. Архітектура нейронної мережі Inception v3 [8]

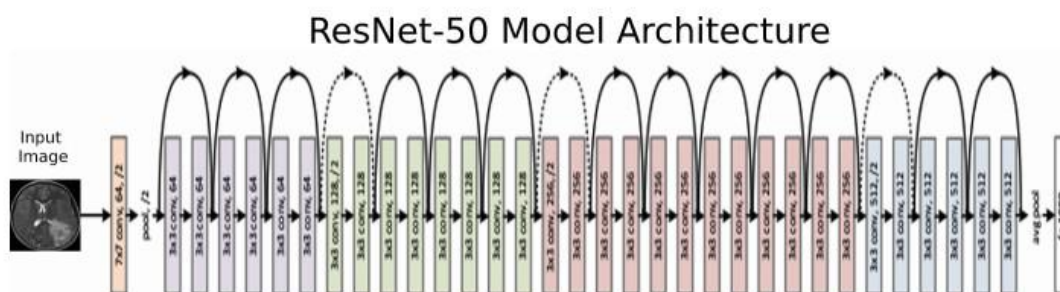


Рис. 4. Архітектура нейронної мережі Resnet v3 [9]

Для кількісного порівняння розглянутих методів було проведено їх тестування на зібраному власноруч наборі даних, що складається з 6 відеофайлів загальною довжиною 1 година. Кожен файл містить відеозаписи руху потягів, які проїжджають повз камеру вдень та вночі в літню пору року. Кількість вагонів складає 120 (75 вантажних та 45 цистерн). Навчання відбувалося на 70% даних та тестувалося на залишкових 30%. Метрикою оцінювання є mAP (mean Average Precision), яка показує середню точність для показника повноти (recall) в діапазоні від 0 до 1. Іншою оцінкою виступав показник FPS (frame per second), що означає кількість кадрів, оброблених за секунду.

## Порівняння розглянутих методів

Метод виявлення	Метод класифікації	mAP	FPS
Bounding box	<i>Inception v3</i>	97,3	53
<b>Bounding box</b>	<b><i>Resnet v3</i></b>	<b>97,7</b>	37
Segmentation	<i>Inception v3</i>	96,8	<b>75</b>
Segmentation	<i>Resnet v3</i>	97,1	44

**Опис системи**

Проаналізувавши існуючі методи, можна зробити висновок, що метод виявлення за допомогою боксів з класифікуючим модулем Resnet v3 працює з найкращою точністю, але з найменшим показником FPS, проте сучасні ресурси дозволяють працювати з показником FPS меншим за 15, тому в нашому випадку цим показником можна знехтувати.

Запропоновано побудувати систему, яка складається з таких окремих модулів та API для їх взаємодії:

- Попередня обробка даних. Моделі пошуку об'єктів насправді працюють лише із зображеннями, тому вхідний потік відео потрібно для початку розбити на фрейми. Далі, для того щоб навчати модель, над фреймами потрібно виконати розмітку, яка дозволить вказати моделі що саме потрібно вивчити та розпізнавати в майбутньому. Оскільки збір даних та їх розмітка - це найскладніший та найдорожчий етап побудови будь-якої системи, запропоновано зробити перетворення фреймів (augmentation). До перетворення фреймів відносяться: дзеркальні відображення фреймів, їх поворот, висвітлення або затемнення, вирізання випадкових регіонів, розмітність, додавання погодних умов. Під час перетворення фреймів, які пов'язані зі зміною розміру, важливо також змінювати файл розмітки, щоб забезпечити відповідність об'єкта і його положення на фреймі.

- Модель на основі виявлення боксів. Даний модуль використовує розмічені дані для навчання згорткових нейронних мереж, які надалі будуть виявляти та класифікувати об'єкти. Основною складовою моделей виявлення є згортковий шар (convolution layer), за яким слідує шар зменшення (pooling layer). Після декількох таких компонентів

використовують щільний шар (dense layer), який прогнозує координати об'єкта.

- Модель класифікації Resnet v3. Зазначена модель бере участь у класифікації знайдених цифр. Наявність даного модуля в алгоритмі виправдана тим, що знайдені номери потрібно класифікувати для визначення складу номера. Модель класифікації складається з шарів згорток, за якими слідує шар зменшення, але самі шари зв'язані не лише послідовно один за одним, а також мають додаткові зв'язки з наступними (skip connections) для покращення оптимізації функції втрат за допомогою методу зворотного поширення (backpropagation). Останнім шаром використовується щільний, який за допомогою функції активації softmax виконує класифікацію цифр від 0 до 9.

- API для інтеграції з джерелами відеопотоку. Сервіс буде отримувати фрейми з відеопотоку через HTTPS-зв'язок, виконуватиме знаходження та розпізнавання номерних знаків за допомогою розробленої моделі нейронної мережі та зберігатиме результати до бази даних.

## **Висновки**

У даній роботі було розглянуто методи побудови системи знаходження та розпізнавання номерних знаків вагонів потягу на відеопотоці за допомогою нейронних мереж, запропоновано методи детекції та класифікації, наведено приклади розмітки даних для навчання вищевказаних нейронних мереж.

Розпізнавання номерних знаків вагонів дозволить здійснювати автоматичний контроль номерного обліку вагонів та ваговий контроль вантажу.

Завдяки комбінації методу виявлення боксів та згорткової нейронної мережі Resnet v3, яка виконує класифікацію номерних знаків, було досягнуто точності розпізнавання, що дорівнює 97,7 при роботі з відеопотоком в 37 кадрів за секунду. Це дозволяє ефективно вирішити на практиці задачу розпізнавання номерних знаків потягу на відеопотоці.

Метою майбутніх досліджень є проведення натурних експериментів для оптимізації ряду гіперпараметрів розробленої моделі.

## **Література**

1. Льюис К. Д. Методы прогнозирования экономических показателей / Льюис К.Д. // М.: Финансы и статистика, 2013. – С. 68-70.
2. Паклин Н. Б. Бизнес-аналитика: от данных к знаниям / Паклин Н.Б., Орешков В.И. // Учебное пособие. 2-е изд., Питер, 2012. – С. 569-570.

3. Training Multi-Object Detector by Estimating Bounding Box Distribution for Input Image. URL: <https://arxiv.org/pdf/1911.12721v4.pdf> (access date: 18.10.2021).
4. You Only Look Once: Unified, Real-Time Object Detection. URL: <https://arxiv.org/pdf/1506.02640.pdf> (access date: 18.10.2021).
5. U-Net: Convolutional Networks for Biomedical Image Segmentation. URL: <https://arxiv.org/pdf/1505.04597.pdf> (access date: 18.10.2021).
6. Image Segmentation Using Deep Learning: A Survey. URL: <https://arxiv.org/pdf/2001.05566.pdf> (access date: 18.10.2021).
7. Rethinking the Inception Architecture for Computer Vision. URL: <https://arxiv.org/pdf/1512.00567.pdf> (access date: 18.10.2021).
8. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. URL: <https://arxiv.org/pdf/1602.07261.pdf> (access date: 18.10.2021).
9. Deep Residual Learning for Image Recognition. URL: <https://arxiv.org/pdf/1512.03385.pdf> (access date: 18.10.2021).

Д.т.н., професор Чертов О. Р., студент Мілев М. С.

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

## МЕТОД ВИЯВЛЕННЯ ТИПОВИХ ПРЕДСТАВНИКІВ КЛАСУ В РОЗМІЧЕНИХ ДАНИХ

### Abstract

Oleg Chertov, Prof., Sc.D.; Maksym S. Milev, student

*Method of identifying typical representatives of a class in the labeled data*

*This article discusses an algorithm and its applications for identifying typical representatives of a labeled sample. To solve the problem, classical machine learning methods such as decision trees and clustering were used. Approaches that allow using the algorithm to search for typical representatives among pictures are also considered. This solution can benefit the business in various ways, such as identifying focus groups, researching the target audience, and so on.*

### Вступ

На сьогоднішній день глибоке дослідження історичних даних є необхідним ключем до прийняття правильних рішень при веденні бізнесу. Одним з видів таких досліджень є знаходження типових представників певних груп та шаблонів їх поведінки. Така задача не має поширених, універсальних алгоритмів рішення. Але з точки зору бізнесу розуміння типових представників групи клієнтів може бути дуже корисним. Наявність такого знання дає уявлення про фокус-групи і допомагає максимально ефективно змінювати свою політику, підлаштовуючись під користувачів. Зосередившись на роботі саме з типовими представниками класів, можна суттєво зменшити обсяг розрахунків при аналізі різноманітних маркетингових стратегій.

### Постановка задачі

Як вхідний набір даних береться розмічений масив, тобто вибірка, в якій кожен її об'єкт належить до певного класу. Задача полягає в тому, щоб знайти типових представників кожного класу, або зробити висновок, що певні класи не мають чітко виражених представників. В даному випадку типовість можна розцінювати як рівень впевненості в тому, що найбільш схожі на типового представника об'єкти будуть з високою

ймовірністю належати до того ж класу, що і типовий представник. Ця задача ускладнюється, коли існує більше одного типового представника певного класу, або коли дані представлені не в табличному виді, а, наприклад, малюнками.

### Опис набору даних

Як набір даних для тестування можливостей методу, що пропонується, було обрано класичний багатовимірний набір даних для задач класифікації під назвою “Іриси Фішера” [1]. Набір даних містить інформацію про 150 вимірювань ірисів трьох різних видів.

Для більш наглядної демонстрації роботи запропонованого методу було взято двовимірний зріз даних, проте алгоритм може бути застосований для задачі будь-якої розмірності.

Вимоги до даних: відсутність пропущених значень, наявність дискретної цільової ознаки, нормування усіх незалежних змінних.

### Опис алгоритму

Пропонуємий алгоритм виявлення типових представників класу в розмічених даних складається з трьох послідовних кроків:

- 1) розбиття простору даних на області, у яких переважає один з класів;
- 2) кластеризація представників домінуючого класу в термінальних вузлах дерева;
- 3) вибір типових представників у кожному з кластерів.

Першим кроком методу є розбиття простору даних на певні області, у яких переважає той чи інший клас. Виділення таких підвибірок з домінуючим класом реалізовано шляхом побудови дерева прийняття рішень. Для цього був використаний відомий алгоритм CART, що детально описаний в [2].

Дерево CART — це бінарне дерево прийняття рішень, яке утворюється шляхом багаторазового поділу вузлів, починаючи з кореневого, що містить весь набір для навчання. На кожному кроці побудови дерева правило, сформоване у вузлі, ділить наявну множину прикладів на дві частини — ту, в якій виконується правило, та ту, в якій правило не виконується. Критерієм для поділу було обрано Information Gain (пониження ентропії) [2]:

$$H(x) = - \sum_{i=1}^n p_i \log_2 p_i$$

де  $p_i$  — це ймовірності, з якими випадкова величина  $x$  приймає незалежні випадкові значення  $x_i$ ;

$i$  — індекс кожного класу ( $i = 1, \dots, n$ ).

Причиною такого вибору є те, що зниження ентропії дозволяє відокремити певну підвибірку таким чином, щоб в ній утворився клас, який перевершує за кількістю представників інші класи, і при цьому всі члени підвибірки мають спільні ознаки. Глибина дерева в даному випадку є гіперпараметром, але емпіричним шляхом було зроблено висновок, що не слід призначати термінальним вузлом лист, в якому не має домінування певного класу над іншими.

Зі збільшенням кількості термінальних вузлів дерева, буде зменшуватися ентропія системи. Задача полягає в тому, щоб виявити — на якій стадії збільшення кількості листків дерева перестає приводити до суттєвого падіння ентропії. Вибір кількості термінальних вузлів пропонується здійснювати також емпіричним методом “Ліктя” (див. Рис. 1). На цьому рисунку прийнятними кандидатами є кількості від 6 до 14.

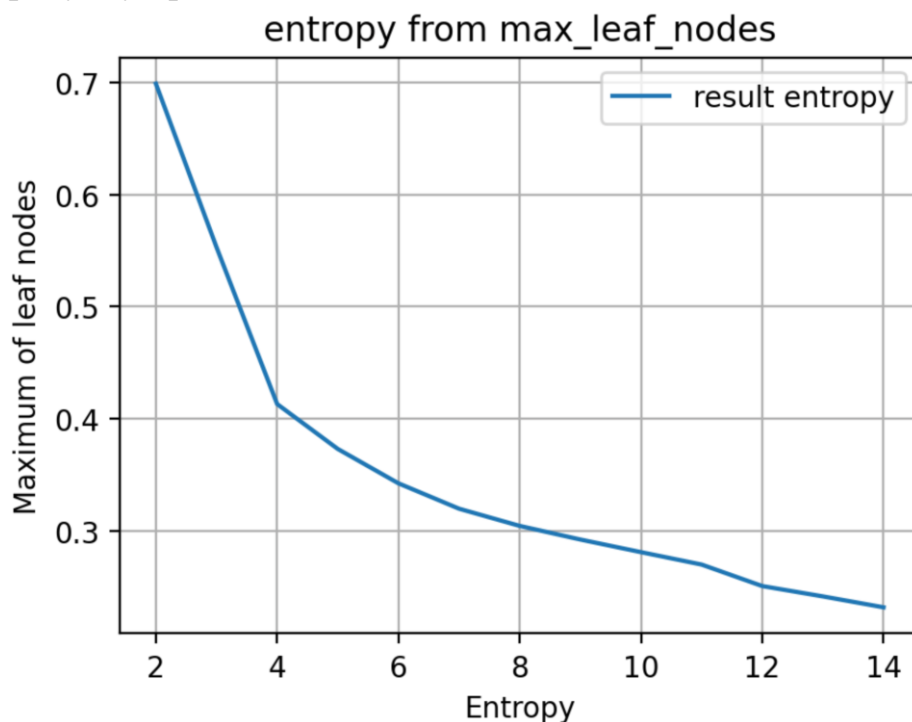


Рис. 1. Залежність ентропії системи від кількості листків дерева

Наступна мета — це пошук типових представників безпосередньо в термінальних вузлах. Оскільки можливі випадки, коли об’єкти з термінального вузла схожі за ознаками, які були враховані деревом, проте за іншими відрізняються, то є шанс знайти двох або більше різних типових представників певного класу.

Як видно з Рис. 2, дерево добре розділило початкову вибірку, але у кожному вузлі багато різних представників одного класу. Для виявлення типових представників слід знайти деяке усереднене значення у сформованих підвибірках. Однак через те, що в отриманих групах дані



можуть мати непередбачувану структуру, просте усереднення може призвести до втрати важливої інформації.

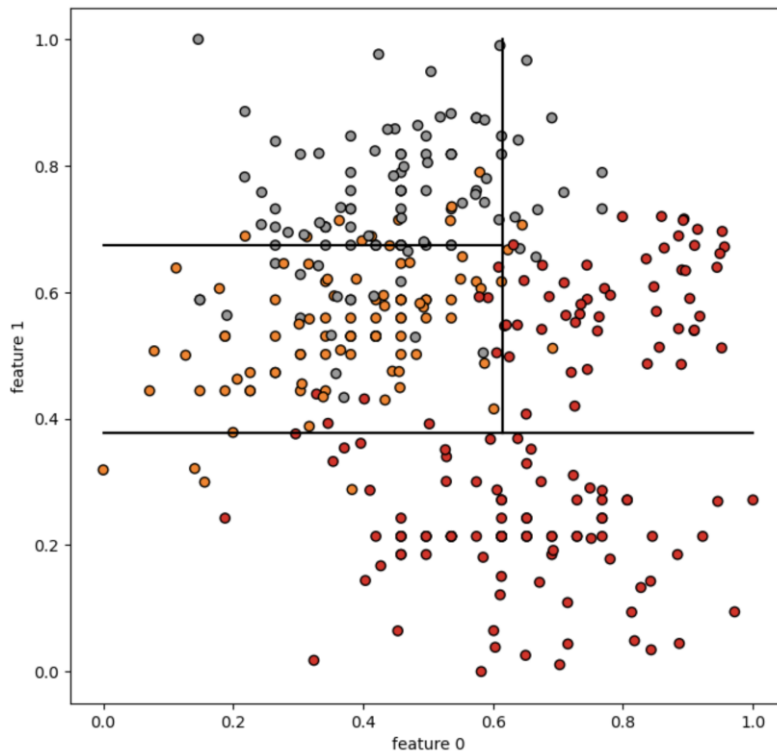


Рис. 2. Початкова вибірка розбита на області з низькою ентропією за допомогою дерева прийняття рішень

Зважаючи на це, другим кроком алгоритму є застосування кластерного аналізу до представників домінуючого класу в термінальних вузлах дерева. Як алгоритм кластеризації був обраний K-Means++ [3].

Перевагою K-Means++ є те, що вибір центроїдів кластерів відбувається не випадковим чином, а на оптимальній відстані один від одного, що в загальному випадку дозволяє виявити більше різних представників.

Кількість кластерів визначається автоматично алгоритмом DBE, який заснований на існуючому алгоритмі для візуальної оцінки кластерної тенденції (VAT) набору даних [4]. За результатами кластерного аналізу можемо перейти до останнього кроку методу.

На третьому кроці відбувається вибір типових представників у кожному сформованому кластері. Існує два способи здійснити цей вибір:

- 1) В якості типових представників обираємо об'єкти, найближчі до центроїдів своїх кластерів. На рисунку 3 такі об'єкти проілюстровано синіми колами;
- 2) Більш складний підхід до відбору типових представників реалізується в три етапи. Перш за все обчислюємо центроїди сформованих кластерів. Далі знаходимо центроїди кожної

підвибірки, утвореної на етапі побудови дерева рішень. Третім кроком розраховуємо відстань між центроїдами кластерів та центроїдами підвбірок, в яких ці кластери розташовані; після чого реалізуємо зміщення центроїд кластерів у протилежному напрямку від відповідних центроїд підвбірок. Найближчі точки до отриманих зміщених центроїд обираємо як типових представників. На рисунку 3 такі точки зображено рожевим кольором.

Додаткове зміщення у другому способі дозволяє шукати типових представників класу на більшій відстані від об'єктів інших класів, проте таким чином ми нехтуємо тим припущенням, що типовий представник є найбільш усереднений об'єкт серед інших в своїй групі.

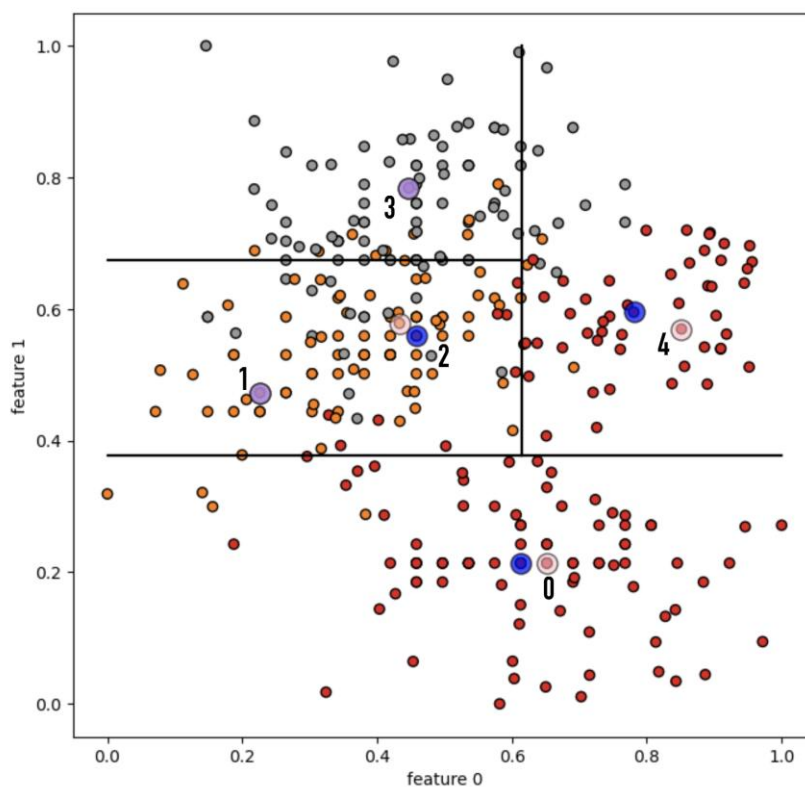


Рис. 3. Обрані типові представники класів

Фіолетові об'єкти — це ті, для яких сині та рожеві кандидати співпадають, тож вони однозначно визначені як типові представники своїх класів.

### Результати

Розроблений алгоритм дозволяє отримати типових представників для кожного з класів вибірки. Роботу запропонованого алгоритму було продемонстровано на тестовому наборі даних, для якого вдалося визначити 5 типових представників трьох різних класів.

Також для оцінки “типовості” кожного відібраного представника була обчислена метрика *isolation*: відношення кількості елементів класу представника, що входять в його кластер, до кількості всіх об’єктів які є членами кластеру (Рис. 4).

	0	1	2	3	4
<b>isolation</b>	0.958333	0.772727	0.775	0.870588	0.793651

Рис. 4. Значення *isolation* для кожного з представників з Рис. 3

## Висновки

Розроблено трикроковий алгоритм, який дозволяє визначити типових представників класів, які є в початковому наборі даних. Запропонований алгоритм було апробовано на класичному наборі даних “Іриси Фішера”, де вдалося виділити 5 типових представників з високим значенням метрики *isolation*.

У подальшому планується проведене дослідження доповнити такими роботами:

- додавання можливості опрацьовувати графічні дані, знаходити типових представників серед зображень;
- додавання етапу пост-обробки та підтримка “м’яких” (fuzzy) класів.

## Література

1. Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.
2. Jason M. Klusowski. Analyzing CART. Department of Statistics Rutgers University – New Brunswick Piscataway, NJ, USA, 8019.
3. Azad, M.; Chikalov, I., Hussain, S.; Moshkov, M. Entropy-Based Greedy Algorithm for Decision Trees Using Hypotheses. Entropy 2021, 23, 808.
4. Rukmi, A.; Iqbal, I. Using k-means++ algorithm for researchers clustering. AIP Conference Proceedings 1867, 020052 (2017).
5. Wang, L.; Leckie, C.; Ramamohanarao, K.; Bezdek, J. Automatically Determining the Number of Clusters in Unlabeled Data Sets. IEEE Transactions on Knowledge and Data Engineering, Vol. 21, Iss.3, 2009.

К.т.н., доцент Сирота С.В., студент Ковтун А.О.

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

## АВТОМАТИЗОВАНА СИСТЕМА ГЕНЕРУВАННЯ ТЕСТОВИХ ЗАПИТАНЬ НА ОСНОВІ МОДЕЛЕЙ ОБРОБКИ ПРИРОДНОЇ МОВИ

### Abstract

Sergiy V. Syrota, assoc. prof., PhD; Artem Kovtun, student,  
*Automated test questions generation system based on NLP*

*This paper concerns the task of automated test questions generation system creation, that leverages modern technologies of natural languages processing. Types of test tasks which creation process can be delegated to such an application. Methods, models and approaches that can be used to embody desired types of tasks. The ways for further research are proposed as well.*

### Вступ

Популярним способом перевірки або підтвердження знань були і залишаються тести. Створення тестів, а саме тестових запитань є рутинною роботою. На сьогодні існує багато спроб автоматизувати або частково автоматизувати процес створення верифікованих тестових завдань маючи на вході інформацію у вигляді тексту. Опис таких технологій наведено в [1]. Дана робота висвітлює спробу застосування сучасних бібліотек обробки природної мови для розв'язання задачі автоматизованої генерації тестових запитань.

### Постановка задачі

Задача полягає у побудові системи формування тестових запитань на основі заданих текстових даних, що представляють собою речення англійською мовою або тексти, що складаються з послідовності пов'язаних речень. Застосунок повинен генерувати тестові запитання до заданого тексту.

### Опис системи

Побудована система представляє собою веб застосунок з графічним інтерфейсом (рис.1) та пропонує генерування чотирьох типів тестових

завдань: «вставити пропущене», «відповідь по ключовому слову», «генерація питань по заданій кількості ключових слів», «так/ні».

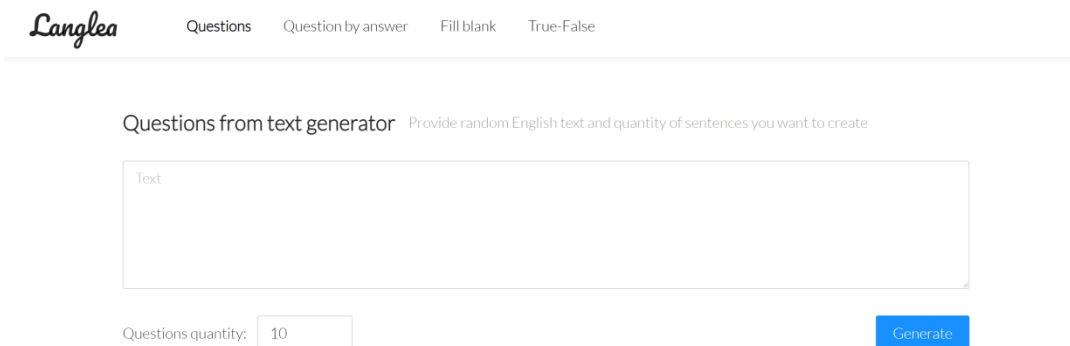


Рис. 1. Графічний інтерфейс створеного застосунку

### Тестове завдання «Вставити пропущене»

Даному розділу застосунку відповідає меню навігації «Fill blank». На вході очікується текст та кількість пропусків, які користувач бажає згенерувати. Система отримавши вхідні дані виконує вибірку потенційних слів, що будуть вилючені з використанням алгоритму RAKE (Rapid Automatic Keyword Extraction Algorithm) [2] – метод виділення ключових слів із тексту, базується на графовому представленні нормалізованого тексту (з видаленням стоп-слів і роздільників фраз, приведенням значущих слів до базових словоформ).

Далі виконується заміна отриманих слів або виразів на пропуски у початковому тексті та формується кінцеве завдання, що складається з відформатованого тексту та виділених слів, які необхідно вставити (рис.2).

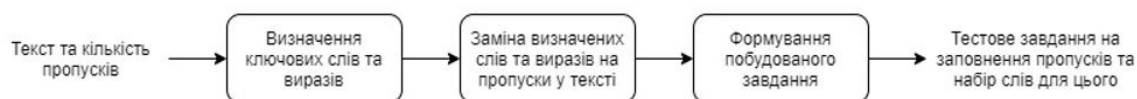


Рис. 2. Побудова завдання типу «вставити пропущене»

### Тестове завдання «За ключовим словом»

Даному розділу застосунку відповідає меню навігації «Question by answer». Система отримує на вхід речення, яке слугує контекстом для побудови питання, та ключове слово з контексту – відповідь на питання, що бажано згенерувати. Система спочатку нормалізує отриманий текст (видаляє стоп слова та зайву інформацію), за допомогою донавченої моделі Text-to-Text Transfer Transformer (T5) [3], що базується на мовній моделі

BERT (Bidirectional Encoder Representations from Transformers) [4], виконується формування питання за заданим контекстом та ключовим словом.

В кінці відбувається формування результату, що складається із згенерованого питання та відповіді – ключового слова (рис. 3).

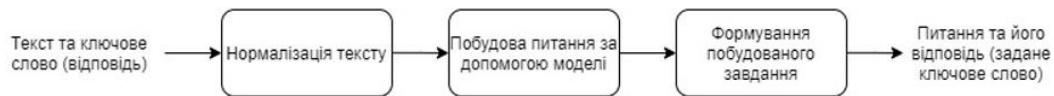


Рис. 3. Формування питання за ключовим словом

Під час тестування даної гілки системи на різних вхідних даних, було помічено, що система не в усіх випадках генерує коректні речення для заданого контексту, наприклад для речення: «Mike loves to play football» для ключового слова «football» система згенерує питання: «What sport does Mike love to play?», що є правильним питанням для відповіді у заданого реченні. Але для речення «Mike loves to play guitar» та «guitar» система також буде генерувати «What sport does Mike love to play?». Тобто у деяких випадках, система не завжди генерує коректне питання щодо заданого контексту. В такому випадку модель T5 потребує донавчання. Даний процес нагадує процедуру дресування домашньої тварини. Тобто складається враження, що необхідно удосконалювати систему стимулів і як позитивних так і негативних.

### Генерація питань за заданою кількістю ключових слів

Даному розділу застосунку відповідає меню навігації «Questions». Частина поєднує елементи двох описаних попередньо. На вхід подається текст та кількість питань, що необхідно згенерувати за заданим текстом. Як і у попередніх випадках виконується нормалізація тексту. За тим, за допомогою методу вибору ключових слів або фраз із тексту RAKE, що використовувався при побудові завдання «вставити пропущене» виконується вибірка слів та виразів для яких виконується побудова речень за отриманим контекстом на вході. Формується фінальний результат у вигляді множини пар: ключове слово, що було обрано з тексту та питання до нього. (рис. 4)



Рис. 4. Формування визначеної кількості питань за заданим текстом

### Формування завдання визначення «так/ні»

Даному розділу застосунку відповідає меню навігації «True-False». Єдиним вхідним параметром, що система очікує, є текст з якого будуть генеруватися речення «неправди». На першому етапі, як і у двох попередніх пунктах, виконується нормалізація вхідного тексту. За тим за допомогою методів синтаксичного розбору речень виділяється головна частина кожного речення. Далі з використанням моделей нейронних мереж OpenAI GPT-2 [5] та бібліотеки Tensorflow [6] виконується побудова неправдивих речень шляхом побудови заперечення вхідного речення.

Результатом виконання є набір пар речень: правдиве (початкове) та неправдиве (згенероване) (рис.5).



Рис. 5. Формування завдання визначення «так/ні»

Під час тестування обраного підходу для застосунку було отримано результати, що відрізнялись від очікуваних на етапі проектування та аналізу. Хоч система і генерує певне речення, яке є граматично правильним, воно у більшості випадків не відноситься до контексту заданого початковим текстом. Тобто не можна зі стовідсотковою ймовірністю казати, що таке речення є неправдивою. Наприклад для контексту:

*«BERT is a Natural Language Processing Model proposed by researchers at Google Research in 2018. When it was proposed it achieve state-of-the-art accuracy on many NLP and NLU tasks such as: General Language Understanding Evaluation? Stanford Q/A dataset SQuAD v1.1 and v2.0 ,Situation With Adversarial Generations. Soon after few days of release the published open-sourced the code with two versions of pre-trained model BERTBASE and BERTLARGE which are trained on a massive dataset. BERT also use many previous NLP algorithms and architectures such that semi-supervised training, OpenAI transformers, ELMo Embeddings, ULMFit, Transformers.»*

система для правдивого речення сформувала:

1. Правда із контексту – *«BERT is a Natural Language Processing Model proposed by researchers at Google Research in 2018.»*
2. Згенерована неправда – *«BERT is a Natural Language Data Manipulation and Synthesis platform based on Haskell.»*

що, так, не є правдою, але це відомо лише через те, що ми знаємо контекст за межами початково заданого, якщо опиратися лише на вхідні дані, то не можливо сказати правда це, чи ні.

## Висновки

При описі системи було виділено 4 типи тестових запитань, які вона може генерувати: вставити пропущене, формування питання за ключовим словом, генерація визначеної кількості питань за заданим текстом та завдання визначення «правда-неправда». Для кожного з них були описані вхідні дані, що очікуються, шлях яких вони проходять для отримання кінцевого результату, та моделі і методи, що використовуються у процесі роботи системи, а саме: для задачі вибору ключових слів – метод RAKE, для формування питань – модель T5 на архітектурі BERT та для створення «речень-неправди» модель нейронних мереж OpenAI GPT-2 та бібліотеки Tensorflow. Було описано недоліки роботи для окремих видів тестів, при їх генерації та наведено приклади.

Метою наступних досліджень є проведення експериментів та покращення запропонованих методів для збільшення точності роботи системи.

## Література

1. Syrota S., Kopychko S., Liskin V. (2020) Ontology-Based Approach for E-learning Course Creation Using Chunks. In: Hu Z., Petoukhov S., Dychka I., He M. (eds) Advances in Computer Science for Engineering and Education II. ICCSEEA 2019. Advances in Intelligent Systems and Computing, vol 938. Springer, Cham
2. Андраде М. та Валенсія Аю. «Автоматична вибірка ключових слів с наукових текстів» (1998 ) Bioinformatics 14(7),600–607.
3. Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J. Liu "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer" (2020)
4. Jia Tracy Shen, Michiharu Yamashita, Ethan Prihar, Neil Heffernan, Xintao Wu, Ben Graff, Dongwon Lee "MathBERT: A Pre-trained Language Model for General NLP Tasks in Mathematics Education" (2021)
5. "OpenAI releases curtailed version of GPT-2 language model". VentureBeat, Archived from the original on 18 December 2020
6. Dean Jeff and Monga Rajat "TensorFlow: Large-scale machine learning on heterogeneous systems" TensorFlow.org. Google Research. 2015.



**К.ф-м.н., доцент Третиник В.В., студентка Трішина А.О.**

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

## **МОДЕЛЬ ARIMA ДЛЯ ПРОГНОЗУВАННЯ ТРАФІКУ ВЕБ-СТОРІНКИ**

### **Abstract**

**Violeta V. Tretynyk, assoc. prof., PhD; Anna A. Trishyna, student**  
*Arima model for website traffic predictions*

*This article discusses the ARIMA model for traffic prediction. Step-by-step model building. With the help of this forecast, we can track how fast the page traffic is growing and what optimization parameters affect the web page.*

### **Вступ**

В наш час люди активно користуються різними веб-сайтами, але мережа насичена сайтами, тому не кожна людина може знайти окремий сайт, відвідуючи Інтернет. Для покращення ранжування сайту у пошуковій системі Google, необхідно оптимізувати веб-сторінку.

Для того щоб оптимізувати сайт, нам необхідно бачити різницю між початковим станом сайту та його кінцевим виглядом. Для відслідковування розвитку сайту ми можемо спостерігати за трафіком сайту, а саме за трафіком відвідування. Оскільки Google постійно оновлює свої алгоритми пошукової видачі, змінюються параметри оптимізації, а отже й падає трафік.

Для вирішення даної проблеми допоможе прогнозування трафіку відвідуваності, оскільки відслідковуються дані за допомогою трафіку на яким можемо побачити після яких змін, як поводить трафік видачі результатів. Отже, даний прогноз є актуальною темою, оскільки ми зможемо відслідковувати чи коректно налаштована оптимізація сторінки та які результати у нас будуть в майбутньому.

### **Постановка задачі**

Метою ранжування та підвищення трафіку сторінки є оптимізація сторінки для виведення сайту в перші позиції пошукової видачі. Для процесу розвитку на основі оптимізації веб-сторінки результатом буде

прогноз, що розроблений на даних, які отримані з самого початку розвитку сайту з подальшою оптимізацією.

Предметом дослідження є математичне та програмне забезпечення системи прогнозування трафіку веб сторінки на основі часових рядів.

Для розв'язку поставленої задачі використовувалися методи Бокса-Дженкінса.

Модель прогнозування отримує на вхід дані про кількість переглядів сторінки та дату, а на виході отримуємо прогнозне значення переглядів, яке нам покаже як розвивається сайт. Ціллю є збільшення трафіку за менший період часу.

### **Термінологія**

Autoregressive Integrated Moving Average (ARIMA) - інтегрована модель авторегресії та ковзного середнього.

### **Опис**

Для розвитку сторінки є багато критеріїв, які впливають на просування в пошуковій видачі Google. Перш за все необхідно оптимізувати сторінку за параметрами. Розглянемо параметри, які впливають на просування сторінки.

Дуже важливу роль відіграють ключові слова. За допомогою ключових слів ми зможемо просувати нашу веб сторінку. Для кожної сторінки повинні бути релевантні ключові слова. Також важливі налаштування Meta тегів, до яких входять title тег, тег description заголовков h1. Також на просування сторінки впливає наповнення сайту контентом. До контенту належать картинки, текст, відео. Великий вплив мають посилання та їх вага, наявність налаштованого файлу robots.txt та мапи сайту.

Важливо, щоб наш сайт видавав код сторінки 200 та не мав помилки 404. Також потрібно, щоб сайт не мав дублікатів та мав власний бренд. Важливим параметром є швидкість роботи сайту мобільної та десктопної версії.

Вище розглянуто мінімальний набір параметрів для оптимізації сторінки. За допомогою оптимізованого сайту швидко індексуються сторінки та збільшується трафік відвідуваності, але оскільки у пошуковій системі змінюється часто алгоритм, то треба постійно відслідковувати оптимізацію сайту, це ми можемо зробити за допомогою прогнозування. Коли сторінка оптимізована, відвідуваність сайту зростає, але коли змінюється алгоритм, відвідуваність сайту падає. Для відслідковуваності даної залежності необхідно будувати прогноз.

Існує багато методів для прогнозування трафіку відвідуваності веб-сторінки. Одним із таких методів є ARIMA для моделювання часових рядів. Даний метод визначає кореляцію між точками даних та враховує різницю значень [1].

Для даної моделі ARIMA характерні деякі властивості.

Необхідно, щоб вхідні дані для обраної моделі були одномірною послідовністю тому, що на основі попередніх значень, дана модель робить прогноз майбутніх значень.

Середнє значення та дисперсія не повинні змінюватися з часом, оскільки ряд даних є стаціонарним [2].

Модель Arima має вигляд:

$$y_t = \alpha_1 y_{t-1} + \dots + \alpha_p y_{t-p} + \varepsilon_t + \theta_1 y_{t-1} + \dots + \theta_q y_{t-q} \quad (1)$$

де  $p$  - порядок авторегресії;

$q$  - порядок ковзного середнього.

Модель ARIMA складається з 3 компонентів:

1. Авторегресія - термін AR - використовується для прогнозування наступного значення на основі попередніх значень. Позначається параметром "р" в даній моделі. Значення  $p$  визначається з використанням графіка ACF. ACF - це графік загальної кореляції між різними функціями запізнення, який має наступний вигляд рисунок 1.

2. Різниця - I - визначає скільки разів операція взяття різниць виконується на ряді, щоб зробити його стаціонарним.

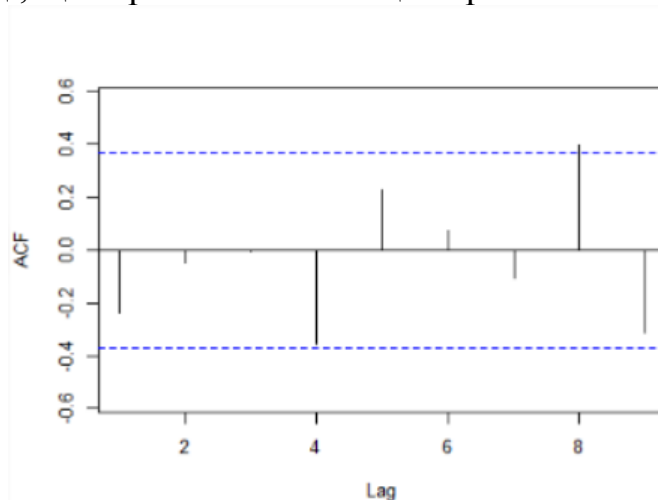


Рис. 1. Графік ACF [2]

3. Ковзне середнє - термін MA - за допомогою MA можна визначити кількість попередніх помилок прогнозу, які використовуються для прогнозування майбутніх значень та позначається параметром "q". Графік PACF використовується для визначення правильного значення  $q$ . PACF знаходить кореляцію залишків з наступним значенням лага.

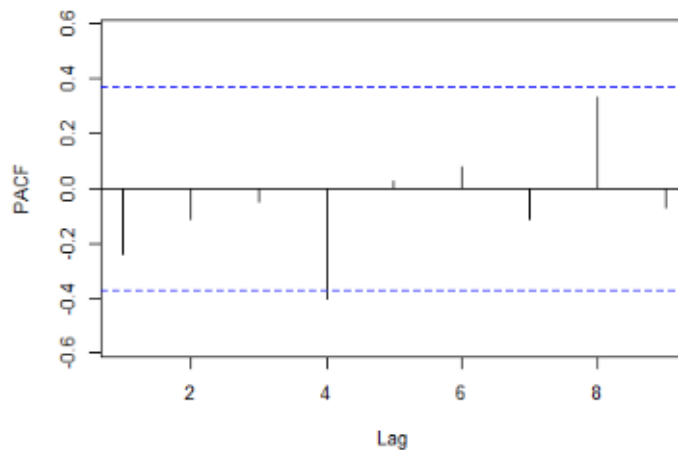


Рис. 2 Графік PACF [2]

Модель ARIMA має параметри-константи визначені як [4]:

- $p$  – порядок лагу (відставання) - кількість лагів спостережень, що входять у модель;
- $d$  – ступінь диференціації - кількість разів початкові дані були продиференційовані;
- $q$  – порядок ковзного середнього.

### Реалізація моделі ARIMA

Для побудови моделі необхідно зібрати дані та завантажити їх. Вхідними даними є дата та кількість переглядів веб сторінки користувачами за день.

	Date	Views
543	16-09-2021	295
544	17-09-2021	289
545	18-09-2021	291
546	19-09-2021	299
547	20-09-2021	283

Попередня обробка даних: залежно від набору даних буде визначено етапи попередньої обробки. Вони можуть включати створення тимчасових позначок, перетворення dtype стовпця дати/часу, перетворення ряду на одновимірний і т.д [1].

```
df.Date=pd.to_datetime(df.Date)
df=df.set_index('Date')
df.tail()
```

Date	Views
------	-------

2021-09-16 295  
 2021-09-17 289  
 2021-09-18 291  
 2021-09-19 299  
 2021-09-20 283

Перетворення ряду на стаціонарний. Щоб задовольнити вимоги алгоритму, необхідно зробити ряд стаціонарним. Цей етап включатиме перевірку стаціонарності та виконання необхідних перетворень.

Оцінити стаціонарність часового ряду можливо за допомогою графіку даних ACF. Якщо ряд стаціонарний, тоді на графіку ACF скоротиться до нуля швидко, тоді як ACF нестационарних даних буде зменшуватися повільно [4].

Також аналіз стаціонарності можна зробити за допомогою тесту Дікі-Фуллера.

Тест Дікі-Фуллера – це тест на одиничний корінь, який перевіряє гіпотезу про те, що  $\alpha = 1$  у наступному рівнянні моделі

$$y_t = c + \beta t + \alpha y_{t-1} + \phi \Delta Y_{t-1} + e_t \quad (2)$$

де  $\alpha$  - коефіцієнт першого відставання від  $Y$

$y(t-1)$  = відставання 1 часового ряду

$\Delta Y(t-1)$  = перша різниця ряду в момент часу (t-1)

Тест має ту ж нульову гіпотезу, як і тест на одиничний корінь. Тобто коефіцієнт за  $Y(t-1)$  дорівнює 1, що означає наявність одиничного кореня. Якщо не відкинуто, ряд вважається нестационарним [3].

Тест ADF розширює рівняння тесту Дікі-Фуллера, щоб увімкнути в модель регресійний процес високого порядку.

$$y_t = c + \beta t + \alpha y_{t-1} + \phi \Delta Y_{t-1} + \phi_2 \Delta Y_{t-2} + \dots + \phi_p \Delta Y_{t-p} + e_t \quad (3)$$

Результати тестування за Дікі-Фулером наступні:

test statistics	-1.024531
p-value	0.444212
lags used	7.000000
number of observation used	540.000000
critical value (%s) %key	-3.442518
dtype: float64	

Можна зробити висновок, що ряд стаціонарним.

Створення графіків ACF та PACF:

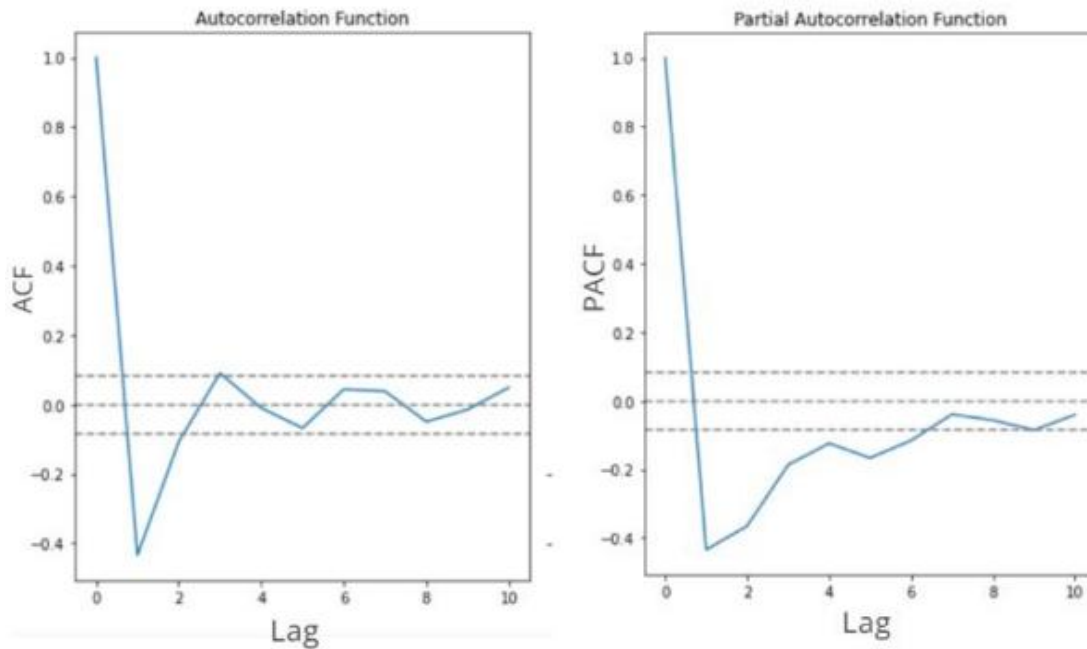


Рис.3 Графіки ACF та PACF

Обидва графіки демонструють деяку нескінченну поведінку, наприклад, експоненційне зменшення у величині коефіцієнтів, також можна помітити значний спад значень після лагу.

Результатом даного прогнозу є наступний графік, який показує прогнозовану кількість переглядів, на основі якої можна побачити чи добре оптимізований сайт:

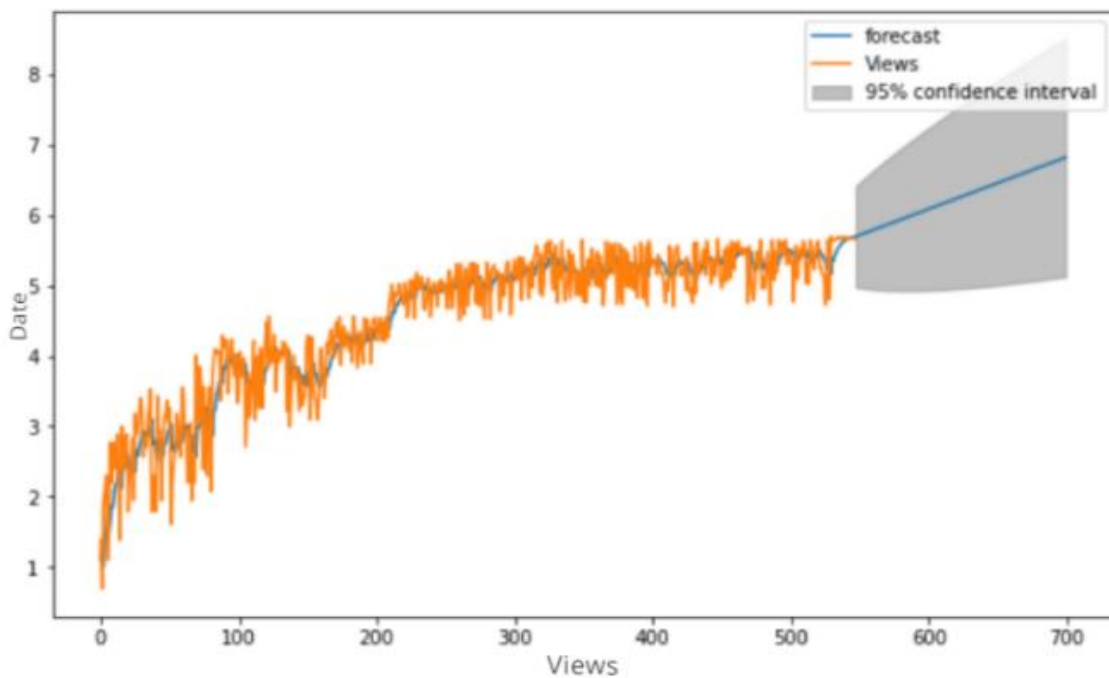


Рис. 4 Прогнозування за допомогою моделі ARIMA

## Висновки

У статті розглянута модель ARIMA, за допомогою якої був прогнозований трафік відвідуваності сайту. Таким чином ми можемо в подальшому проаналізувати як оптимізується сайт. Аналізуючи наскільки швидко зростає трафік після оптимізації веб-сторінки та які параметри впливають більше на трафік, ми можемо зрозуміти, які алгоритми поновив Google.

## Література

1. Дуброва, Т. А. Статистичні методи прогнозування. [Текст]/Т. А. Дуброва М.: ЮНІТІ-ДАНА, 2003. 206 с.
2. Канторович Г.Г. Аналіз тимчасових рядів [Текст] / Г. Г. Канторович. М.: Економічний журнал ВШЕ, №3, 2002.
3. Прогнозування та аналіз часових рядів. Методичні вказівки до практичних занять та самостійної роботи студентів спеціальності 051 «Економіка» освітня програма «Економічна кібернетика», «Економічна аналітика» / Укл.: Юрченко М. Є. – Чернігів: ЧНТУ, 2018. – 88 с.
4. Бокс Дж., Дженкинс Г. Анализ временных рядов. Прогноз и управление. Выпуск 1. – М.:Мир, 1974. – 406 с.

К.т.н., доцент Боярінова Ю.Є., студент Гнатенко В.Д.

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

## СПОСІБ ШТРИХОВОГО КОДУВАННЯ ТА ДЕКОДУВАННЯ ІНФОРМАЦІЇ

### Abstract

**Yuliia Y. Boiarinova, assoc. prof., PhD; Hnatenko Vitaliy, student**  
*Method of barcode coding and decoding of information*

*This paper describes the method of barcode encoding and decoding. Classical linear barcode algorithms do not provide the necessary level of compact placement, when encoding a large amount of characters. And 2d barcode algorithms are too overloaded and do not provide the required level of simplicity for decoding. The proposed method provides the needed level of simplicity, compactness and speed, and does not require any specific hardware.*

### Вступ

Разом з розвитком інформаційних технологій досить великого розповсюдження набули технології кодування інформації. Видів кодування та відображення даних створено безліч. Одним з найрозповсюдженіших є штриховий код. Штрих-код використовується всюди: від логістики до ідентифікації людини. Технологія штрихового кодування базується на розпізнаванні шаблонів, закодованих у штрихах та просторах визначених розмірів. Існує чимало способів кодування інформації у форматі штрих-коду, кожен з яких використовується для вирішення певної специфічної задачі.

Через збільшення вагової кодованої інформації основна низка існуючих способів або є надлишковими при їх використанні, або не підходить для рішення задачі з відображенням закодованої інформації в обмеженому просторі з великою щільністю даних. Для рішення проблеми було створено власний спосіб двовимірного штрихового кодування та спосіб зворотного кодування, для подальшого декодування видозмінених даних.

### Постановка задачі

Задача полягає в створенні способу двовимірного кодування інформації для відображення в умовах малої площі розміщення з великою щільністю даних з метою подальшої можливості швидкого декодування.



## **Термінологія**

*Штриховий код* – закодована інформація, яка представлена у графічному вигляді та друкується на поверхню, що дозволяє зберігати та зчитувати дані різними пристроями.

*1D штрих-код* – лінійне штрихове кодування, яке кодує інформацію таким чином, що зчитувальний пристрій ігнорує вертикальну складову і сканує лише горизонтальну частину.

*2D штрих-код* – двовимірне штрихове кодування, що зберігає інформацію не лише горизонтально, як лінійне кодування, але ще й вертикально.

*Stacked barcode* – вид двовимірного штрихового кодування, що складається з вертикально розміщених лінійних штрих-кодів і декодується лінійно.

## **Опис способу кодування**

Розроблений спосіб, задовольняючий усім критеріям, має досить просту специфікацію. Кожен символ закодований за допомогою унікальної комбінації смуг та пробілів, що складаються з чотирьох чорних і трьох білих смужок різної товщини. Ширина смужок знаходиться в діапазоні від 1 до 4 мінімальних одиниць товщини. Сума усіх довжин чорних і білих смужок повинна дорівнювати 13. При необхідності можна змінити кількість смуг, які кодують один символ, що дає змогу в подальшому розширити набір символів, які підтримує кодування.

Даний спосіб спроможний кодувати основні символи таблиці ASCII [1], а саме: починаючи з коду 32 (Space) і закінчуючи символом з кодом 127 (DEL). В залежності від кількості символів, висота штрихового коду може знаходитись в діапазоні від 1 до 50 строк, що дає змогу закодувати близько 850 символів. Якщо виникне потреба для збільшення кількості символів, можна прибрати штучну заборону на використання більше 50 строк кодування, тоді алгоритм теоретично може зберегти дані об'ємом до 2700 символів, але для зчитування та декодування штрих-коду потрібно буде більше обчислювальної потужності пристрою, що в результаті збільшить час на розпізнавання та обробку коду. Зазвичай використовується діапазон від 7 до 30 строк. На рис. 1 зображено штриховий код, в якому закодовано 290 символів. Він досить схожий на одновимірні штрих-коди, адже кодує символи за схожим чином, тому що дотримується стандарту GS1 [2].

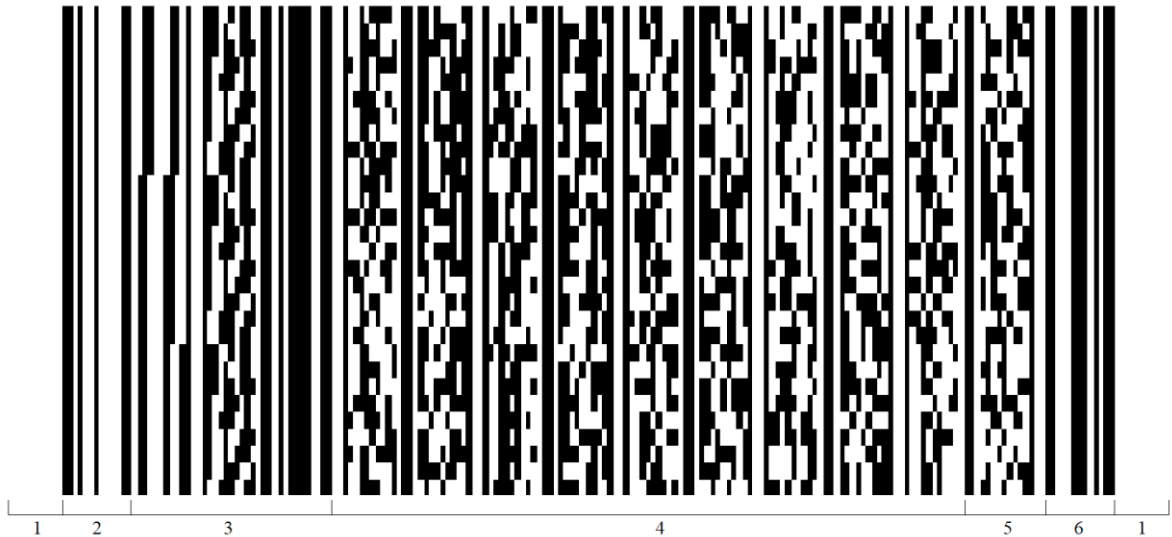


Рис. 1. Зображення штрих-коду

Сам код складається з таких зон:

1. Зона без даних або пуста зона. Використовується на початку та у кінці для більш простого розуміння про знаходження стартового та кінцевого символів;
2. Стартовий символ. Вказує на початок коду для полегшення зчитування;
3. Область додаткових даних до стовпчика. Зберігає дані для коректного зчитування строки;
4. Зона даних. Саме у цій зоні знаходяться закодовані дані;
5. Символ для перевірки коректності зчитування. В ньому зберігається сума усіх товщин чорних смуг в області даних;
6. Кінцевий символ. Вказує на кінець кодування штрихового коду.

### **Опис способу декодування**

Для декодування штрихового коду не потрібен спеціальний сканер, вистачить й звичайної камери смартфона. Для вхідних даних алгоритму достатньо лише фотографії, що дає змогу впровадження реалізації способу кодування на будь-якому сучасному смартфоні.

Саме декодування виконується в декілька етапів:

1. На вхід подається фотографія у будь-якому форматі (png, bmp, jpg тощо);
2. Зображення перебудовується у чорно-білий варіант та перетворюється на числовий масив даних, що були побудовані на основі яскравості кольорів пікселів, де значення 0 відповідає білому кольору, а значення 255 – чорному;

3. Відбувається перший етап сканування, а саме знаходження первинних координат штрихового коду, починаючи з середини зображення;
4. Якщо координати знайти не вдалось – проходимо по всьому зображенню в пошуках координат коду. Під час знаходження координат відбувається перехід до наступного пункту і закінчення пошуку;
5. Якщо координати знайшлись – відбувається перехід до верхньої частини штрихового коду і почергове сканування кожної строки. Після вдалого сканування строки відбувається перевірка інформації на відповідність за допомогою даних, що зберігаються в зонах 3 та 5, зображених на рис. 1. Якщо дані були пошкодженні, то сканування переходить до наступного ряду пікселів;
6. Уся оброблена інформація перевіряється і, якщо жоден з рядків не був пропущений під час сканування, повертається текстовий результат.

### **Висновки**

Штрихове кодування стало невід’ємною частиною нашого сучасного життя. Чимало речей так чи інакше пов’язані з використанням штрихкодів і з кожним днем їх стає ще більше.

Представлені способи кодування та декодування штрихового коду дозволяють кодувати інформацію для відображення в умовах малої площі розміщення з великою щільністю даних з метою подальшої можливості швидкого декодування. Дане рішення зменшило швидкість як кодування інформації, так і декодування, в порівнянні зі схожим за будовою PFD417[3]. Декодування PFD417 займає 206мс, а у запропонованого способу – 141мс. Тобто, в 1.4 рази швидше.

### **Література**

1. ISO-IR-006: ASCII Graphic character set [Електронний ресурс] – Режим доступу до ресурсу: <https://www.itscj.ipsj.or.jp/iso-ir/006.pdf>
2. GS1-128 standard [Електронний ресурс] – Режим доступу до ресурсу: <https://www.gs1-128.info/>
3. PDF417 standard [Електронний ресурс] – Режим доступу до ресурсу: <https://www.iso.org/obp/ui/#iso:std:iso-iec:15438:ed-3:v1:en>

УДК 519.688

К.т.н., доцент Боярінова Ю. Є., студент Самофалов А. В.

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

## СПОСІБ УЩІЛЬНЕННЯ ДАНИХ БЕЗ ВТРАТ

### Abstract

**Yuliia Y. Boiarinova, assoc. prof., PhD, Andrii Samofalov, student**  
*Method of lossless compression*

*This paper concerns the lossless compression methods. The classical Huffman coding and Lempel-Ziv compression algorithms are discussed. Another method of lossless compression is proposed. The comparative analysis of existing and the proposed algorithms is given.*

### Вступ

У наш час алгоритми стиснення даних займають важливе місце в сферах діяльності, пов'язаних з інформаційними технологіями. Вони дозволяють зменшити обсяги даних, що зберігаються, або передаються, завдяки чому можна отримати значні вигоди для технічних та фінансових ресурсів компаній.

На даний момент використовуються декілька основних алгоритмів стиснення даних: алгоритм Гаффмана [1] та сімейство алгоритмів Лемпеля-Зіва [2].

Ідея алгоритму Гаффмана наступна: знаючи ймовірності появи символів у повідомленні, можна описати процедуру побудови кодів змінної довжини, що складаються з цілої кількості бітів. Символам з більшою ймовірністю ставляться у відповідність коротші коди. Коди Гаффмана володіють властивістю префіксності (тобто жодне кодове слово не є префіксом іншого), що дозволяє однозначно їх декодувати.

Процес стиснення в алгоритмах Лемпеля-Зіва виглядає наступним чином: послідовно зчитуються символи вхідного потоку і відбувається перевірка, чи існує в створеній таблиці рядків такий рядок. Якщо такий рядок існує, зчитується наступний символ, а якщо рядок не існує, то в потік заноситься код для попереднього знайденого рядка, рядок заноситься в таблицю, і пошук починається знову [3].

## Постановка задачі

Задача полягає в створенні нового способу стиснення даних, який дозволить отримувати не гірші результати при стисненні даних, ніж алгоритми, які використовуються у сьогоденні.

## Термінологія

*RLE* – *Run-length encoding* (англ.), *Кодування довжин серій або Кодування повторів* [4] – простий алгоритм стиснення даних, який оперує серіями даних, тобто послідовностями, в яких один і той же символ зустрічається кілька разів поспіль. При кодуванні рядок однакових символів, що становлять серію, замінюється рядком, який містить сам повторюваний символ і кількість його повторів.

## Опис способу

Метод, що пропонується, має назву діапазонне кодування.

*Діапазонне кодування* – метод стиснення даних, в якому послідовності даних, за можливості, кодуються меншою кількістю біт, ніж максимально встановленою.

В якості максимально встановленого значення в даному способі розглядається байт, що містить 8 біт. Отже, нам необхідно відокремити такі діапазони в даних, на кодування яких ми зможемо використати від 7 до 0 біт на один байт. Кількість біт, необхідна для запису значень в певному діапазоні, називається *бітовою шириною*. *Максимальною шириною діапазону* є максимальне значення, яке може записати дана бітова ширина: наприклад, для 7-бітного діапазону вона становить  $2^7 = 128$ , для 5-бітного діапазону –  $2^5 = 32$ , тощо.

Введемо позначення всіх діапазонів, використовуючи їх максимальну ширину: x256 (8-бітний діапазон), x128 (7-бітний діапазон), x64 (6-бітний діапазон), x32 (5-бітний діапазон), x16 (4-бітний діапазон), x8 (3-бітний діапазон), x4 (2-бітний діапазон), x2 (1-бітний діапазон). В нас ще залишається кодування довжин серій, яке має позначення x0 (тобто витрачає 0 біт на запис одного байту). Таким чином, усім відомий метод RLE є підмножиною діапазонного кодування.

Розглянемо на Рис. 1 цей підхід при наступному тестовому наборі даних (по осі ординат відкладені значення, які приймає кожен байт, на осі абсцис – номер байту).

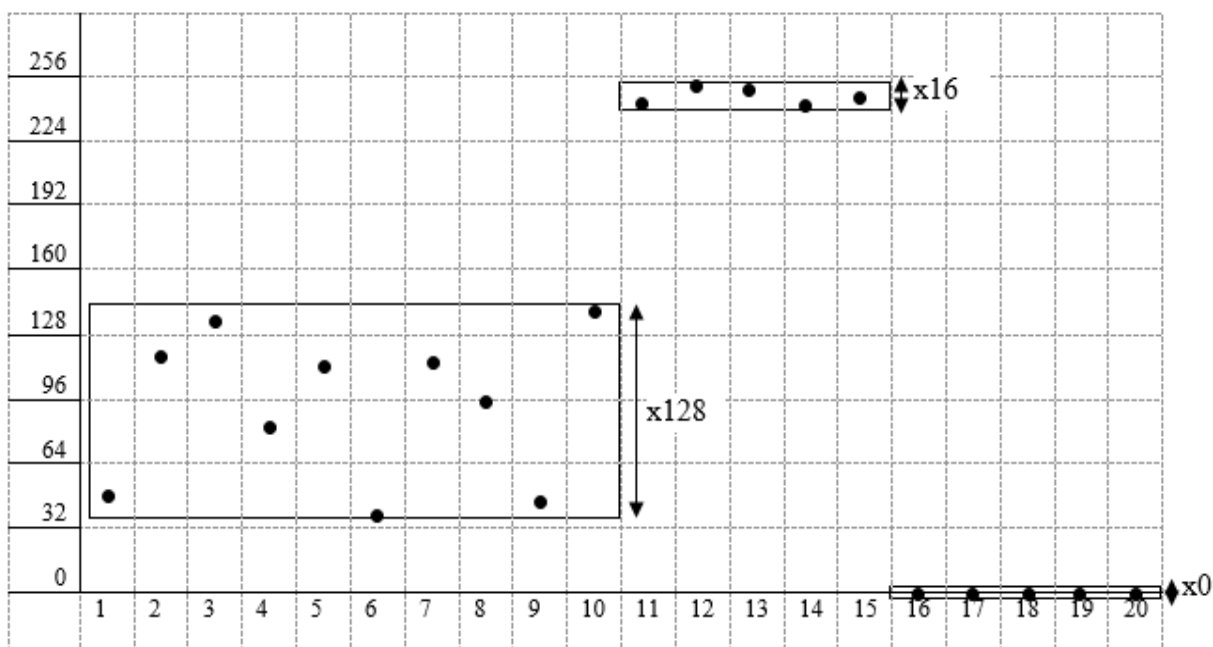


Рис. 1. Тестовий набір даних

Як бачимо, в нас усього двадцять байтів. Їх загальна довжина в бітах дорівнює 160. Але ми можемо організувати цю послідовність байт в певні діапазони з меншою бітовою шириною, а після цього скористатися цією меншою бітовою шириною для запису значення кожного байту. Для цього нам необхідно ввести поняття *базового значення*, або просто *бази*, яке означає мінімальне значення, яке може прийняти байт в кожному діапазоні. Усі значення байт в діапазоні отримуються як різниця значень поточного байту і бази. Ця різниця може приймати значення від 0 (значення байту співпадає з базою) до максимальної ширини діапазону мінус одиниця (бітовій ширині, всі біти якої встановлено в одиниці). Умовою входження чергового байту в діапазон є те, що отримана різниця між базою і значенням байту має бути меншою за максимальну ширину діапазону. Якщо ж ця умова не виконується, то нам необхідно завершити поточний діапазон та спробувати розширити його до найближчого більшого.

Для нашого тестового набору даних, в діапазоні x128 реальне мінімальне значення байту дорівнює 38, максимальне – 143. Тоді базу в цьому діапазоні буде значення 38, а максимальне перетворене значення байту буде  $143 - 38 = 107$ , для запису якого ми можемо використати сім біт, замість восьми.

При декодуванні, база, навпаки, додається до кожного перетвореного значення байту в діапазоні, і, таким чином, ми відтворюємо початкові значення байтів.

Отже, при використанні діапазонного кодування, нам необхідно витратити  $10 * 7 + 5 * 4 + 5 * 0 = 90$  біт для запису значень байтів замість 160 при звичайному записі. Але це ми отримуємо без врахування додаткових біт для запису інформації про властивості та довжини діапазонів.

Для запису інформації про властивості діапазонів нам необхідно обов'язково вказувати його бітову ширину (в нашому випадку необхідно 4 біта для вказання всіх можливих значень бітової ширини – від 0 до 8 включно) та значення бази (встановленою довжиною біт – в нашому випадку це довжина байту в 8 біт).

Для запису інформації про довжини діапазонів можна виділити наступні підходи:

- біт перевірки після кожного значення байту, який показує закінчився поточний діапазон, чи ще ні;
- стоп-комбінація – наприклад, для діапазону  $x128$  значення 127 (всі біти – одиниці) буде вказувати, що цей діапазон закінчився. Тоді максимальна ширина діапазону буде на одиницю менша;
- використання заголовку, в якому вказано довжину діапазону.

Але в перших двох підходах нам все одно треба вказувати довжину діапазону  $x0$  (RLE), оскільки в іншому випадку воно взагалі тоді не має сенсу.

## **Висновки**

Основною метою діапазонного кодування є спроба мінімізувати надлишковість у записі певних послідовностей значень байтів, об'єднуючи ці послідовності в діапазони з меншою бітовою шириною, завдяки чому ми можемо досягти певних ступенів стиснення даних.

Використання діапазонного стиснення, порівняно з алгоритмом Гаффмана, дає нам змогу використовувати RLE і не витратити на записи значень цих байтів навіть одного біту. Також це надає можливість спробувати знайти діапазони з меншою бітовою шириною серед даних, у яких кількість різних значень байтів у файлі більша за 128 (тобто ми не можемо виділити якусь дуже маленьку кількість біт для запису найчастішого значення, оскільки інші значення будуть мати довжини 9 біт і більше).

При порівнянні з алгоритмами Лемпеля-Зіва, ми маємо змогу відокремити менші діапазони, навіть, якщо значення в них без схожих послідовностей символів всередині. Завдяки цьому, ми можемо досягти певних покращених результатів при стисненні даних. Використовуючи

діапазонне кодування, ми можемо створити, як мінімум, діапазон, який використовує на 25% біт менше.

### Література

1. *Кормен, Томас; Лейзерсон, Чарльз; Рівест, Рональд; Стайн, Кліфорд* (2019). 16.3: Коды Гафмена. Вступ до алгоритмів (вид. 3). К.І.С. с. 443-451.
2. *Lempel A., Ziv J.* Compression of individual sequences via variable-rate coding // *IEEE Transactions on Information Theory*[en]. – 1978. – Т. 24, № 5. – С. 530-536.
3. *Introduction to Data Compression:*  
<http://www.cs.cmu.edu/~guyb/realworld/compression.pdf>
4. *Arnold R., Bell T.*[en]. A corpus for the evaluation of lossless compression algorithms // *IEEE Data Compression Conference.* – 1997. – С. 201-210.



Доцент, к. т. н., с.н.с. Боярінова Ю.Є., студент Штогрин П.П.

Національний технічний університет України  
“Київський Політехнічний Інститут імені Ігоря Сікорського”

## АВТОМАТИЧНЕ УПРАВЛІННЯ КВАДРОКОПТЕРОМ З ПОШУКОМ ОПТИМАЛЬНОГО МАРШРУТУ

### Abstract

**Yuliia Y. Boiarinova, assoc. prof., PhD; Shtohryn Pavlo, student**  
*Automatic quadcopter control with search for the optimal route*

*This paper describes the automatic flight control of a quadcopter. Data from on-board sensors is transmitted to the flight controller. Based on this data, the controller controls the engines and moves the quadcopter in a given direction. The paper proposes the use of a swarm algorithm to build the optimal flight route.*

### Вступ

Безпілотні літальні апарати, зокрема, їх різновид – квадрокоптери останнім часом все активніше використовуються у різних сферах людського життя. Застосування цим пристроям знаходять люди багатьох професій – від фотографів до військових. Основною сферою застосування апаратів такого типу є фото та відеозйомка, моніторинг різних параметрів навколишнього середовища з використанням додаткового обладнання.

Безпілотні літальні апарати завдяки своєму розміру можуть виконувати польоти в умовах обмеженого простору, у місцях, де є загроза життю людини, а також там, де недоцільно чи надто дорого використовувати повноцінні літальні апарати, наприклад, гелікоптери.

Головною задачею при розробці таких апаратів є підвищення автономності пристрою, що дозволяє мінімізувати вплив людського фактору, збільшення простоти використання, зменшення вимог до кваліфікації оператора.

Контролер польоту змінює швидкості обертання гвинтів для того щоб квадрокоптер міг рухатись у необхідному напрямку. Існує багато рівнів автоматизації польоту – від найпростіших, таких, як зависання у одній точці

чи політ на задані координати до складних, які базуються не лише на GPS, а і на датчиках відстані, барометрах, тощо.

Одною з проблем є можливість не тільки впоратись із коректним польотом із умовної точки А у точку В, а також прийняття рішення коригування маршруту для мінімізації втрат часу та енергії, якщо точок є багато.

### **Постановка задачі**

Задача полягає у створенні алгоритму управління безпілотним літальним апаратом, який дозволить автоматично орієнтуватись у просторі, знаходити оптимальний (або близький до оптимального) маршрут польоту кількох заданих локацій.

### **Термінологія**

БПЛА – безпілотний літальний апарат.

Автопілот – засіб керування технічним засобом, який передбачає відсутність потреби постійного вмішування людини у процес керування.

GPS модуль – пристрій, який може визначати свої координати за допомогою системи глобального позиціонування.

Акселерометр – пристрій, що вимірює прискорення.

Гіроскоп – пристрій, здатний реагувати на зміну орієнтації основи, на якій його встановлено, відносно інерціального простору.

Контролер польоту – апаратний модуль БПЛА, який відповідає за отримання даних із датчиків чи передавачів, їх обробці та керуванні рухом БПЛА на основі цих даних.

*PWM* (ШІМ) – pulse-width modulation (широтно-імпульсна модуляція). Процес управління потужністю методом пульсуючого вмикання та вимикання споживача енергії.

Лідар – пристрій для отримання та обробки інформації про віддалені об'єкти за допомогою активних оптичних систем.

### **Опис загального підходу**

Для того щоб БПЛА міг самостійно здійснювати пересування у просторі, він потребує як апаратних так і програмних удосконалень. Найпростіший режим автопілота – режим стабілізації. Його важко назвати повністю автоматичним, оскільки оператор все ще повинен задавати напрямок польоту, але квадрокоптер здатен сам утримувати заданий напрямок.[1] Для цього використовується триосьовий гіроскоп і триосьовий

акселерометр. Акселерометр вимірює прискорення, гіроскоп вимірює кутову швидкість. Завдяки цьому контролер польоту дізнається про поточні кути тангажу, крену та ризання[2]. Зовнішні умови, такі як вітер, розподіл ваги апарата або зношеність двигунів можуть впливати на кут нахилу. Задача режиму стабілізації – залежно від поточних параметрів коригувати швидкість гвинтів щоб досягти заданої позиції. На рисунку 1 зображено схему взаємодії модулів квадрокоптера.

Для зависання у заданій точці чи на заданій висоті використовуються схожі принципи, але необхідно додати GPS модуль для визначення координат, а також дальномір (наприклад сонар) та барометр для визначення висоти.

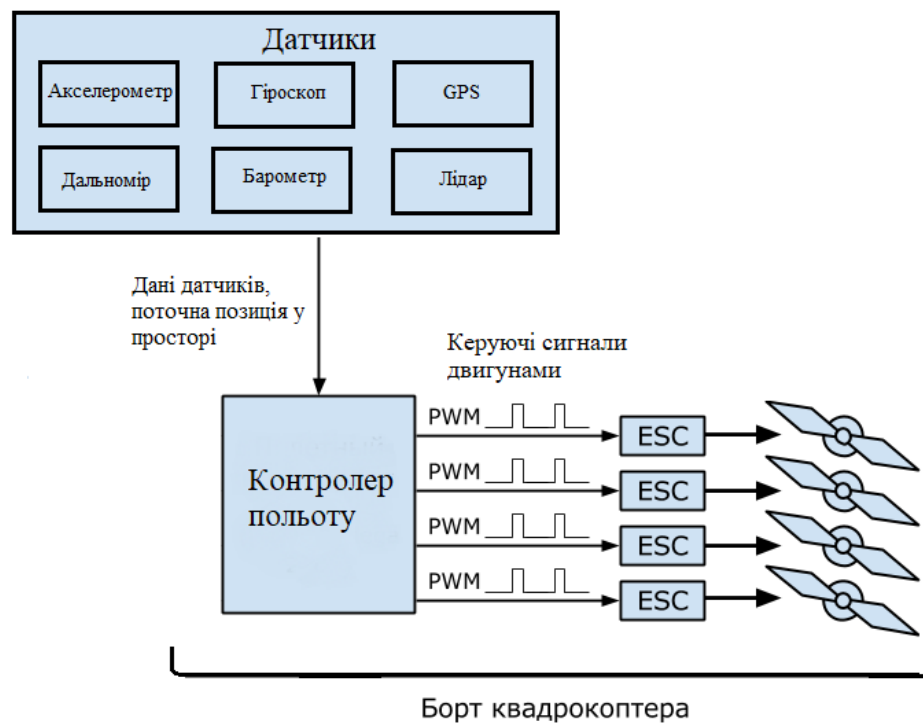


Рис. 1. Взаємодія модулів квадрокоптера

Для самостійного польоту по заданому маршруту – певному набору координат крім GPS модуля потрібен ще дальномір та лідар для того щоб квадрокоптер міг визначати та оминати перешкоди такі як дерева, будівлі, тощо.[3] Оскільки точність GPS не є ідеальною, то при наближенні до точки на певну відстань, яка називається радіусом контрольної точки, контролер польоту, за потреби, виконує фотозйомку чи інші задані дії, а потім встановлює наступну точку у якості цілі[1].

Якщо БПЛА повинен облетіти декілька точок у автономному режимі, то доцільно розрахувати оптимальний маршрут для економії часу та заряду акумулятора. Для вирішення такої задачі можна використати роеві алгоритми, наприклад мурашиний. Класичний мурашиний алгоритм не потребує багато пам'яті та складних обчислень на кожному кроці. Кожна мураха запам'ятовує список пройдених нею вузлів, який називається «список заборон». У ці вузли вона повертатись більше не буде на поточній ітерації. На наступній ітерації список заборон обнулюється. Крім списку заборон мурахи орієнтуються також по «привабливості» ребер. Ця привабливість визначається відстанню між вузлами, а також слідами феромонів, які залишили мурахи, що проходили по ребру раніше. Сліди феромонів обновляються на кожній ітерації[4].

Після виконання алгоритму квадрокоптер отримує послідовність координат, до яких потрібно рухатись. За допомогою режиму самостійного польоту по заданих координатах відбувається виконання поставленої задачі.

## **Висновок**

У цій роботі досліджено методи автоматичного керування БПЛА, а саме квадрокоптером. А також запропоновано використання роевого алгоритму для побудови оптимального маршруту.

Для того щоб квадрокоптер міг самостійно пересуватись у просторі потрібні його апаратні покращення – такі як GPS, дальномір, барометр, акселерометр, гіроскоп, лідар. Дані із цих пристроїв потрібно коректно обробити за допомогою програмного забезпечення та згенерувати необхідні керуючі сигнали двигунам на основі цих даних.

Задача пошуку оптимального маршруту може бути виконана окремо програмним способом. Роевий алгоритм створює послідовність координат, а алгоритми орієнтування у просторі та контролю польоту забезпечує пересування по заданих точках.

## **Література**

1. Разработка системы автоматического управления беспилотным летательным аппаратом мультироторного типа [Електронний ресурс]. Режим доступу : <https://mipt.ru/upload/medialibrary/72e/139-152.pdf> (дата звернення: жовтень 2021).

2. Программирование квадрокоптера на Arduino [Электронный ресурс]. Режим доступа : <https://habr.com/ru/post/227425/> (дата звернення: жовтень 2021).
3. Autopilot Design for a Quadcopter [Электронный ресурс]. Режим доступа: [https://www.researchgate.net/publication/331299062\\_Autopilot\\_Design\\_for\\_a\\_Quadcopter](https://www.researchgate.net/publication/331299062_Autopilot_Design_for_a_Quadcopter) (дата звернення: жовтень 2021).
4. Муравьиные алгоритмы [Электронный ресурс]. Режим доступа : <https://habr.com/ru/post/105302/> (дата звернення: жовтень 2021).

К.т.н., доцент Боярінова Ю. Є. студент Шуть М. В.

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

## ВИКОРИСТАННЯ ФРЕЙМВОРКУ ReactJS НА ПРИКЛАДІ РОЗРОБКИ ІНФОРМАЦІЙНО-ДОВІДКОВОЇ СИСТЕМИ ЛІКАРЯ

### Abstract

**Yuliia Y. Boiarinova, assoc. prof., PhD; Shut' Maxym, student**  
**USING THE REACTJS FRAMEWORK ON THE EXAMPLE OF DEVELOPING A  
DOCTOR INFORMATION AND REFERENCE SYSTEM**

*This article describes the analysis of the most popular services that can be used to optimize the workflow and increase the productivity of doctors, as well as the development of its own optimal information and two-way system for medical staff, which solves the problems of existing services. The created web application consists of two parts: front-end and back-end. ReactJS was used to develop the front-end part, as well as the following technologies: ReactJS Hooks, React Router, Axios. Technologies such as: Node.JS / JSON-Server were used to create the back-end part.*

### Вступ

Оптимізація робочого процесу будь-яких лікарів є актуальною впродовж багатьох років, зокрема в наш час. Більшість лікарських організацій прагне підвищити продуктивність всіх своїх співробітників за допомогою різноманітних методів, зокрема, сучасних інформаційних технологій. Одним з основних методів оптимізації робочого процесу працівників медичних закладів є організація прийомів пацієнтів, з якими кожного дня працюють співробітники.

Існує багато різноманітних способів для вирішення цієї проблеми, проте найбільш популярним і водночас сучасним та ефективним з них, є використання веб-технологій для цих завдань. На ранніх етапах розвитку комп'ютерної техніки майже не існувало спеціалізованих систем для організації робочого процесу, раніше більшість записів зберігалася виключно на папері. Проте сьогодні, на щастя, є можливість використовувати великі за обсягом хмарні сховища даних, на основі яких створюються різноманітні системи управління, які в свою чергу забезпечують доступ до більшості файлів з будь-якої точки світу.

## **Постановка задачі**

Задача полягає у проведенні аналізу найбільш популярних сервісів, котрі можливо використати для оптимізації робочого процесу та підвищення продуктивності роботи лікарів, а також в розробці власної оптимальної інформаційно-довідкової системи для працівників медичних закладів за допомогою фреймворку ReactJS.

## **Огляд існуючих рішень**

На сьогодні існує відома система HELSI. Нажаль, іноді вона працює нестабільно, тому лікарі вимушені вести нотатки на папері. Але в час діджиталізації все ж краще використовувати інформаційні технології для зберігання інформації. Для цього можна використовувати різноманітні сервіси, які дозволяють робити планування задач. Серед таких найпопулярнішими є: Microsoft Project, Google Таблиці, Todoist, Leader Task, Trello, а також Jira [1].

Кожен із вищезгаданих сервісів має свої переваги та недоліки. По-перше, всі вони мають спільну проблему - потребу в постійному доступі до мережі Інтернет. Сьогодні це досить легко вирішується, але, інколи, виникають такі обставини, при яких доступу може не бути. По-друге, не всі сервіси мають власні мобільні додатки. Це значною мірою зменшує гнучкість програми у використанні з різноманітних девайсів. По-третє, більшість з них не має можливості інтегрувати в себе корисні та ефективні додаткові сервіси, наприклад СМС-оповіщення. Врешті-решт, не всі сервіси можливо використовувати за допомогою таких найбільш популярних операційних систем, як: Windows та Mac OS, - одночасно [1].

Отже, необхідно створити такий веб-додаток, що міг би вирішити кожен з вищезгаданих проблем вже існуючих відомих сервісів. Таким чином, потрібно розробити додаток з багатофункціональним та водночас простим, зрозумілим для будь-яких користувачів інтерфейсом, в якому буде можливість планування прийомів, гнучкість та легкість в оперуванні з ними, детальні описи кожного з них, а також можливість створення нових прийомів, видалення уже існуючих та генерація СМС-оповіщень для користувачів за деякий час до початку прийому лікаря. Веб-додаток повинен стабільно працювати на різноманітних операційних системах, а також без постійного доступу до мережі Інтернет.

## **Розробка веб-додатку**

Для вирішення даної задачі необхідно розробити дві частини: front-end частину та back-end частину. Для проектування першої слід використовувати такі мови програмування, як: JavaScript, C++, Python або ж C#. Для розробки другої ситуація повністю аналогічна [2].

Для нашого проєкту найбільш доцільним та ефективним вибором буде використання мови JavaScript. Вибір саме цієї мови програмування обумовлений її орієнтованістю на веб-додатки, а також наявністю необхідних фреймворків, що значною мірою прискорять розробку проєкту.

Для створення front-end частини буде використано ReactJS, а також наступні технології: ReactJS Hooks, React Router, Axios.

Переваги ReactJS [3]:

- повторне використання коду;
- інтегровані нативні компоненти;
- всі необхідні інструменти уже включені в платформу;

Недоліки React Native:

- специфічність платформи;

ReactJS Hooks представляють собою функції, за допомогою яких можна «підчепитися» до стану і методів життєвого циклу React із функціональних компонентів. Слід зазначити, що хуки не працюють всередині класів – вони надають можливість використовувати React без них [3].

React Router представляє собою стандартну бібліотеку маршрутизації (routing) у React. Вона зберігає інтерфейс програми синхронізованим із URL у браузері. Дана технологія дозволяє маршрутизувати потік даних (data flow) в додатку зрозумілим способом. Слід зазначити, що головною перевагою React Router є можливість використовувати веб-додаток за допомогою URL-адресів, тобто він буде працювати, як звичайний сайт. В залежності від введеної адреси буде відкриватися окремий компонент або сторінка веб-додатку [4].

Axios — це широко відома JavaScript-бібліотека, що в свою чергу представляє собою HTTP-клієнт, заснований на промісах, та призначений для браузерів і Node.js. Саме її буде використано для створення GET, POST, PATCH та PUT запитів для оновлення даних БД [5].

Для створення back-end частини будуть використані такі технології, як: Node.JS / JSON-Server.

JSON-Server представляє собою сервер, що значно полегшує розробку додатків, котрі оперують с БД. Він необхідний для взяття та оновлення інформації в БД. Використання саме цієї технології обґрунтовано тим, що додаток буде реалізований за допомогою мови програмування JavaScript, на якій і побудована дана технологія, що є доволі зручним рішенням для back-end частини [6].

Отже, веб-додаток буде створено за допомогою технологій, що базуються на мові програмування JavaScript – ReactJS, та додаткових технологій для back-end частини - Node.JS / JSON-Server. Також будуть



використані інші технології та модулі для front-end частини, такі як: ReactJS Hooks, React Router та Axios.

## Висновки

В роботі проведено аналіз найбільш популярних сервісів, які можливо використати для оптимізації робочого процесу та підвищення продуктивності роботи лікарів, а також розроблено власну оптимальну інформаційно-довідкову систему для працівників медичних закладів за допомогою фреймворку ReactJS.

До найбільш популярних сервісів, які дозволяють робити планування задач, належать: Microsoft Project, Google Таблиці, Todoist, Leader Task, Trello, а також Jira. Кожен із них має свої переваги та недоліки. Після аналізу можна виділити їх головні вразливості: потреба в постійному доступі до мережі Інтернет, відсутність власних мобільних додатків, СМС-оповіщень, а також відсутність кросплатформності.

Отже, було розроблено інтерактивну систему, що вирішує кожен з вищезгаданих проблем та може використовуватись працівниками медичних закладів. Веб-додаток було виконано з використанням сучасних технологій, що надає можливість мати дуже високу швидкість взаємодії між клієнтською та серверною складовими. Також було використано компонентний підхід, що є однією із найбільших вимог в сучасній веб-розробці. Для створення front-end частини було використано ReactJS, а також наступні технології: ReactJS Hooks, React Router, Axios. Для розробки back-end частини було використано такі технології, як: Node.JS / JSON-Server.

## Література

1. Google Sheets [Електронний ресурс]. Режим доступу: [support.google.com/docs/topic/9054603?hl=ru&ref\\_topic=1382883](https://support.google.com/docs/topic/9054603?hl=ru&ref_topic=1382883)
2. Javascript [Електронний ресурс]. Режим доступу: [devdocs.io/javascript/](https://devdocs.io/javascript/)
3. ReactJS [Електронний ресурс]. Режим доступу: [ru.reactjs.org/docs/getting-started.html](https://ru.reactjs.org/docs/getting-started.html)
4. React Router [Електронний ресурс]. Режим доступу: [reacttraining.com/react-router/web/guides/quick-start](https://reacttraining.com/react-router/web/guides/quick-start)
5. Axios [Електронний ресурс]. Режим доступу: [www.npmjs.com/package/axios](https://www.npmjs.com/package/axios)
6. JSON-server [Електронний ресурс]. Режим доступу: [github.com/typicode/json-server](https://github.com/typicode/json-server)

УДК 004.62, 004.627

Студент Війтенко А.М., к.т.н., Павловський В.І.

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

## КОМП'ЮТЕРНА ERP-СИСТЕМА ОСУЧАСНЕНИХ РОЗРАХУНКІВ ТЕПЛОПОСТАЧАННЯ М. ОДЕСИ

### Abstract

**Viitenko Artem, student; Pavlovskiy Vladimir, assoc. prof., PhD**

***Computer ERP-system of modernized calculation of heat supply of Odessa***

*This paper concerns problems in the heat supply of a utility and offers a new way of presenting data. The proposed method of data construction and calculation can be used in the utility company "Heat supply of the city of Odessa" to automate business processes. Based on the proposed data representation, the performance of the calculation algorithm can be significantly increased and more receptive to system users.*

### Вступ

Автоматизація розрахунку теплопостачання є однією з важливих задач комунальних підприємств, які надають послуги теплопостачання населенню міста. До неї зводиться вирішення чималої кількості інших задач – побудова адекватної структури даних, актуальність даних, облік теплобалансу, формування платіжних квитанцій, формування актів за надані послуги юридичним особам і т.п.

У даній статі розглядаються питання побудови структури даних та автоматизації розрахунків спожитого тепла фізичними та юридичними особами у м. Одеса у відповідності до осучасненої методики Мінрегіонбуду. Запропоновані рішення є складовою ERP-системи на платформі IT-Enterprise.

### Постановка задачі

Метою роботи є побудова розвинутої структури даних та ERP-системи, яка, відповідно до вимог методики Мінрегіонбуду України про розподіл між споживачами обсягів спожитих у будівлі комунальних послуг №315, враховує всі складові системи теплозабезпечення кінцевого споживача та реалізація на цій основі автоматичного розрахунку споживання тепла фізичними та юридичними особами.

## **Термінологія**

*Теплобаланс* – рівність кількості теплоти палива, що надходить у котел, сумі корисно використуваної в ньому теплоти і теплових втрат. Зведення цього балансу дозволяє оцінити ККД котла та перевірити теплові розрахунки.

*ERP-система* – система планування всіх видів ресурсів підприємства. Дозволяє пов'язати всі інформаційні ресурси підприємства в єдине ціле, оптимізувати та прискорити процеси розрахунків, що ведуться на підприємстві.

## **Недоліки поточної системи оплати теплопостачання**

Останні роки комунальне підприємство «Теплопостачання м. Одеси» мало значні збитки. Аналіз показав, що це відбувається за рахунок того, що в розрахунку оплати теплопостачання не враховуються витрати на функціонування системи та обігрів місць загального користування. Суттєвими є втрати від "мінімального нарахування", коли фізична або юридична особа з приладом регулювання та обліку має нульові показники витрат, хоча сусіди отоплюють її з різних сторін. Подібна ситуація має місце у всіх містах країни, де є централізоване опалення.

На даний момент підприємство використовує систему «Биас», яка має ряд недоліків: значний час повного розрахунку – до 48 годин безперервної роботи; складна структура даних; розрахунок за застарілою методикою; високі вимоги до спеціальної підготовки операторів системи, що обмежує їх коло; жорстка залежність від розробників системи.

## **Удосконалена методика розрахунку оплати теплопостачання**

В рамках даної роботи слід виділити чотири принципові особливості методики Мінрегіонбуду України про розподіл між споживачами обсягів спожитих у будівлі комунальних послуг №315:

1. Змінюється порядок нарахування вартості опалення для окремих помешкань, але загальна вартість опалення для конкретних будівель не збільшується;

2. Споживачі тепла мають платити за обігрів місць загального користування (МЗК). Для цього відсоток МЗК віднімається від загального споживання тепла на дім і розподіляється між кожним споживачем в залежності від квадратури квартири/офісу. Для цього методика передбачає врахування всіх нежитлових приміщень та поверховості будинку, бо від цього залежить відсоток МЗК;

3. Споживачі тепла мають платити за функціонування системи (ФС) - кількість тепла, що припадає на втрати теплової енергії у домовій тепломережі, а також за обсяг втрат теплової енергії від різних транзитних ділянок трубопроводів, що надходять у приміщення, де комунальна послуга з постачання теплової енергії не надається [3]. Для цього відсоток

ФС, який залежить від поверховості будинку, віднімається від загального споживання тепла на дім і розподіляється між кожним споживачем в залежності від квадратури квартири/офісу. Для цього методика передбачає врахування всіх структурних особливостей теплопроводної мережі дому;

4. Модифікація способу нарахування плати за обігрів приміщень з приладами регулювання та обліку тепла, де споживачі вимикають батареї і їх отоплюють сусіди з різних боків. В цьому разі особам, якщо вони не спожили мінімальна частину середнього питомого споживання, додається мінімальне нарахування. При цьому частина тепла, яка лягла на мінімальне нарахування, віднімається від загального споживання помешкань без приладів регулювання та обліку [2]. Для цього методика передбачає врахування всіх приміщень з приладами регулювання та обліку тепла та їх показників.

### **Структури даних оновленої системи оплати теплопостачання**

Впровадження методики 315 вимагає докорінної зміни існуючої структури даних і врахування всіх складових системи теплозабезпечення кінцевого споживача. Враховуючи великий обсяг даних, що обробляє система, ця структура має забезпечувати швидкий доступ до всіх необхідних даних.

Основним об'єктом, від якого починається розрахунок оплати, є дім, до якого приєднуються квартири та допоміжні будівлі. У кожній квартири є певні технічні характеристики (навантаження, розрахункова площа, демонтована площа, джерело виробітку теплової енергії, прилади обліку тощо) та додаткова інформація (точна адреса, контрагенти, тип об'єкта – офіс чи житлова квартира тощо). За допомогою цих зв'язків забезпечується можливість виконання розрахунку теплопостачання та створення квитанцій та актів розрахунку для юридичних осіб, аналіз теплобалансу, обмін даними з управлінням соціального захисту населення, банком тощо.

Основні таблиці:

- об'єкти-споживачі;
- прилади обліку;
- показники приладів обліку за періодами (рік-місяць);
- особові рахунки;
- договори з юридичними особами;
- нарахування по особовим рахункам в грн. за періоди;
- нарахування спожитого тепла в Гкал по об'єктам-споживачам за періодом;
- контрагенти;

Важливе значення для прискорення розрахунків мають довідники, відповідно до яких проводиться нарахування:

- функціонування системи (%) та місця загального користування (%) з урахуванням поверховості будинку;

- тарифи для типу нарахувань (населення, юридичні особи, бюджетні організації, релігійні організації тощо);
- адреси споживачів;
- котельні;
- температурні коефіцієнти за періодом;

За допомогою цих та багатьох інших таблиць та довідників реалізовано облік розподілення тепла та нарахувань за опалення.

### **ERP-система розрахунків теплопостачання м. Одеси**

Безпосередньо розрахунок вартості теплопостачання для конкретно споживача є тільки однією складовою великої системи розрахунку теплопостачання м. Одеси. Сюди відноситься облік і управління всієї матеріальною, технічною, фінансовою, юридичною, комунікаційною і багатьма іншими складовими системи теплопостачання. На сьогодні для роботи з усіма цими ресурсами використовуються ERP- системи. В рамках реалізації цього проекту вибрана система української розробки IT-Enterprise. Це повнофункціональна ERP-система, що включає в себе системи MRPII, MES, APS, EAM, SCM і CRM [1]. Вона надає персоналу підприємства та користувачам можливості отримання повної та достовірної інформації про стан теплозабезпечення та розрахунків за нього.

Система має трирівневу організацію: вебсервер, сервер додатків та сервери баз даних. Вона підтримує дві серверні платформи: Microsoft SQL Server 2012/2008R2 та Oracle 11gR2. В якості платформи розробки бізнеслогіки використовується .Net Framework 4.0 та C# 4.0.

До переваг системи відноситься її мікросервісна архітектуру, що є зручним для реалізації процесу безперервної поставки програмного продукту.

В рамках розрахунку вартості теплопостачання та взаємодії зі споживачами і сумісними організаціями система надає можливості, наведені на рис. 1.

Час повного розрахунку вартості опалення зменшено до 5 годин безперервної роботи. Це зумовлено особливостями структури даних та організації системи.

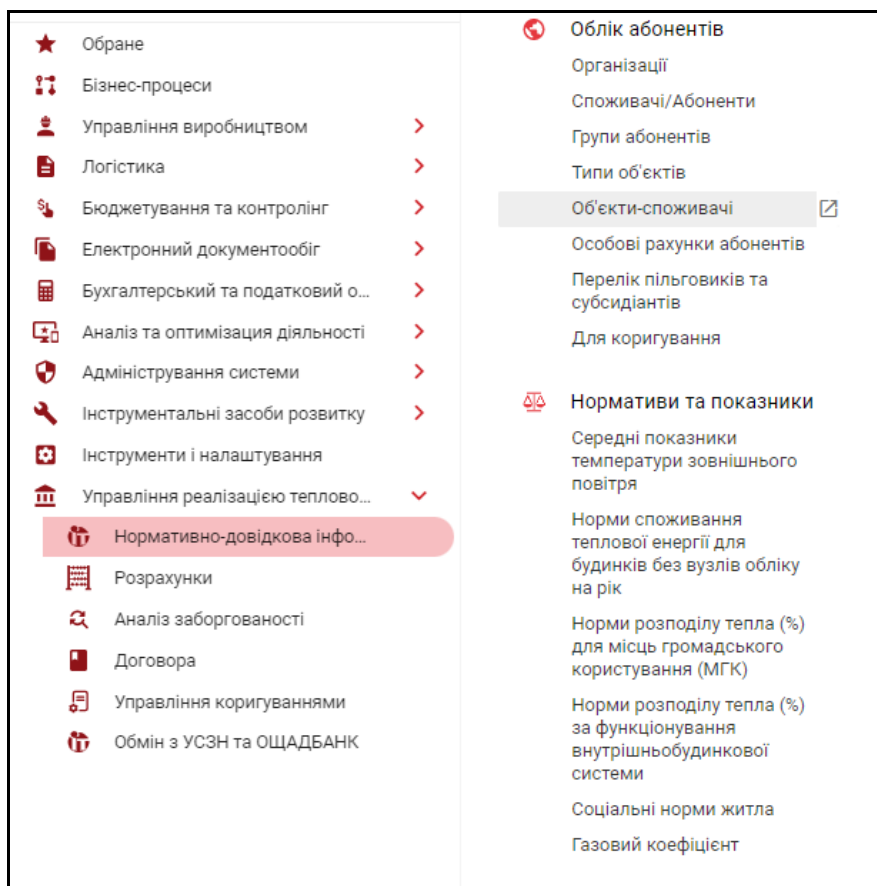


Рис. 1. Інформаційні та функціональні можливості ERP-системи розрахунків тепlopостачання

## Висновки

У роботі розглянуто ERP-систему розрахунку теплоспоживання. Вона буде використана у комунальному підприємстві «Теплопостачання м. Одеси» за для автоматизації бізнес процесів підприємства, таких як – формування нарахувань, створення квитанцій оплати, відслідковування теплобалансу, занесення показників приладів обліку і багатьох інших.

Порівняно з попередниками система вона забезпечує суттєво вищу швидкодню, гнучкість в модифікації та зручність для користувачів.

## Література

1. IT-Enterprise. *IT-Enterprise*: веб-сайт: URL: <https://www.it.ua/ru> (дата звернення: 29.10.2021)
2. ТМО - Теплопостачання міста Одеси. *ТМО*: веб-сайт: URL: <https://teplo.od.ua/> (дата звернення: 28.10.2021)
3. Methodology №315 - heat distribution technique. *Methodology №315*: веб-сайт URL: [http://search.ligazakon.ua/1\\_doc2.nsf/link1/RE32954.html](http://search.ligazakon.ua/1_doc2.nsf/link1/RE32954.html) (дата звернення: 28.10.2021)

УДК 519.688

Магістрант Додяк Д.І., ст. викладач Дробязко І.П.

Національний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»

## МЕТОДИ ОБФУСКАЦІЇ ПРОГРАМНОГО КОДУ НА ОСНОВІ СКЛАДЕНИХ ФУНКЦІЙ ТА АФІННИХ ШИФРІВ

### Abstract

**Dmytro I. Dodiak, student; Iryna P. Drobiazko, senior lecturer**  
**Obfuscation methods of programmed code based**  
**on composite functions and affine chipers**

*This paper concerns the task of code obfuscation and protection of intellectual property in software world. Two new obfuscation algorithms based on composite functions and one new algorithm based affine chipers are proposed and shown in examples with mathematical explanation and additional analysis.*

### Вступ

Швидкий розвиток мультимедійних та інтернет-технологій викликав необхідність досліджень у сфері захисту даних. Кожна компанія володіє інтелектуальну власністю. До інтелектуальної власності часто відносяться алгоритми, які безпосередньо вбудовані в програмне забезпечення, що підлягає продажу клієнтам. При забезпеченні конкурентної здатності компаній на ринку, захищеність такого програмного забезпечення є перевагою, тому важливе значення мають підхід і алгоритми, що використовуються для захисту.

Статистичні дані показують, що чотири з кожних десяти програм у всьому світі є піратськими [1]. Обфускація коду є новим напрямком досліджень у сфері захисту програмного забезпечення. Останнім часом вона набуває все більшої актуальності [2-4]. Дана технологія є формою захисту програмного забезпечення від несанкціонованого зворотного інжинірингу.

### Постановка задачі

Метою даної роботи є підвищення рівня захищеності програмного коду шляхом створення нових алгоритмів обфускації, що поєднують складені функції та афінні шифри.

## Термінологія

*Обфускація* – це процес шифрування програмного коду. Результатом обфускації є програмний код, який складніше зрозуміти, але функціонально він еквівалентний оригіналу.

*Обфускатор* – це програма, яка використовується для перетворення програмного коду.

## Опис розроблених алгоритмів обфускації

На основі дослідження існуючих алгоритмів обфускації розроблені наступні алгоритми.

1) Перетворення індексу масиву з використанням складених функцій.

Нехай  $I = f(i) = 2 * i + 3$  – функція, що представляє нове значення  $I$ . Тоді  $J = g(I) = f((I - 3)/2)$  – функція, що представляє нову позицію  $i$ -го елемента в перевпорядкованому масиві. Отже, змінна-член  $i$  може бути представлена як складена функція  $f(g(i))$ .

```
int i = 1;                               int I = f(i);
while (i < 1000) {                         while (I < f(1000)) {
    A[i];                                    A [g (I)];
    i++;                                     I = I+2;
}                                           }
```

Рис. 1. Приклад програмного коду для алгоритму перетворення індексу масиву

Приклад програмного коду (рис. 1) показує використання складених функцій для приховування елемента індикатора позиції масиву. Результати, які представлені в табл. 1, показують, що значення змінної  $i$  залишається незмінним до  $i$  після обфускації.

Таблиця 1  
Порівняння змінних  $i$ ,  $A$ ,  $J$  до та після процесу обфускації

$i$	$I = f(i) = 2 * i + 3$	$J = g(i) = f((I-3)/2)$
1	5	1
2	7	2
3	9	3
4	11	4



## 2) Перетворення аргументів за допомогою складених функцій.

Використання складених функцій може бути розширено, щоб приховати аргументи методів, що належать класам. Нехай  $f(i)=i+1$  — функція, що представляє аргумент методу  $B$ . Тоді  $g(i)=f(i)+2$  — функція, що представляє значення, яке повертається методом  $B$ . Композиція функцій  $f$  і  $g$  може бути використана як представлено в наступному фрагменті коду (рис. 2). Нехай  $A = f(i) = i + 1$ ,  $B = g(i) = f(i) + 2$  і  $j = B(i)$  після обфускації,  $j = B(i) + 1$  перед обфускацією. Значення  $A$ ,  $B$  і  $j$  наведено в табл. 2 для  $j = 0, 1, 2, 3, 4$ . Сірий стовпець представляє значення до обфускації, а білий стовпець — значення після обфускації. Результати в табл. 2 показують, що хоча проміжні значення, які використовуються складеними функціями, різні, кінцеве значення  $j$  залишається незмінним.

```

int j=0;
for (int i = 0; i< 5; i++) {
    j = B (i) + 1;
}
int method B (int a) {
    int b = a + 2;
    return (b);
}

int j =0;
for (int i = 0; i< 5; i++){
    j =B (i);
}
int method B (f (i)) {
    return (g (i));
}

```

Рис. 2. Приклад програмного коду для алгоритму перетворення аргументів

Таблиця 2  
Порівняння змінних  $i$ ,  $A$ ,  $B$ ,  $j$  до та після процесу обфускації

$i$	0	0	1	1	2	2	3	3	4	4
$A$	-	1	-	2	-	3	-	4	-	5
$B$	2	3	3	4	4	5	5	6	6	7
$j$	3	3	4	4	5	5	6	6	7	7

## 3) Шифрування рядкових літералів за допомогою афінних шифрів

Рядкові літерали в програмі можуть бути обфусковані в шифрований текст на основі техніки афінного шифру [1], [7]. Кожен алфавіт у рядку буде просуватися вперед певною кількістю (параметр обфускації) алфавітів, яка вибирається випадковим чином. Кожен рядок зашифрований в обфускованій програмі. Для отримання розшифрованого значення рядкових літералів на одержувачі, будь-яке посилання на рядок має бути замінено викликом методу, який його розшифровує.

Як приклад розглянемо рядок «Caser». Нехай випадково вибраний параметр обфускації дорівнює 5. Це означає, що кожна літера рядкового літералу буде висунута вперед на 5 літер. Отже, зашифроване значення рядка має вигляд – «ігукх». Для відновлення початкового значення, програма деобфускації викличе функцію  $f^{-1}$ , яка матиме значення параметра обфускації 5, що використовувалось для шифрування повідомлення під час обфускації. Функція  $f^{-1}$  розшифрує назву, відсуваючи кожен алфавіт назад на 5 літер.

Цю техніку можна легко скомпрометувати, якщо зловмисник дізнається параметр обфускації. У випадку, якщо параметр обфускації можна приховати як композицію деяких чисел, сума яких є простим числом, реверс-інженеру буде важко скомпрометувати систему реєнтабельним способом. Отже, очевидно, що алгоритми афінного шифрування у поєднанні зі складеними числами можуть бути використані для обфускації програмного коду .

## Висновки

В даній роботі запропоновано нові алгоритми обфускації, а саме: перетворення індексів масивів, перетворення аргументів методів за допомогою складених функцій, шифрування рядкових літералів за допомогою афінних шифрів. Використання цих алгоритмів дозволяє ефективно приховувати літерали, логічні оператори та аргументи методу. Кожен з алгоритмів експериментально підтверджено. Запропоновані алгоритми обфускації можуть стати хорошим доповненням до класичних алгоритмів обфускації і тим самим сприятимуть підвищенню захищеності програмного коду.

## Література

1. *Katzenbeisser*, “Information Hiding Techniques for Steganography and Digital Watermarking”, Boston Artech House, 2000 .
2. *M.R Stytz*, “Considering Defense in Depth for Software Applications”, IEEE Security & Privacy, February 2004.
3. *P. Tyma*, “Encryption, hashing, & obfuscation”, ZD Net April 2003.
4. *D. Low*, “Protecting Java Code via Code Obfuscation”, ACM Crossroads Student Magazine, Spring 1998.

УДК 519.6

Ст. викладач Дробязко І.П., студент Дученко О.С.

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

## МОДУЛЬ АВТОМАТИЧНОГО ВИБОРУ АЛГОРИТМУ ПЕРЕВПОРЯДКУВАННЯ ЕЛЕМЕНТІВ РОЗРІДЖЕНОЇ МАТРИЦІ ДЛЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ РОЗВ'ЯЗУВАННЯ СЛАР

### Abstract

**Iryna P. Drobiazko, senior lector; Oleksandr S. Duchenko, student**  
*Module of automatic selection of sparse matrix permutation algorithm for intellectual  
SLAE solving system*

*This paper discusses the solution for linear systems solving optimization. The solution is a machine learning-based module. The Convolutional neural network as a core of the module is responding for selecting the best of possible permutation algorithms for given sparse matrix.*

### Вступ

Невід'ємною складовою багатьох наукових досліджень та технічних розробок є математичне та комп'ютерне моделювання. У значній кількості випадків при математичному моделюванні виникає потреба розв'язувати задачі лінійної алгебри. Матриці цих задач, по-перше, мають надвеликі порядки, по-друге, є розрідженими, що призводить до значних обсягів використовуваної пам'яті та складних обчислень для їх розв'язання. Крім того, кожна нова прикладна задача породжує матрицю унікальної розрідженої структури.

Для зменшення обсягу пам'яті, кількості міжпроцесних обмінів даними, відповідно прискорення обчислень пропонується здійснювати перевпорядкування матриць.

### Постановка задачі

Метою даної роботи є створення модуля для автоматичного вибору оптимального алгоритму перевпорядкування елементів розріджених матриць на основі аналізу ефективності різних алгоритмів з використанням згорткової нейронної мережі.

## **Термінологія**

**СЛАР** – система лінійних алгебраїчних рівнянь.

**Розріджена матриця** – матриця, більша частина елементів якої є нульовими.

**Портрет матриці** – зменшене зображення структури розрідженої матриці.

**CNN** (Convolutional neural network) – згорткова нейронна мережа.

**Matlab** – пакет прикладного програмного забезпечення для числового аналізу з вбудованими алгоритмами перевпорядкування.

**Tensorflow** – бібліотека, що містить функції для високопродуктивних паралельних матричних обчислень на графічних прискорювачах Nvidia.

## **Проблема аналізу вхідних даних при розв’язуванні СЛАР**

На сьогоднішній день для розв’язання система лінійних алгебраїчних рівнянь (СЛАР) розроблено та реалізовано достатньо багато паралельних алгоритмів. Але залишається проблема аналізу вхідних даних та подальшого вибору оптимального шляху розв’язання конкретної СЛАР. Вітчизняними вченими вже розроблено модуль [1] інтелектуальної системи [2], який, аналізуючи портрет матриці, обирає оптимальний алгоритм розв’язання відповідної СЛАР.

Вдосконаленням даної системи є розробка модуля, який здійснюватиме вибір оптимального алгоритму перевпорядкування елементів матриці. Оскільки поелементний аналіз матриці може потребувати значних обсягів обчислень та пам’яті, для вибору алгоритму перевпорядкування доцільно використовувати портрети матриці. Таке перевпорядкування дозволить зменшити кількість елементів, які будуть розвинуті під час розв’язання СЛАР, що, у свою чергу, зменшить обсяги необхідних обчислень та пам’яті. Алгоритм перевпорядкування необхідно обирати для кожної конкретної задачі, оскільки він може бути оптимальним для однієї задачі, а для іншої – ні.

## **Опис модуля для вибору алгоритму перевпорядкування**

До складу модуля входить процедура відображення портрету матриці та нейронна мережа. Аналіз портретів пропонується здійснювати нейронною мережею спеціального класу, розробленого для обробки зображень, а саме – CNN. Розроблена нейронна мережа отримує вхідні дані у вигляді зображення розміром 74x74 пікселі. Навчальна вибірка складається з 387 матриць, відібраних з відкритої колекції університету Formerly Флориди [3]. За допомогою вбудованих в Matlab засобів оригінальні матриці перевпорядковуються з використанням наступних алгоритмів [4]:

- colperm – алгоритм, який впорядковує стовпчики матриці за збільшенням кількості їх ненульових елементів;
- amd – алгоритм мінімальної степені;
- dissect – алгоритм вкладених перерізів;
- symrcm – симетричний обернений алгоритм Катхіл-Маккі.

Перевпорядковані матриці, а також їх оригінали, проходять процедуру генерації портрету. Оскільки виконується аналіз портрету, неможливо визначити точний обсяг пам'яті, який займатиме власне розвинена матриця. Для вирішення цієї проблеми розроблено алгоритм приблизного відображення розвинення матриці на основі вбудованого в Tensorflow алгоритму символічного розвинення Холецкого [5]. Порівняння портретів розвинення оригінальної матриці та приблизного передбачення розвинення цієї матриці за її портретом представлено на рис. 1.

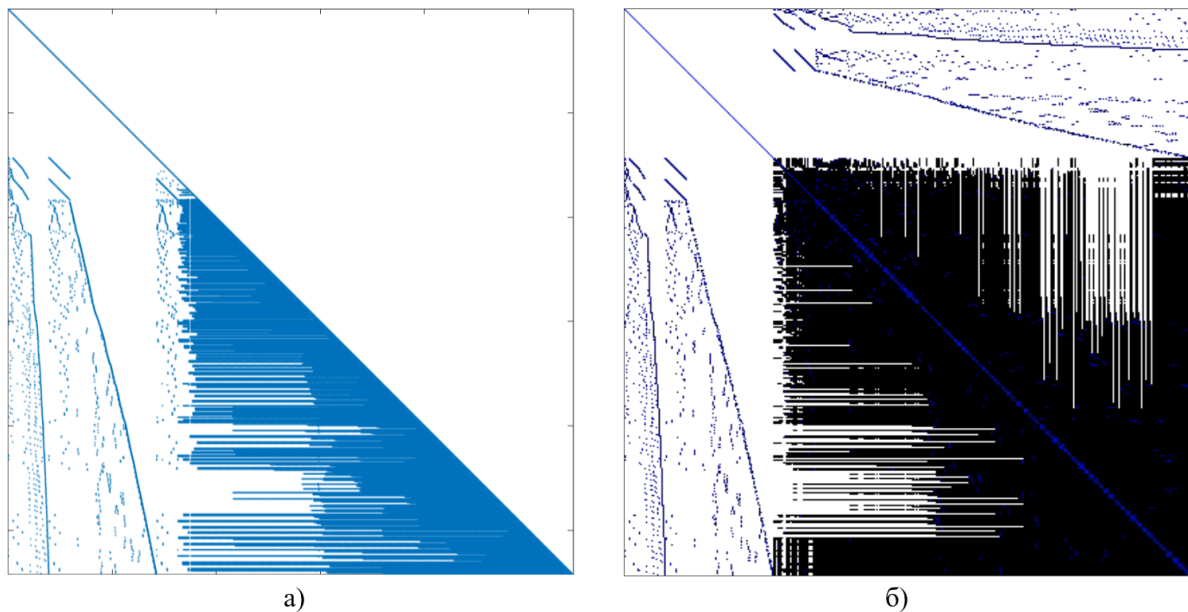


Рис. 1. Порівняння розвинення а) оригінальної матриці (відображено лише нижній трикутник) з б) приблизним розвинення за портретом

Передбачені для розвинення елементи передаються за допомогою альфа-каналу зображення. Пікселі, в яких передбачається розвинення елементів в альфа-каналі зображення, мають значення 255, інші ж – 0. За допомогою передбачення розвинення елементів відбувається обчислення приблизного обсягу пам'яті, необхідного для визначення оптимального алгоритму перевпорядкування. Так, приблизний обсяг пам'яті, необхідний для розвинення матриці, дорівнює добутку кількості розвинених елементів на розмірність змінної, що зберігає елемент, у байтах. Кількість елементів обчислюється за наступною формулою:

$$elems = \left( \sum_{i=0}^w \sum_{j=0}^h A(i, j) \right) * res ,$$

де *elems* – шукана кількість елементів, *w* та *h* – ширина і висота зображення відповідно, *res* – кількість елементів оригінальної матриці, що представляється одним пікселем її портрету.

Для кожної з оригінальних матриць серед її перевпорядкованих варіантів обирається оптимальний за обсягом пам'яті, необхідної для розвинення. Якщо цей обсяг для оригінальної матриці менший або дорівнює обсягу пам'яті матриці в інших варіантах, обирається варіант «не перевпорядковувати».

На виході створеної нейронної мережі є п'ять станів, а саме: «не перевпорядковувати», «solperm», «amd», «dissect», «symrcm».

Нейронна мережа складається з 3 шарів згортки, 3 максимізаційних агрегувальних шарів та має структуру, представлену на рис. 2.

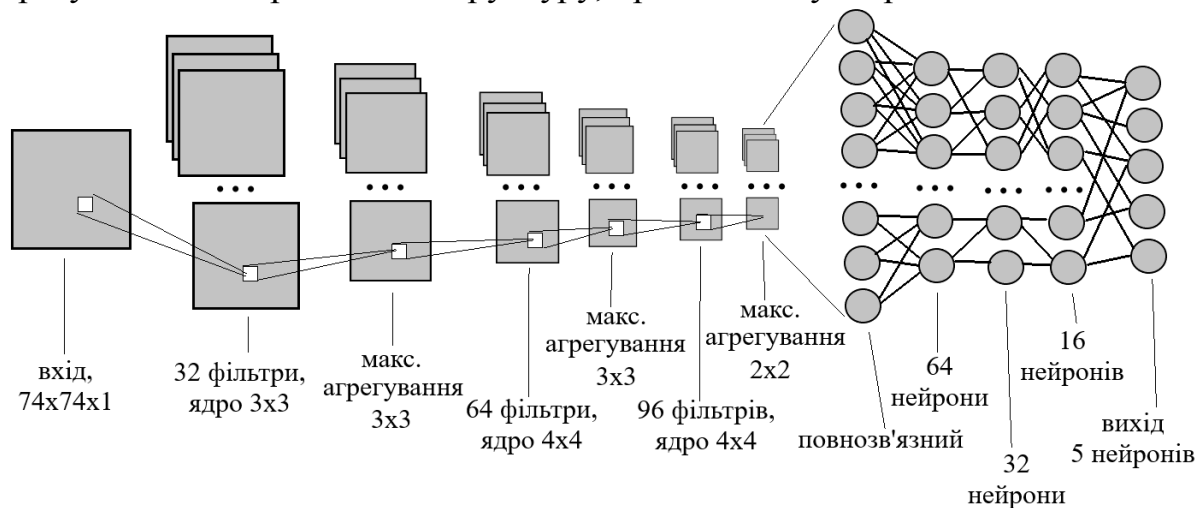


Рис. 2. Структура нейронної мережі

Вхідна матриця перевпорядковується за передбаченим оптимальним алгоритмом, зберігається і для неї будується портрет. Визначається обсяг пам'яті, необхідний для подальшого розвинення. Якщо він більший, ніж у випадку оригінальної матриці, на систему розв'язування подається оригінальна матриця, а якщо менший – перевпорядкований варіант.

На тестовій вибірці з 97 матриць отримано наступні результати:

- точність передбачення оптимального алгоритму – 75.25%;
- середній обсяг пам'яті, необхідний для розв'язування оригінальних матриць – 74 849 Мб;
- обсяг пам'яті для ідеального варіанту, при якому система передбачає всі оптимальні алгоритми – 45 277 Мб;
- обсяг пам'яті при розв'язуванні за допомогою поточного передбаченого набору алгоритмів – 46 150 Мб;

- обсяг пам'яті при розв'язуванні всієї вибірки одним оптимальним для неї алгоритмом – 47 096 Мб.

## Висновки

В ході роботи розроблено алгоритми представлення структури матриці, а також передбачення приблизного розвинення у вигляді зображення. Також розроблено нейронну мережу, яка передбачає оптимальний алгоритм перевпорядкування за портретом вхідної матриці.

Розроблений модуль потенційно забезпечить оптимізацію структури розріджених матриць на етапі їх подання на систему для розв'язання СЛАР, що в свою чергу зменшить обсяги обчислень та пам'яті. На тестовій вибірці зменшення необхідного обсягу пам'яті досягає 38,3%.

Подальшими шляхами для вдосконалення запропонованих рішень є збільшення навчальної вибірки для нейронної мережі, оптимізація структури мережі та покращення представлення структури матриці.

## Література

1. В.А. Сидорук, П.С. Єршов, Д.О. Богурський, О.Р. Марочканич Інтелектуалізація обчислень для задач математичного моделювання складних процесів та об'єктів: [Електронний ресурс]. – Режим доступу: <http://dspace.nbuv.gov.ua/bitstream/handle/123456789/161944/18-Sidoruk.pdf?sequence=1>
2. И.В. Сергиенко, И.Н. Молчанов, А.Н. Химич Интеллектуальные технологии высокопроизводительных вычислений: [Електронний ресурс]. – Режим доступу: <http://dspace.nbuv.gov.ua/bitstream/handle/123456789/45637/20-Sergienko.pdf?sequence=1>
3. Колекція університету Formerly Флориди: [Електронний ресурс]. – Режим доступу: <https://sparse.tamu.edu>
4. Алгоритми перевпорядкування елементів розріджених матриць: [Електронний ресурс]. – Режим доступу: <https://www.mathworks.com/help/matlab/math/sparse-matrix-reordering.html>
5. Алгоритм символьного розвинення Холецкого : [Електронний ресурс]. – Режим доступу: [https://en.wikipedia.org/wiki/Symbolic\\_Cholesky\\_decomposition](https://en.wikipedia.org/wiki/Symbolic_Cholesky_decomposition)

УДК 519.688

проф., професор Зайцев В.Г., студент Поліщук А.В.

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

## МОДИФІКАЦІЯ АЛГОРИТМУ PD2 ПЛАНУВАННЯ ЗАДАЧ У СИСТЕМАХ РЕАЛЬНОГО ЧАСУ

### Abstract

Vladymir G. Zaitsev, prof., PhD; Andrii Polishchuk, student  
*Modified PD2 algorithm for task planning in real-time systems*

*This paper concerns the of the PD2 algorithm for task planning. Existing algorithms have their own specifics of use in certain situations. Each of the algorithms has both strengths and weaknesses. Thus, a new hybrid planning algorithm was proposed that combines both distributed and global planning approaches. Thus, it was possible to reduce overhead costs and task planning costs.*

### Вступ

Планування задач у системах реального часу є однією з основних функцій яка також являється головним механізмом реалізації багатозадачності. Сучасні високопродуктивні програми пред'являють серйозні вимоги до роботи систем реального часу з обмеженими ресурсами. Для успішної одночасної роботи складних алгоритмів з дотриманням часових обмежень вкрай необхідно оцінювати продуктивність алгоритмів планування задач, оскільки від оптимального вибору того чи іншого алгоритму залежить стабільність роботи системи в цілому, що може бути принципово для жорстких систем реального часу, де недотримання часових обмежень може спричинити до фатальних наслідків

### Постановка задачі

Задача полягає в модифікації існуючого глобального алгоритму планування PD2 таким чином, щоб зменшити накладні витрати та витрати на планування.

### Термінологія

*Система реального часу* – це система яка в передбачуваний час на непередбачувану появу зовнішніх подій. Це визначення пред'являє до системи цілком певні базові вимоги.



**Своєчасна реакція.** Після того як відбулася подія, реакція має бути не пізніше, ніж через необхідний час. Перевищення цього часу розглядається як серйозна помилка.

**Багатозадачність** - властивість операційної системи забезпечувати можливість паралельної (або псевдопаралельної) обробки декількох завдань.

### Опис алгоритму

PD2 – найбільш ефективний, відомий алгоритм для планування періодичних задач у багатопроцесорних системах реального часу серед Pfair (proportionate fair) алгоритмів. Термін Pfair означає, що задачі справедливо розподіляються серед доступних процесорів. Завдання потребують виконуватися з постійною швидкістю, проте задачі не може бути виділено відразу весь виконавчий блок процесора. Замість цього задача ділиться на підзадачі рівної довжини та виділяються кванти процесорного часу. Алгоритм використовує квантування, це значить, що процесорний час ділиться на дискретні кванти певного часового розміру та кожного разу коли починається квант – планувальник приймає рішення якій задачі слід виконуватися за час поточного кванту. Квант часу повинен мати оптимальний розмір, тобто тривати не дуже мало та і не дуже довго. При дуже коротких квантах збільшується кількість викликів планувальника, що збільшує накладні витрати, та витрати на планування. При дуже великих квантах процесор може багато часу простоювати без діла.

Головною властивістю Pfair планування є те, що кожна задача повинна виконуватися з постійною швидкістю на всіх процесорах в системі, що досягається присвоєнням ваг кожній задачі (T.w - task weight)

$$T.w = \frac{T.c}{T.p}$$

Де T.c це процесорний час необхідний для виконання задачі, а T.p це період.

В алгоритмі PD2 кожна задача ділиться на підзадачі однакової довжини які мають свій час завершення роботи, та кінцевий термін виконання (дедлайн). Ці часові характеристики називаються псевдочасом завершення та псевдодедлайном  $r.T_i$  та  $d.T_i$  відповідно.

$$r.T_i = \left\lfloor \frac{i-1}{T.w} \right\rfloor \quad d.T_i = \left\lceil \frac{i}{T.w} \right\rceil$$

Тут  $i$  позначається  $i$ -та підзадача роботи задачі  $T$ , де  $i \Rightarrow 1$ .

Алгоритм PD2 визначає пріоритети підзадач на основі їх псевдодедлайну. Підзадачі з меншим псевдодедлайном надається більший пріоритетом ніж іншим. Це виражається в наступному рівнянні.

$$X_i < Y_i \text{ якщо } d(X_i) < d(Y_i)$$

(якщо  $X_i < Y_i$ , то підзадачі  $X_i$  надається більший пріоритет ніж  $Y_i$  )

Коли підзавданням з однаковим псевдодедлайном потрібно визначити пріоритет, у алгоритмі використовується 2 правила. За першим правилом береться бітове значення, пов'язане з кожною підзадачею. Бітове значення підзадачі дорівнює 1, коли вікно роботи підзавдання перекривається вікном наступної підзадачі в межах основної роботи задачі; інакше бітове значення дорівнює 0. Також бітове значення дорівнює 0 у останньої підзадачі в межах роботи цілої задачі. Підзадачам з бітовим значенням 1 надається пріоритет вищий ніж підзадачам з бітовим значенням 0.

$$b(T_n) = \begin{cases} 1, & \text{якщо } d(T_n) > r(T_{n+1}) \\ 0, & \text{якщо } d \leq r(T_{n+1}) \end{cases}$$

Підзадачам у яких перекриваються вікна надається пріоритет більше ніж підзадачам у яких вікна не перекриваються. Коли дві підзадачі мають однаковим псевдодедлайн і однакове бітове значенням, тоді треба враховувати вагу підзадач. Завдання вважається легким, якщо його вага менше 0,5, і важким, якщо його вага більше 0,5.

Дедлайн групи визначається формулою:

$$D(T_i) = \left\lceil \frac{\left\lfloor \frac{i}{T \cdot w} \right\rfloor - 1}{1 - T \cdot w} \right\rceil = \left\lceil \frac{d(T_i) - i}{1 - T \cdot w} \right\rceil$$

Основним недоліком такого алгоритму планування є його складність у реалізації та наступні великі витрати на планування через часте витіснення задач та їх міграцію. Ще один недолік який суттєво впливає на ефективність роботи процесора це те, що коли підзадача завершує виконання в першому слоті свого вікна і якщо в системі немає іншого завдання решта слотів першого вікна не будуть зайняті корисною роботою, оскільки в алгоритмі PD2 підзадача може бути розташована лише на початку свого вікна.

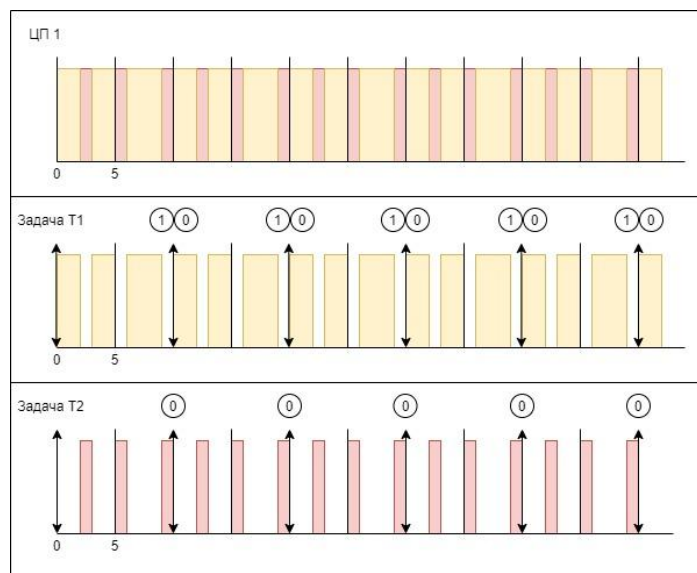


Рис. 1. Приклад алгоритму PD2

### Гібридний алгоритм EDF

Цей гібридний алгоритм (EDF – Earliest Deadline First) використовується для м'яких систем які мають ієрархічний спільний кеш (тобто група процесорів у системі, які спільно використовує загальний кеш). Цей підхід є комбінацією підходу розподіленого планування та глобального планування. Цей гібридний алгоритм ґрунтується на ідеї планування за найранішим дедлайном та має три основні кроки.

1. Система розділяється на кластери ядер.
2. Задачі розподіляються за допомогою розподіленого підходу планування.
3. В межах кластеру задачі плануються за допомогою алгоритму GEDF.

Таким чином гібридний алгоритм EDF це поєднання розподіленого планування та глобального планування алгоритму GEDF.

Переваги гібриду полягають у тому, що він використовує доступний кеш, оскільки міграція відбувається лише в межах спільного кешу, це сприяє зменшенню накладні витрат.

В основі модифікації алгоритму PD2 є ідея створення гібридного алгоритму для планування завдань у реальному часі на багатоядерних системах з ієрархічним спільним кешем. Завдання розподіляються в кластері ядер. Доступні ядра поділяються на кластери зі спільним кешем, як у гібридному EDF. Потім завдання плануються в кластері відповідно до реалізованого алгоритму планування Pfair (PD2). Завдання розподіляються між процесорами по принципу first-fit. Кластери заповнюються завданнями до їх максимальної потужності, яка визначається застосованою схемою планування. Кожного разу, коли потрібно виділити нове завдання, усі

непорожні кластери перевіряються на наявність, відповідно індексів. Тільки якщо жоден непорожній кластер не може вмістити нове завдання, вибирається новий кластер. Іншими словами, щоразу, коли виділяється завдання, всі непусті кластери перевіряються перед тим, як розподілити його в новий кластер. Максимальна ємність для кластера відповідно до алгоритму планування PD2 полягає в тому, що вага суми призначених завдань до кластера має бути менше, ніж кількість процесорів.

## **Висновки**

Існуючі алгоритми мають свою специфіку використання у тих чи інших ситуаціях. Кожен з алгоритмів має як сильні сторони так і недоліки. Отже було запропоновано новий гібридний алгоритм планування який поєднує в собі підходи як розподіленого планування так і глобального. Таким чином вдалося зменшити накладні витрати та витрати на планування задач.

## **Література**

1. THE REAL TIME SCHEDULER [Електронний ресурс] // SCHEDULING ALGORITHMS FOR FAULT-TOLERANT REAL-TIME SYSTEMS – Режим доступу до ресурсу: <https://docs.oracle.com/cd/E19120-01/open.solaris/817-4415/chap7rt-19493/index.html>
2. Philip A. Laplante Real-Time System Design and Analysis Tools for the Practitioner / Philip A. Laplante, Seppo J. Ovaska. // Space Weather. – 2012. – Volume 584.
3. RTOS [Електронний ресурс] // SpaceWeatherlive.com Real-time operating system: Components, Types, Examples – Режим доступу до ресурсу: <https://www.guru99.com/real-time-operating-system.html>
4. Erciyes K., Distributed Real-Time Systems // Int. J. Control. – 2019. – 569.
5. Hermann Kopetz, Real-Time Systems// Design Principles for Distributed Embedded Applications. – 2011. – 378.

УДК 004.93

К.т.н., доцент Клятченко Я.М., студент Ясенко Л.С.

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

## ПОРІВНЯЛЬНИЙ АНАЛІЗ ШВИДКОСТІ ВИКОНАННЯ АЛГОРИТМІВ НЕЙРОННИХ МЕРЕЖ ЗАЛЕЖНО ВІД АПАРАТНИХ ЗАСОБІВ

### Abstract

Yaroslav M. Klyatchenko, assoc. prof., PhD; Lev Yassenko, student  
*Comparative analysis of the speed of execution of neural network algorithms  
depending on hardware*

*This paper tests the performance of the image classification task using different hardware. The task is performed using popular frameworks and datasets. Existing equipment is used for image processing.*

### Вступ

Сучасні задачі штучних нейронних мереж (ШНМ) використовують величезні обсяги даних і відповідно потребують потужніших апаратних засобів (АЗ) для прискорення обробки цих даних.

Для виконання алгоритмів з використанням нейронних мереж (НМ) використовують центральні процесори (з англ. Central Processing Unit - CPU), графічні процесори (з англ. Graphics Processing Unit - GPU), і останнім часом, так звані тензорні процесори (з англ. *tensor processing unit* - TPU). Зазвичай потужності центрального вистачає для застосування уже натренованих нейронних мереж. Графічні процесори є більш популярними засобами для тренування і застосування нейронних мереж. Проте, зараз почали з'являтися спеціалізовані апаратні засоби для роботи, винятково, з нейронними мережами - тензорні процесори.

В даній статті представлені порівняльні результати швидкості тренування нейронної мережі на задачі класифікації зображень на різних апаратних засобах. Для нейронної мережі використовувалися популярні фреймворки PyTorch та Tensorflow. Серед апаратних засобів: CPU (intel core i5 7th Gen), GPU - Nvidia GTX 1050, та Google Cloud TPU[1].

### Апаратні засоби

Необхідно визначити параметри за якими можна порівнювати АЗ.

Для порівняння різних АЗ можна використовувати кількість операцій з плаваючою комою за секунду (з англ. Floating Point Operations Per Second - FLOPS)[2].

Для даного CPU - (FP32 performance) 422 GFLOPS, GPU - (FP16 performance) 29.10 GFLOPS, (FP32 performance) 1.862 TFLOPS, (FP64 performance) 58.20 GFLOPS. Для TPU ~ 420 TFLOPS[3-5]. Також для порівняння можна використати TDP (з англ. *thermal design power*) - кількість енергії, що може бути спожита при теоретичному максимальному навантаженні[6]. Для CPU - 45 Вт, GPU - 75 Вт, TPU - 600 Вт[3-5].

Представимо ці дані в наступній таблиці.

Таблиця 1

Таблиця характеристик АЗ

	CPU	GPU	TPU
FLOPS (FP32)	$0.422 * 10^{12}$	$1.862 * 10^{12}$	$420 * 10^{12}$
Power (W)	45	75	600
FLOPS/Power (1/W)	$9.3778 * 10^9$	$24.8267 * 10^9$	$700 * 10^9$

Отже чим більша кількість операцій з плаваючою комою, тим більше споживається енергії. Проте GPU споживає енергію ефективніше за CPU, а TPU ефективніше за GPU, для операцій з плаваючою комою.

На відміну від GPU і CPU, Google TPU використовує не ядра з арифметико-логічними пристроями (АЛП), а спеціалізовано розроблений Матричний Модуль. Схема TPU представлена наступним зображенням[7].

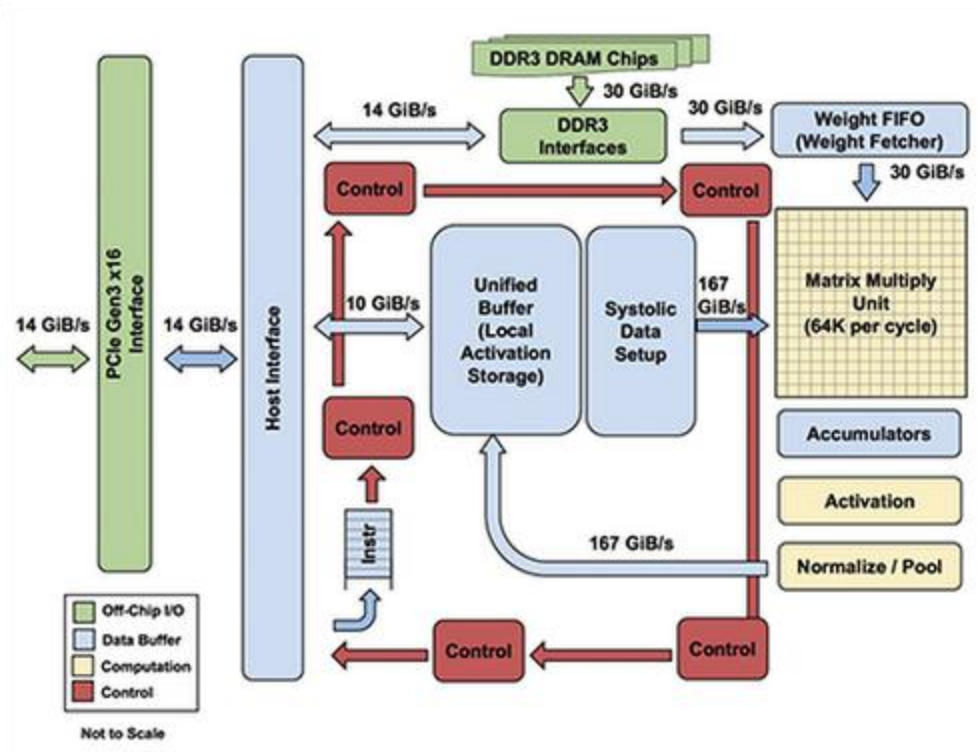


Рис 1. Схема тензорного процесора від Google.

## Застосування

Для перевірки було взято задачу класифікації зображень з використання популярного набору даних MNIST[1]. Для цієї задачі використовувався фреймворк Pytorch, оскільки рівень абстракції моделі НМ дозволяє переглянути як структуру НМ так і алгоритм навчання.

Predicted Lables



Рис 2. Приклад результату натренованої НМ на CPU.

НМ тренувалася на наборі MNIST впродовж 50 епох та показала високі результати акуратності передбачень. Під час тренування з використанням GPU був залучений CPU, оскільки дані з пам'яті завантажувалися за допомогою CPU, а обчислення вагів НМ відбувалися за допомогою GPU.

Результати тренувань представлені в наступній таблиці.



Таблиця порівняння споживання енергії

	CPU	GPU	TPU
Time	15 хв. 40,9 с.	5 хв. 2,3 с.	4 хв. 29,8 с.
accuracy	0.8794	0.8832	0.8919
Power * Time	42,34 кДж	36,27 кДж	161,8 кДж
Power	45	(+cpu) 120	600

Також було зроблено перевірку на задачі класифікації з перетворення рукописних цифр в цифрові за допомогою фреймворка Tensorflow. Проте даний фреймворк містить в собі деякі алгоритми оптимізації в результаті яких на CPU і GPU можна отримати результати набагато кращі ніж на TPU. Також треба зауважити, що програмне забезпечення(ПЗ) необхідно адаптовувати до АЗ, що постійно роблять розробники даних фреймворків.

### Висновки

Для прискорення розробок нових НМ, що повинні вирішувати поставлені задачі необхідно забезпечити належну швидкість їх тренування. Задачі НМ потребують окремих АЗ щоб знизити навантаження на CPU та GPU, а також виконуватись з меншими затратами енергії. Для цих задач створюють спеціалізовані АЗ серед яких присутні TPU. З результатів можна побачити, що на TPU тренування НМ відбулося швидше, проте затрати енергії були більшими, що може свідчити як про невідповідність ПЗ до АЗ, так і про необхідність покращувати АЗ.

### Джерела

1. «How To Implement CNN Model Using PyTorch With TPU», Url: <https://analyticsindiamag.com/how-to-implement-cnn-model-using-pytorch-with-tpu>
2. «Understanding Peak Floating-Point», Michael Parker, - 2017, Url: <https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/wp/wp-01222-understanding-peak-floating-point-performance-claims.pdf>
3. «Intel Core i5-7300HQ (characteristics)», Url: <https://www.cpu-panda.com/zh-cn/cpu-intel-core-i5-7300hq>

4. «NVIDIA GeForce GTX 1050 (characteristics)», Url:  
<https://www.techpowerup.com/gpu-specs/geforce-gtx-1050.c2875>
5. «TPU vs. GPU for GNN training», Xiangyang Ju, Lawrence Berkeley National Lab, Url:  
[https://indico.cern.ch/event/1015407/contributions/4261881/attachments/2209662/3739396/20210316\\_IML\\_TPUvsGPU.pdf](https://indico.cern.ch/event/1015407/contributions/4261881/attachments/2209662/3739396/20210316_IML_TPUvsGPU.pdf)
6. «Thermal Design Power (TDP) in Intel® Processors», - 2019, Url:  
<https://www.intel.com/content/www/us/en/support/articles/000055611/processors.html>
7. «An in-depth look at Google's first Tensor Processing Unit (TPU)», Kaz Sato, Cliff Young, David Patterson, - 2017, Url:  
<https://cloud.google.com/blog/products/ai-machine-learning/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu>

УДК 004:37

**К.т.н., доцент Марченко О.І., студент Вовчок О.В.**

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

## **СЕМАНТИЧНИЙ СПОСІБ ПОЄДНАННЯ ВЕРСІЙ ПРОГРАМНОГО КОДУ ПРИ ОДНОЧАСНІЙ РОЗРОБЦІ ПРОГРАМ ДЛЯ ПІДМНОЖИНИ МОВИ PYTHON**

### **Abstract**

**Oleksandr Marchenko, Assoc. Prof., PhD; Oleksii Vovchok, student**  
*Semantic software versions merge conflict resolution approach for  
a subset of Python language*

*This paper presents an approach for semantic software versions merging conflict resolution. This approach employs the semantic properties of the Python language and transforms its subset to the intermediate data. This data is transformed into a form that is supported by a program equivalence deducing algorithm. Also we provide refactoring quality criteria and an analyzer for summarizing the semantic impact of a modification. The implemented tool has been tested on the open-source projects by means of reproducing the conflicts and the percentage of correctly resolved conflicts has been evaluated.*

### **Вступ**

В даний час програмні продукти створюються групами розробників. Зазвичай, кожному розробнику видається своє окреме завдання, результатом якого є видозмінений програмний продукт. Розробники працюють незалежно і одночасно, а тому можуть вносити суперечливі зміни. Ці зміни унеможливають автоматичну інтеграцію незалежно від створених модифікацій в функціональність програмного продукту.

Попередні дослідження показали, що такі конфлікти трапляються часто і погіршують продуктивність розробки, оскільки їх розуміння та вирішення є складним і нудним завданням, причому неякісне виконання якого може внести дефекти в продукт [1], [2].

### **Семантичне поєднання**

При використанні семантичних способів поєднання, програмний код необхідно представити в структурах, що відображають семантичні

властивості даної мови. Зокрема, семантично орієнтовані способи аналізу коду застосовано в інструменті Semantic Diff [3]. Запишемо попередній приклад конфлікту однією з мов, що підтримуються цим інструментом, а саме мовою C, і застосуємо спосіб аналізу вказаного інструменту. В результаті отримуємо поєднання (рис. 1), яке виявляється семантично правильним завдяки наявності модуля в інструменті Semantic Diff, що відповідає за визначення семантичного ефекту модифікації. Одним з недоліків інструменту Semantic Diff є відсутність підтримки семантичного поєднання коду для мови Python.

```

1 int factorial(int n) {
2     /// Calculates the factor
3     int r = 1;
4     int i = 1;
5     while (i <= n) {
6         r = r * i;
7         i = i + 1;
8     }
9     return r;
10 }

1 int factorial(int n) {
2     /// Calculates the fact
3     int r = 1;
4     int i = 1;
5     if (n > 1) {
6         while (i <= n) {
7             r = r * i;
8             i = i + 1;
9         }
10    }
11    return r;
12 }

1 int factorial(int num) {
2     int r = 1;
3     int i = 1;
4     if (num > 1) {
5         while (i <= num) {
6             r = r * i;
7             i = i + 1;
8         }
9     }
10    return r;
11 }

```

Рис. 1. Результат поєднання на основі семантичних даних

### Опис запропонованого способу

У даній роботі пропонується спосіб семантичного поєднання для мови Python, подібний за загальною ідеєю способу, що використовується у Semantic Diff для інших мов програмування.

У загальному випадку результат вирішення конфлікту поєднання модифікацій одного й того ж коду двома програмістами – поєднаний модифікований код – може бути, зокрема, таких видів:

- додавання нової функціональності;
- стильове покращення – рефакторинг (англ. "refactoring").

Додавання нової функціональності обов'язково призводить до зміни результату виконання програми, тоді як рефакторинг змінює тільки текст програми без зміни результату її виконання.

З великою ймовірністю конфліктні фрагменти можуть одночасно містити будь-яку кількість модифікацій коду обох видів. Нехай маємо конфлікт поєднання, що складається з модифікацій в обох версіях деякого коду. Пропонується розбиття обох версій на часткові модифікації коду, що однозначно підпадають під один з видів, та поєднувати окремо кожну з них. Таким чином маємо окремі конфлікти поєднання, що обмежуються наступними випадками комбінацій видів модифікації коду:

- рефакторинг поєднується з рефакторингом. Такий конфлікт можливо розв'язати за наявності критерію якості рефакторингу. Для однозначного визначення яку із версій модифікації обрати, необхідні наперед задані пари фрагментів конструкцій мови програмування із однозначно вказаним фрагментом, якому надається перевага;
- нова функціональність поєднується з рефакторингом. Для розв'язання такого конфлікту, необхідно застосувати рефакторинг до модифікації, що змінює функціональність;
- нова функціональність поєднується з новою функціональністю. В такому випадку необхідно надавати можливість користувачу самостійно збирати обидві версії модифікації коду в одну за мінімальну кількість вручну виконуваних дій. Для цього застосуємо спосіб запропонований в [3] для виведення семантичних ефектів модифікації коду.

Для виконання рефакторингу виникає необхідність визначення еквівалентності коду. Зауважимо, що для системи повної за Тюрингом, - це алгоритмічно нерозв'язна задача [4]. Але якщо внести певні обмеження на функціональну повноту, то можна застосувати циклічний метод для визначення еквівалентності програм [5]. Для цього програми приводяться до проміжного представлення, збираючи сентенції, що не підпадають під вимоги циклічного методу.

### **Експериментальні дані**

Спочатку, на основі історії контролю версій певних програмних проєктів з відкритим кодом, взятих із системи контролю версій git, були відтворені конфлікти поєднання, що мали місце при побудові функціональності цих проєктів. Потім ці конфлікти були передані на автоматичну обробку та визначена кількість конфліктів, що були правильно розв'язані. Після чого, були підраховані відсотки автоматично розв'язаних конфліктів на цих проєктах:

- CPython <https://github.com/python/cpython> (83.2%)
- Django <https://github.com/django/django> (94.7%)
- Home Assistant <https://github.com/home-assistant/core> (98.1%)

### **Висновки**

Поєднання версій програмного коду, які виникають при його одночасній модифікації декількома розробниками є невід'ємною частиною

розробки програмного забезпечення. Тому ефективне рішення задачі автоматичного вирішення конфліктів при таких поєднаннях може значно допомогти розробникам в прискоренні процесу розробки програмного забезпечення та покращенні його якості. У даній роботі було запропоновано семантичний спосіб поєднання версій програмного коду при одночасній розробці програм для підмножини мови Python, який ґрунтується на наступних правилах:

- правила визначення якості рефакторингу;
- правила виведення семантичних ефектів модифікації;
- правила приведення до проміжного представлення для циклічного методу.

На основі експериментальних даних було показано ефективність інструменту поєднання, що працює за запропонованим способом.

## Література

1. Brun Y, Holmes R, Ernst MD, Notkin D (2013) Early detection of collaboration conflicts and risks. *IEEE Trans Softw Eng* 39:1358–1375
2. Kasi BK, Sarma A (2013) Cassandra: proactive conflict minimization through optimized task scheduling. In: *Proceedings of the 2013 international conference on software engineering. ICSE '13*. IEEE Press
3. D. Jackson and D.A. Ladd, “Semantic Diff: A Tool for Summarizing the Effects of Modifications,” *Int’l Conf. Software Maintenance*, 1994.
4. A. M. Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem", 1937.
5. D. Lucanu, V. Rusu, “Program Equivalence by Circular Reasoning”, 2012.

УДК 004.4

К.т.н., доцент Марченко О.І., магістрант Кравчук В.В.

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

## СПОСІБ ВИЗНАЧЕННЯ НЕЗАЛЕЖНИХ КОНСТРУКЦІЙ REDEFINES

### Abstract

Oleksandr I. Marchenko ; Kravchuk Volodymyr, student

*Method of determining independent use of REDEFINES fields*

*This paper discusses the problem of REDEFINES types usage. The data flow analysis algorithm for determining independently used REDEFINES fields is developed. It allows to find and show REDEFINES.*

### Вступ

Union-типи – це типи даних, які мають певний набір полів, де кожне поле вказує на одну й ту ж область пам'яті. В мові COBOL для підтримки таких типів використовується конструкція REDEFINES. Приклад використання REDEFINES показано на рис.1.

```
WORKING-STORAGE SECTION.  
01 A PIC X.  
01 B REDEFINES A PIC 9.
```

Рис. 1. Приклад використання REDEFINES.

У цьому прикладі оголошено поле A, яке є символом, та поле B, яке є числом, при цьому конструкція B REDEFINES A означає що поля A і B посилаються на одну й ту саму область пам'яті. Будь-яка зміна значення одного з цих полів змінить і значення іншого поля.

Використовуються такі структури зазвичай для досягнення двох цілей: 1) для економії пам'яті, щоб можна було зберігати поля різних типів даних в одній області пам'яті; 2) для швидкої конвертації даних, наприклад, можна зберігати дату одночасно і як рядок, і як структуру, яка має окремі поля для року, місяця, числа.

Проте сучасні комп'ютери мають достатньо потужності та пам'яті і тому більше уваги приділяється іншим аспектам програмування, наприклад структурованості та читабельності коду, що дозволяє легко

підтримувати та переносити на інші апаратні платформи. А використання типів даних, що використовують конструкцію REDEFINES, погіршує саме ці характеристики коду і збільшує імовірність допустити помилку під час програмування, яка може призвести до невизначеної поведінки програми, до того ж пошук таких помилок може бути не простим. Крім того, використання REDEFINES типів створює проблеми під час трансляції коду з мови COBOL у мови, які таких конструкцій не підтримують.

### **Постановка задачі**

Метою роботи є розробка способу статичного аналізу на базі способу DFA для визначення полів REDEFINES, які в програмному коді використовуються незалежно, що дозволить спростити ре факторинг коду та трансляцію коду в мови, які не підтримують union-типи, шляхом видалення REDEFINES.

### **Термінологія**

CFG – (англ. ControlFlowGraph) граф потоку керування – це множина всіх можливих шляхів виконання програми, що представлена у вигляді графу[1].

DFA – (англ. DataFlowAnalysis) аналіз потоку даних – це спосіб для визначення деякої інформації про дані, якими оперує код, в кожній точці програми[2].

Union-тип – це тип даних, який є об'єднанням кількох полів, що розміщені в одній області пам'яті.

REDEFINES – конструкція мови COBOL для створення union-типів.

### **Опис способу**

Першим кроком — є лексичний та синтаксичний аналіз вхідного коду мовою COBOL з побудовою дерева розбору (синтаксичного дерева) [3]. Для створення лексичного та синтаксичного аналізаторів використовується граматики мови COBOL та генератор аналізаторів ANTLR.

Після цього, із отриманого дерева розбору, потрібно побудувати CFG. CFG – це орієнтовний граф потоку керування програми, кожна вершина якого є базовим блоком. Базовий блок відповідає прямолінійній ділянці коду, що не містить в собі операцій передачі керування[3]. Спрямовані ребра графу використовуються для представлення інструкцій переходу.

Базові блоки реалізовані у вигляді структур, у яких міститься фрагмент синтаксичного дерева для прямолінійної ділянки коду та посилання на наступні блоки.



Після побудови CFG на ньому уже можна виконувати статичний аналіз DFA. Суттю цього аналізу є обхід усіх базових блоків та виконання для кожного блоку наступних дій:

$$out_b = transfer_b(in_b) \quad (1)$$

$$in_b = join_{p \in pred}(out_b) \quad (2),$$

де  $b$  – блок коду,  $in/out$  – набір даних на вході та виході із блоку,  $transfer$  – функція, яка виконує якісь обчислення на блоці і змінює там набір даних,  $join$  – функція, яка об'єднує набори даних з різних шляхів виконання. Реалізація функцій  $transfer$  та  $join$ , а також якими саме наборами даних вони оперують, залежить від конкретного алгоритму статичного аналізу.

Для визначення незалежного використання полів REDEFINES таким набором даних буде множина полів, яким було присвоєно якесь значення на попередньому кроці  $Write$ , та множина полів, значення яких було прочитане без попереднього запису  $ReadBeforeWrite$ . На початку аналізу для всіх базових блоків ці обидві множини порожні.

Потім запускається DFA і для кожної вершини обчислюються формули 1 та 2.

Функція  $transfer$  буде отримувати вхідні ( $in$ ) дані та виконувати наступні дії над кожним блоком:

- Визначення множини полів, яким було присвоєно значення  $CurrentWrite$ .
- Визначення множини полів, які були прочитані  $CurrentRead$ .
- Визначення множини полів, які були прочитані та не записані перед тим  $CurrentReadBeforeWrite$ . Це буде різниця множин  $Write(in)$  та  $CurrentRead$ .
- Збереження вихідних даних  $Write(out) = об'єднання CurrentWrite та Write(in)$ .
- Збереження вихідних даних  $ReadBeforeWrite(out) = об'єднання CurrentReadBeforeWrite та ReadBeforeWrite(in)$ .

Функція  $join$  буде виконувати об'єднання двох множин для всіх вхідних даних та зберігати вихідні дані.

Після завершення DFA необхідно виконати наступні кроки:

- Зібрати список всіх множин полів  $unionAllUnions$ . Наприклад, для коду на рис. 1 це буде список з однієї множини (A, B).
- Взяти вихідну множину  $ReadBeforeWrite$  і для кожного елемента списку  $AllUnions$  виконати об'єднання цього елемента з множиною  $ReadBeforeWrite$ .

- Якщо результатом є порожня множина, то це означає, що всі поля REDEFINES використовуються незалежно і цей елемент зберігаємо до *IndependentUnions*.

### **Результати роботи способу**

На виході отримуємо список *IndependentUnions*, де містяться всі поля конструкції REDEFINES, яких можна позбутися і поведінка програми не зміниться. Поля, які не увійшли до цього списку, використовуються залежно одне від одного і розділяти їх на окремі змінні не можна. Зібрана інформація про поля, які можна перетворити на звичайні змінні, та поля, які читаються перед записом, виводиться на екран для користувача.

### **Висновки**

Запропонований спосіб дозволяє визначати поля конструкцій REDEFINES, які використовуються незалежно, що дозволяє розробникам позбутися небажаного використання REDEFINES, що, в свою чергу, спрощує подальшу підтримку коду або дозволяє трансляцію коду з мови COBOL у мови, такі як Java, C# та інші, які не мають підтримки union-типів. Додаткова інформація може допомогти знайти місця в коді, де поле читається без попереднього запису. Це дозволить переписати код так, щоб конкретний union відповідав критеріям незалежного використання і теж міг бути розділений на звичайні змінні. Також варто зауважити, що показаний спосіб можна реалізувати й для інших мов, наприклад мови C, у якій також є union-типи.

### **Література**

1. *Frances E. Allen, Control Flow Analysis, July 1970*
2. О статическом анализе на чистоту [Електронний ресурс] – Режим доступу: <https://habr.com/ru/company/solarsecurity/blog/439286/> (дата звернення 17.10.2021).
3. Граф потока керування [Електронний ресурс] – Режим доступу: [https://uk.wikipedia.org/wiki/Граф\\_потоку\\_керування](https://uk.wikipedia.org/wiki/Граф_потоку_керування) (дата звернення 17.10.2021).

**УДК 004.02**

**К.т.н, доцент Марченко О.І., магістрантка Курдус А.О.**

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

## **RANDOM WALKS АЛГОРИТМИ**

### **Abstract**

**Anastasiia Kurdus, student**

***Random Walks Algorithms***

*The article describes the basic random walk algorithms. A comparative analysis of random walk algorithms by speed. Highlights the issues of efficiency, main memory and computation time of existing algorithms.*

### **Вступ**

Random Walks - це випадковий процес, який описує шлях, що включає послідовність випадкових кроків у математичному просторі. Він стає все більш популярним у різних дисциплінах, таких як математика та інформатика. Класичні та квантові алгоритми random walks можна використовувати для розрахунку наближеності між вузлами та вилучення топології у мережі. В загальному випадку, Random Walks алгоритми можуть бути застосовані в різних областях, що має велике значення для таких завдань, як передбачення зв'язків, рекомендації, комп'ютерний зір.

### **Постановка задачі**

Метою роботи є огляд класичних та квантових random walks алгоритмів, а також аналіз та виявлення відмінностей між цими алгоритмами.

### **Основна частина**

Суть random walks алгоритмів в тому, що описується траєкторія, яку утворюють послідовні випадкові кроки, які виконуються в якомусь математичному просторі. Прикладом, який пояснює, що таке random walk є випадкове блукання по числовій прямій цілих чисел. Воно починається в точці 0 і на кожному кроці зсувається на +1 або на -1 з однаковою ймовірністю [2]. Ще одним прикладом є броунівський рух і таких прикладів досить багато в реальному житті. Ці приклади показують, що

модель випадкового блукання (random walk) може використовуватися в різних галузях: екологія, психологія, інформатика, фізика, хімія, біологія, економіка і соціологія.

### Базові алгоритми

Серед базових random walks алгоритмів виділяють: Page Rank, Personalized Page Rank, RWR, LRW, B\_LIN та K-dash.

Page Rank – алгоритм оснований на розрахунку ваги веб-сторінки шляхом підрахунку документів, які посилаються на неї. Алгоритм оцінює важливість вершини графа на основі посилань в графі. Кожній з вершин призначається певне числове значення. Це значення визначає «важливість» або «авторитетність» цієї вершини серед інших [1]. Моделюється поведінка користувача, який випадковий чином переходить за певними посиланнями в графі. Користувач може не пройти по посиланню з ймовірністю  $\alpha$ , а перестрибнути на іншу випадкову вершину. Метод оцінює частоту потрапляння в кожен вузол. Чим більше посилань на сторінку, тим вона «важливіше». Формульно random walk алгоритм Page Rank можна записати так:

$$R_{t+1} = M^T R_t.$$

де,  $R$  - вектор PageRank,  $M^T$  це матриця ймовірностей переходу,  $t$  це кількість кроків .

Personalized PageRank – алгоритм майже в точності аналогічний попередньому алгоритму, за винятком того, що замість стрибка на випадкову вершину, користувач завжди потрапляє в заздалегідь задану вершину [2]. Формульно алгоритм Personalized PageRank виглядає так:

$$R_{t+1} = (1 - \alpha) M^T R_t + \alpha p.$$

де,  $\alpha$  - коефіцієнт загасання,  $p$  - персоналізований PageRank вектор, що відображає важливість кожного вузла на графіку для конкретного користувача.

Random Walk with Restart (RWR) – алгоритм використовується для реалізації можливості отримати рекомендації, які ґрунтуються на різноманітних даних соціальної мережі, таких як історія прослуханих треків, соціальні зв'язки, схожість смаків в різних темах і багато іншого. Ці дані представлені в алгоритмі у вигляді графа відносин між об'єктами, де об'єктами є користувачі та те, що їх цікавить (музика, фото, інша людина і т.д.). Ці об'єкти і зв'язки між ними є вершинами і дугами графа відповідно, ваги дуг визначаються силою зв'язності. Починаючи з вершини  $x$ , RWR на кожному наступному кроці виконує перехід з поточної вершини до наступної по дугам, причому ймовірність переходу прямо пропорційна вазі дуги. Важливо зазначити, що кожного разу з фіксованою ймовірністю

$\alpha$  обхід може повернутися в початкову вершину  $x$  [1]. Формульно цей алгоритм можна відобразити так:

$$p^{t+1} = (1 - a)Sp^t + aq$$

де,  $p^t$  вектор-стовпець, в якому  $p_i^t$  означає ймовірність того, що на кроці  $t$  RWR знаходиться в вершині  $i$ . Нехай  $q$  вектор-стовпець, в якому  $qx = 1$ , а решта елементи дорівнюють нулю, і  $S$  - нормалізована за стовпцями матриця суміжності графа.

Lazy Random Walk (LRW) – використовується для вирішення проблем сегментації зображення [1]. Особливістю цього алгоритму є те, що перехід завжди відбувається з ймовірністю  $1/2$ . Тобто ймовірність того, що ходунок знаходиться на поточній вершині завжди дорівнює  $1/2$ , але і ймовірність, що він перейде на сусідню вершину також рівна  $1/2$ . Формульно алгоритм виглядає так:

$$p_{t+1}(i) = 1/2(p_t(i)) + 1/2 \left( \sum_{j:(i,j) \in E} p_t(j) 1/d(j) \right)$$

$B\_LIN$  та  $K\_dash$  – це доволі нові алгоритми, які були створенні з метою покращення швидкості роботи алгоритмів random walks. Досягається ця мета за рахунок того, що ці алгоритми обчислюють близькість наступної вершини за допомогою розрідженої матриці, а також пропускає непотрібні обчислення під час пошуку найближчих вершин [2]. Теоретичний аналіз цих алгоритмів показує, що вони гарантують високу точність отриманих результатів при високій швидкодії.

### Алгоритми Quantum Walks

Quantum Walks є підрозділом random walks, але від класичних random walks ці алгоритми відрізняються тим, що не сходяться до обмежувальних розподілів, і через силу квантових перешкод вони можуть поширюватися значно швидше, ніж їх класичні еквіваленти. Алгоритми Quantum Walks умовно можна розділити на дві групи: Continuous-time Quantum Walk та Discrete Quantum Walk [3].

Continuous-time Quantum Walk – цю модель можна уявити на прикладі алгоритму дерева рішень. Систематично досліджується ціле дерево за ймовірнісним правилом. Розглядаються вузли дерева як квантові стани в просторі Гільберта [3]. Потім на основі отриманих результатів будується гамільтонова функція, яка допомагає дослідити еволюцію часу квантової системи. Прикладом алгоритму, який підпадає під опис цієї моделі є Quantum Decision Tree.

Discrete Quantum Walk – умовно кажучи, дискретні квантові прогулянки можна реалізувати шляхом заміни кожного стану класичного неорієнтованого графа на квантовий стан за допомогою використання

допоміжного стану “coin” для визначення напрямку, яким буде рухатися ходунок квантової прогулянки [3]. До цієї групи входить такий алгоритм, як Quantum PageRank.

## **Висновки**

У цій роботі представлено основні random walks алгоритми, як класичні так і квантові. Типовими класичними алгоритмами є PageRank, Personalized PageRank, RWR та LRW. Вони працюють досить довго при застосуванні до великих графів. Алгоритми B\_LIN та K-dash є альтернативою для попередніх алгоритмів, вони виконують обчислення швидше і дають досить точні результати. Також було розглянуто два типи квантових алгоритмів: безперервні квантові алгоритми та дискретні квантові алгоритми. Порівнюючи класичні та квантові random walks алгоритми, виявляємо, що по-перше квантові значно швидше можуть виконувати обчислення порівняно з їхніми класичними еквівалентами через силу квантових перешкод, а по-друге ці алгоритми відрізняються тим, що не сходяться до обмежувальних розподілів.

Random walks досить популярна і актуальна тема, адже ці алгоритми можна використовувати для обчислення близькості між двома вузлами та вилучення топології мережі. Досліджуючи застосування random walks алгоритмів у галузі інформатики, було виявлено багато проблем із існуючими реалізаціями алгоритмів, які виникають при роботі з гігантськими мережами. Ці проблеми полягають в повільній швидкості обчислень для великих графів, а також в недостатній ємності пам'яті. Подальші дослідження в цьому напрямку допоможуть у теоретичному та практичному покращенні random walks алгоритмів.

## **Література**

1. Feng Xia Random Walks: A Review of Algorithms and Applications, 2019.[https://www.researchgate.net/publication/337501163\\_Random\\_Walks\\_A\\_Review\\_of\\_Algorithms\\_and\\_Applications](https://www.researchgate.net/publication/337501163_Random_Walks_A_Review_of_Algorithms_and_Applications).
2. Jagers A. Random Walks, 2000.  
<https://www.sciencedirect.com/science/article/pii/0304414974900015>
3. Renato Portugal Quantum Walks and Search Algorithms, 2013.

УДК 004.4`4

К.т.н., доцент Марченко О. І., студент Чорний Є.Г.

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

## СПОСІБ ОЦІНЮВАННЯ ОБЧИСЛЮВАЛЬНОЇ СКЛАДНОСТІ АЛГОРИТМІВ ЗА КРИТЕРІЄМ ЧАСУ ВИКОНАННЯ АСЕМБЛЕРНИХ ІНСТРУКЦІЙ

### Abstract

**Oleksandr I. Marchenko, assoc. prof., PhD; Yehor Chorny, student**  
*Technique of computational complexity considering assembly instruction execution time*

*This paper concerns the task of estimating computational complexity of an algorithm considering assembly instruction execution time. The most popular way of computing computational complexity is discussed. The new technique of computing that gives quantitative rating as an additional point of view is proposed. Advantages and ways for further development are discussed, too.*

### Вступ

Розуміння складності певного алгоритму є важливим аспектом у процесі розробки будь-якого програмного забезпечення, адже завдяки цьому можна говорити про ефективність програмної системи та шукати шляхи її покращення. За допомогою визначення складності використаних алгоритмів, стає можливим виявлення проблем, пов'язаних з продуктивністю усієї програмної системи, ще до того, як вони фактично вплинуть на цю систему після її розгортання. У багатьох програмних проектах відсутня інформація, пов'язана з алгоритмічною складністю, тому розробнику важко оцінити продуктивність системи для різних розмірів вхідних даних. Тому, в даній доповіді розглядається можливість визначення складності алгоритмів саме з точки зору часу його виконання.

В даній статті запропоновано спосіб оцінювання часової складності алгоритмів за критерієм часу виконання асемблерних інструкцій.

### Мета дослідження та постановка задачі

Метою дослідження є аналіз уже існуючих способів оцінювання обчислювальної складності алгоритмів та розробка способу оцінювання обчислювальної складності алгоритмів за критерієм часу виконання асемблерних інструкцій.

## Термінологія

*Алгоритм* — набір інструкцій, які описують порядок дій виконавця, щоб досягти результату розв'язання задачі за скінченну кількість дій; система правил виконання дискретного процесу, яка досягає поставленої мети за скінченний час.

*Складність обчислювальних процесів* — це поняття теорії складності обчислень, оцінка ресурсів (зазвичай часу та пам'яті) необхідних для виконання алгоритму.

*Обчислювальна складність алгоритму* — складність алгоритму, яка описує час потрібний для виконання алгоритму в певних одиницях виміру.

*Асимптотичні позначення* — це математичні позначення, які використовуються для опису обчислювальної складності алгоритму, коли вхідні дані прагнуть до певного значення або граничного значення.

## Аналіз існуючих рішень

Зазвичай аналіз алгоритму включає в себе визначення функції, яка пов'язує довжину вхідних даних алгоритму з кількістю кроків, які він виконує (його складність за часом) або об'ємом пам'яті, який він використовує (його просторова складність). Алгоритм вважається ефективним, коли значення цієї функції малі або ростуть повільно, порівняно зі зростанням розміру вхідних даних.

Для визначення та опису обчислювальної складності алгоритму використовують поняття асимптотичних позначень. Найчастіше, це Big-O Notation (O-notation) — верхня межа часу роботи алгоритму. Таким чином, це дає найгірший випадок складності алгоритму. Саме такий випадок широко використовується для аналізу алгоритмів, оскільки нас завжди цікавить найгірший сценарій, тобто що для будь-якого значення  $n$  час роботи алгоритму не перетинає час, передбачений  $O(f(n))$ .

Розробники зазвичай вибирають найгірший сценарій, оскільки не очікується, що алгоритм працюватиме в найкращих або навіть середніх випадках під час обчислення часової складності алгоритму. Це дозволяє робити аналітичні твердження, наприклад, «у гіршому випадку, алгоритм швидко масштабується».

Окрім способу з використанням асимптотичних позначень (Big-O Notation), для розрахунку складності алгоритму використовують метрики, засновані на аналізі кількості рядків та синтаксичних елементів вхідного коду – метрика Холстеда, та метрика Мак-Кейба, заснована на показниках цикломатичної складності. Розглянемо їх детальніше.

В основі метрики Холстеда лежать чотири характеристики програми:

1. NUOprr – кількість унікальних операторів програми;
2. NUOprrnd – кількість унікальних операндів програми;
3. Noprtr – загальна кількість операторів програми;
4. Noprnd – загальна кількість операндів програми.



На основі цих характеристик розраховуються оцінки:

1. Словник програми:  $HPVoc = NUOprtr + NUOprnd$ ;
2. Довжина програми:  $HPLen = Noprtr + Noprnd$ ;
3. Об'єм програми:  $HPVol = HPLen \log_2 HPVoc$ ;
4. Складність програми:  $HDiff = (NUOprtr/2) \times (Noprnd/NUOprnd)$ .

Показник цикломатичної складності (метрика Мак-Кейба) відноситься до групи показників оцінки складності потоку управління програмою і обчислюється на основі графу логіки програми (control flow graph). Даний граф будується у вигляді орієнтованого графу, в якому обчислювальні оператори або вирази представляються у вигляді вузлів, а передача управління між вузлами – у вигляді дуг.

Спрощена формула обчислення цикломатичної складності:

$$C = e - n + 2, \quad (1)$$

де  $e$  – кількість ребер, а  $n$  – кількість вузлів на графі.

Як правило, при розрахунку цикломатичної складності логічні оператори не враховуються.

Існує значна кількість модифікацій показника цикломатичної складності.

1. Модифікована – розглядає не кожен гілку оператора множинного вибору (switch), а весь оператор як єдине ціле.
2. Строга – включає логічні оператори.
3. Спрощене обчислення передбачає обчислення не на основі графу, а на основі підрахунку управляючих операторів.

### **Спосіб оцінювання за критерієм часу виконання асемблерних інструкцій**

Способи (метрики) Холстеда та Мак-Кейба оцінюють код програм (який є текстовим поданням алгоритму) з інших точок зору, ніж швидкість роботи алгоритму, а результатом оцінювання обчислювальності складності за Big-O нотацією є функція або число, які є занадто узагальненими і не враховують специфіку процесора та мови програмування, на якій реалізовано алгоритм. Але різні мови програмування можуть транслювати один і той же алгоритм в різні набори асемблерних інструкцій. Також різні процесори можуть виконувати одні і ті ж інструкції за різний час.

В даній доповіді пропонується більш «низькорівневий» спосіб оцінювання обчислювальності складності алгоритмів за критерієм часу виконання команд асемблера.

Оскільки кожен алгоритм, написаний мовою програмування для виконання його процесором, буде перетворений в набір асемблерних інструкцій, то такий спосіб буде придатним для оцінювання будь-яких алгоритмів. Перевагою такого способу є те, що отримавши певний кількісний еквівалент набору інструкцій асемблера, який враховує час їх виконання, можна оцінити обчислювальну складність алгоритму з іншої,

більш «дрібнозернистої» точки зору. Іншою перевагою цього способу є можливість порівняння обчислювальної складності одного алгоритму для різних процесорів і різних мов програмування. Важливо зазначити, що даний спосіб є кількісним, тобто результатом оцінювання є число, яке розробник може легко порівнювати з іншими кількісними оцінками обчислювальної складності алгоритмів, отриманими даним способом.

## **Висновки**

Аналіз існуючих способів визначення обчислювальної складності алгоритмів показав, що є актуальною розробка більш способу оцінювання обчислювальної складності алгоритмів у більш дрібних одиницях вимірювання, який враховує критерій часу виконання асемблерних інструкцій. Хоча існуючі підходи є достатніми у переважній кількості ситуацій, запропонований спосіб дозволяє розглянути складність алгоритмів з іншої, більш детальної, сторони, що розширяє можливості вдосконалення та оптимізації програмного забезпечення.

На основі запропонованого способу може бути створений програмний інструмент, який дозволяє оцінювати складність алгоритму на рівні асемблерних команд до його впровадження та використання в програмному забезпеченні.

## **Література**

1. *Istvan Gergely Czibula, Zsuzsanna Onet-Marian, Robert-Francisc Vida. Automatic Algorithmic Complexity Determination Using Dynamic Program Analysis / Istvan Gergely Czibula, Zsuzsanna Onet-Marian, Robert-Francisc Vida //Proceedings of the 14th International Conference on Software Technologies (ICSOFT 2019). — 2019. — С. 186-193*
2. *Sipser, Michael. Introduction to the Theory of Computation. Course Technology / M. Sipser // Inc. ISBN 0-619-21764-2 — 2006.*

УДК 519.718

**К.т.н. Морозов К.В.; д-р техн. наук, проф. Романкевич О.М.;  
магістрант Білих О.С.**

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

## **ТЕСТОВИЙ ЕКСПЕРИМЕНТ ПО ВИЗНАЧЕННЮ СТАНУ ПРОЦЕСОРІВ БАГАТОПРОЦЕСОРНОЇ СИСТЕМИ**

### **Abstract**

**K.V. Morozov; A.M. Romankevich; O.S. Bilykh**

*Test experiment to determine the state of processors of a multiprocessor system*

*The work is devoted to experimental confirmation of a formal method for determining the state of processors of a multiprocessor system during testing. A multiprocessor system is formed on the basis of the Preparata-Metze-Chien model and is presented in the form of a graph, the vertices of which are the processors, and the edges are the connections between them. To carry out the work software was written. It generates an equation based on the graph model and solves it. The result is either a single solution that corresponds to the states of the system processors, or a set of solutions, among which there is necessarily a correct one.*

### **Вступ**

Однією з важливих проблем проектування багатопроцесорних систем є реалізація здатності системи до самотестування.

В [1] запропоновано метод визначення стану всіх процесорів  $M$ -діагностуємої [2] системи за результатами послідовних взаємних тестувань процесорів. Вважається, що систему побудовано в рамках моделі Препарати-Метца-Чена. Якщо дано багатопроцесорну систему, де кількість несправних процесорів не перевищує величини  $M$ , то можна однозначно визначити стани процесорів за результатами взаємного тестування її процесорів.

Для аналізу багатопроцесорну систему зручно описувати за допомогою орграфу, де процесори багатопроцесорної системи представлені вершинами цього графу. В такому випадку дуги графа представляють собою зв'язки між процесорами. Орієнтованість дуги вказує на те, які процесори, що відповідають вершинам графа, тестують та тестуються.

При одноразовому виконанні всіх послідовних перевірок, які відповідають дугам графа, отримується ряд результатів тестувань, що

можна представити у вигляді булевого рівняння. Його розв'язання дозволяє отримати або єдине рішення, або отримати множину рішень у вигляді досконалої диз'юнктивної нормальної форми (ДДНФ), яка гарантовано містить вірне рішення, що представляє собою елементарну кон'юнкцію. Операнди, що входять до неї без інверсій, відповідають справним процесорам, а всі інші - несправним.

У випадку, якщо отримано ДДНФ, що містить більш ніж одну кон'юнкцію та система є М-діагностованою, виконання додаткових тестів без змін топології системи дозволяє в кінці кінців отримати єдине рішення та однозначно визначити стани всіх процесорів системи.

### **Постановка задачі**

Задача полягає в експериментальній перевірці методу визначення стану процесорів М-діагностуємої системи.

Для цього необхідно побудувати програмний засіб, що має наступний функціонал: генерація графів довільної топології з заданою кількістю вершин та зв'язків; визначення максимального числа М несправних процесорів, при якій система може бути точно діагностована; генерація псевдовипадкових станів процесорів системи; проведення умовних взаємних тестувань процесорів відповідно до топології системи; отримання булевого рівняння та його спрощення; порівняння розв'язку рівняння з фактичним станом системи.

Використовуючи програмний засіб, провести ряд експериментів та визначити точність методу визначення стану процесорів, який запропоновано в [1].

### **Опис алгоритму тестування процесорів багатопроцесорної системи**

1. На вхід застосунку подається граф зі станами процесорів.
2. Ймовірністю правильного визначення стану процесору при тестуванні несправним процесором справного або несправного обрано 0.5.
3. Один раз проводяться всі можливі тести процесорів (відповідно до всіх дуг графу, що описує систему).
4. Отримане рівняння спрощується та розкривається з урахуванням правил поглинання та спрощення. Отримується єдина кон'юнкція або диз'юнкція кон'юнкцій;
5. У випадку, якщо рівняння звелось до єдиного рішення (єдиної кон'юнкції) - отримано формулу, що відповідає станам процесорів. Інверсія в елементі кон'юнкції означає, що відповідний процесор є несправним.

6. Якщо рішення не єдине - виконується перевірка, чи є серед рішень те, що відповідає дійсному стану системи.
7. Збереження результатів тестування та їх порівняння з дійсним станом системи.

### Опис алгоритму експерименту

1. Задається кількість вершин графу  $V$ , мінімальна кількість вхідних ребер  $I$  та вихідних ребер  $O$  для кожної вершини графу та кількість ітерацій  $E$  експерименту для заданих параметрів.
2. Задану кількість разів генерується випадковий граф з заданими параметрами. Для графу визначається максимальне число  $M$  несправних процесорів, при якому система може бути  $M$ -діагностуємою.
3. Для кожного графу  $M$  разів генеруються випадкові стани процесорів, де кількість несправних процесорів відповідає номеру ітерації (від 1 до  $M$ , відповідно).
4. Для кожного графу зі згенерованими станами процесорів запускається алгоритм взаємного тестування процесорів системи відповідно до зв'язків, що відповідають дугам графу, результат зберігається.

### Отримані результати

Таблиця 1

Результати експериментів

Вхідні дані				Результат			
V	I	O	E	Рішення єдине	Декілька рішень, серед яких є вірне	Система не $M$ -діагностуєма	Максимально можливе $M$ (кількість серед експериментів)
20	1	1	20	8	6	6	0 (6), 1 (14)
	1	2		15	3	2	0 (2), 1 (18)
	1	3		18	2	0	1
	1	4		25	0	0	1 (15), 2 (5)
	2	1		36	4	0	2

2	2	40	0	0	2
2	3	40	0	0	2
2	4	40	0	0	2
3	1	60	0	0	3
3	2	59	1	0	3
3	3	60	0	0	3
3	4	60	0	0	3
4	1	80	0	0	4
4	2	80	0	0	4
4	3	80	0	0	4
4	4	80	0	0	4
5	1	100	0	0	5

### **Аналіз результатів**

Отримані результати тестувань для різних топологій графів при різних заданих параметрах поділяються на три варіанти:

1. Система, якій відповідає граф, може бути не М-діагностуємою, якщо кількість вхідних зв'язків кожного процесору є недостатньою.
2. Отримано однозначне рішення.
3. Отримано множину рішень, серед яких є те, що відповідає дійсному стану системи.

### **Висновки**

На базі перевірки формального методу визначення станів процесорів М-діагностуємої багатопроцесорної системи на основі моделі Препарати-Метца-Чена шляхом представлення результатів тестувань в якості

булевого рівняння та його рішення, було експериментально підтверджено в рамках об'єму експерименту те, що розв'язання такого рівняння дійсно дозволяє отримати формулу, котра містить або єдине рішення, елементи якого відповідають стану системи, або їх множину, серед якої обов'язково є вірне. В такому випадку, проведення додаткових тестувань дозволяє отримати єдине вірне рішення.

Перспективними для наступних робіт є, зокрема, питання перевірки ймовірності здатності системи до самодіагностування при кількості несправних процесорів більше, ніж  $M$ . а також знаходження оптимальних шляхів організації додаткових тестувань у випадках, коли після перетворення виразу отримано СДНФ, що містить більш ніж одну елементарну кон'юнкцію.

## Література

1. *A. M. Romankevich, K. V. Morozov, V. A. Romankevich*, “A formal method for determining the state of processors in a multiprocessor system under testing”, *Autom. Remote Control*, 82:3 (2021), 460–467
2. *Hakimi S.L., Amin A.T.* Characterization of Connection Assignment of Diagnosable Systems // *IEEE Trans. Comput.* 1974. C-23. No. 1. P. 86–88.
3. *Preparata F.P., Metze G., Chien R.T.* On the Connection Assignment Problem of Diagnosable Systems // *IEEE Trans. Electron. Comput.* 1967. ES-16. No. 6. P. 848– 854.

УДК 519.688

**Старший викладач, к.т.н. Наливайчук М.В.,  
студент Шостак М.С.**

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

## **АНАЛІЗ СТАТИСТИЧНИХ ДАНИХ ІНТЕРНЕТ РЕСУРСІВ ІЗ ВИКОРИСТАННЯМ АЛГОРИТМУ МАШИННОГО НАВЧАННЯ LINEAR REGRESSION**

### **Abstract**

**Mykola Nalyvaichuk, senior lecture; Mykhailo Shostak, student**  
*Analysis statistical data of internet resources by using the linear regression machine  
learning algorithm*

*This article discusses the use of a linear regression machine learning algorithm as a powerful tool for  
predicting linear relationships of various aspects of Internet resources.*

### **Вступ**

Використання штучного інтелекту, хмарних обчислень і машинного навчання дає змогу власникам інтернет-ресурсів збирати та обробляти статистичні дані на різних веб-сайтах для подальшого аналізу. Завдяки вивченню статистичних даних власники інтернет-ресурсів мають можливість використовувати різноманітні програмні рішення для просування та оновлення власних інтернет-ресурсів.

Усі ці рішення дозволяють власникам інтернет-ресурсів утримувати та залучати нові цільові аудиторії та розвивати певні аспекти власних інтернет-ресурсів на основі статистичних даних та моделей активності користувачів.

### **Постановка задачі**

Задача полягає у розробленні комплексного програмного рішення із використанням алгоритму машинного навчання LinearRegression, яке буде мати можливість на основі отриманих статистичних даних аналізувати та створювати модель прогнозу різних аспектів інтернет ресурсів, таких як імовірні обсяги продажі товарів, прогнозування активності користувачів інтернет ресурсів та інше.



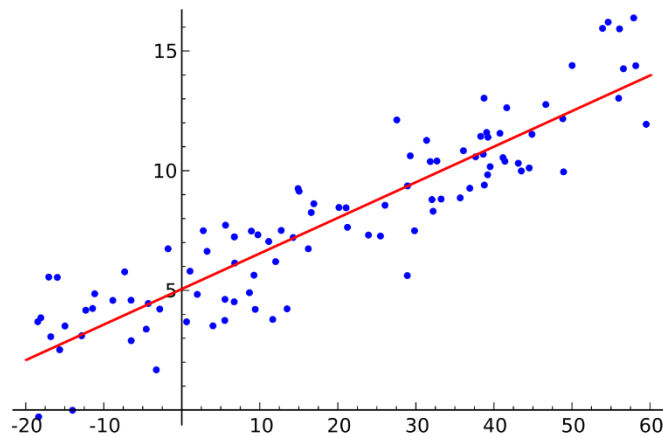
## Термінологія

Інтернет – ресурс - це набір інтегрованих засобів технології та програмно-апаратного характеру, і призначений для використання у всесвітній мережі Інтернет. Ресурси Інтернету можуть містити різну інформацію у вигляді тексту, мультимедіа, графіки.

Алгоритми машинного навчання – галузь штучного інтелекту в програмуванні, яка використовує статистичні методи, щоб дозволити комп'ютерам «вчитися» за допомогою отриманих даних (тобто поступово покращувати виконання даного завдання) без явного програмування.

## Опис алгоритму LinearRegression

Лінійна регресія — це одна з найпростіших форм машинного навчання, в якій ми навчаємо моделі передбачати поведінку вхідних даних на основі певних змінних. У випадку лінійної регресії, число незалежних змінних дорівнює одиниці, а між незалежною ( $x$ ) і залежною ( $y$ ) змінними існує лінійна залежність. Червона лінія на графіку нижче називається прямою лінією. Грунтуючись на координатах точок, ми намагаємося побудувати лінію, яка найкращим чином буде моделювати точки.



Лінія може бути змодельована на основі даного лінійного рівняння:

$$y = a_0 + a_1x$$

Де  $a$  і  $b$  дано за формулами:

$$a_1 = \frac{n \sum xy - (\sum x)(\sum y)}{n \sum x^2 - (\sum x)^2}$$
$$a_0 = \frac{n \sum y - b(\sum x)}{n}$$

Де:

- $a_1$  – нахил лінії.
- $a_0$  –  $y$ -точка перетину лінії.
- $x$  – незалежна змінна з набору даних на лінії регресії.
- $y$  – залежна змінна з набору даних на лінії регресії.

Приклади застосування лінійної регресії:

- Прогнозування тенденцій або цілей із продажу – передбачити, наскільки популярна галузь або скільки цілей із продажу галузь може досягти в майбутньому.
- Прогноз активності – використання регресії для прогнозування зміни активності користувачів.
- Управління ризиками – використання регресії для аналізу управління ризиками у фінансовому та страховому секторі.

### **Функція витрат**

Функція вартості допомагає визначити найкращі можливі значення для  $a$  і  $b$ , що забезпечує найкращу відповідність лінії для точок даних.

Функція витрат оптимізує коефіцієнти або вагові коефіцієнти регресії та вимірює ефективність моделі лінійної регресії. Функція вартості використовується для визначення точності функції відображення, яка співставляє вхідну змінну з вихідною. Ця функція відображення також відома як функція гіпотези.

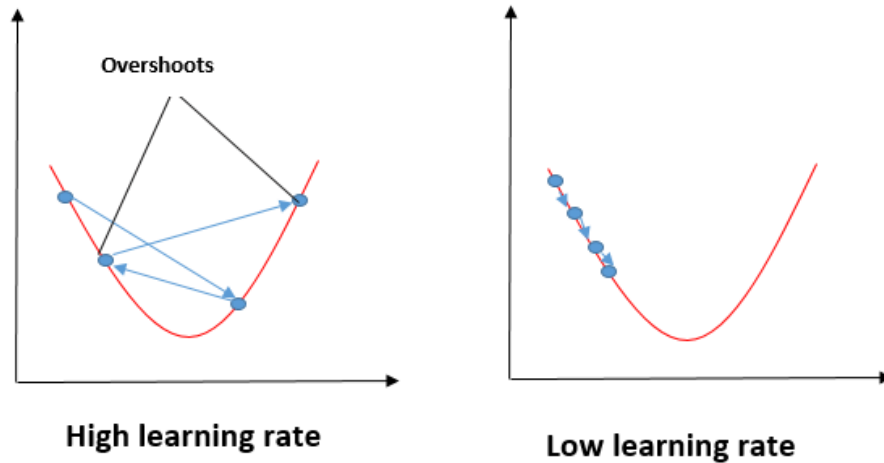
У лінійній регресії використовується функція вартості середньої квадратичної помилки (MSE), яка є середнім квадратом помилки, що виникла між прогнозованими і фактичними значеннями.

$$MSE = \frac{1}{N} \sum_{i=1}^n (y_i - (mx_i + b))^2$$

Використовуючи функцію MSE, ми змінюємо значення  $a_0$  і  $a_1$ , щоб встановити значення MSE на мінімальне значення. Параметрами моделі  $x_i$ ,  $b$  ( $a_0$  і  $a_1$ ) можна маніпулювати, щоб мінімізувати функцію вартості. Ці параметри можна визначити за допомогою методу градієнтного спуску, щоб зберегти значення функції вартості до мінімуму.

### **Градiєнтний спуск**

Градiєнтний спуск – це метод оновлення  $a_0$  і  $a_1$  для мінімізації функції вартості (MSE). Регресійна модель використовує градієнтний спуск для оновлення коефіцієнтів лінії ( $a_0$  і  $a_1 \Rightarrow x_i$ ,  $b$ ) шляхом випадкового вибору значень коефіцієнтів для зменшення функції вартості, а потім ітеративно оновлюючи значення для досягнення функції мінімальних витрат.



Щоб оновити  $a_0$  і  $a_1$ , ми беремо градієнти з функції вартості. Щоб знайти ці градієнти, ми беремо часткові похідні для  $a_0$  і  $a_1$ .

$$a_0 = a_0 - \alpha \cdot \frac{2}{n} \sum_{i=1}^n (pred_i - y_i)$$

$$a_1 = a_1 - \alpha \cdot \frac{2}{n} \sum_{i=1}^n (pred_i - y_i) \cdot x_i$$

$$J = \frac{1}{n} \sum_{i=1}^n (pred_i - y_i)^2$$

$$J = \frac{1}{n} \sum_{i=1}^n (a_0 + a_1 \cdot x_i - y_i)^2$$

$$\frac{\partial J}{\partial a_0} = \frac{2}{n} \sum_{i=1}^n (a_0 + a_1 \cdot x_i - y_i) \implies \frac{\partial J}{\partial a_0} = \frac{2}{n} \sum_{i=1}^n (pred_i - y_i)$$

$$\frac{\partial J}{\partial a_1} = \frac{2}{n} \sum_{i=1}^n (a_0 + a_1 \cdot x_i - y_i) \cdot x_i \implies \frac{\partial J}{\partial a_1} = \frac{2}{n} \sum_{i=1}^n (pred_i - y_i) \cdot x_i$$

Часткова похідна — це градієнт, який використовується для оновлення значень  $a_0$  і  $a_1$ . Альфа - це швидкість навчання.

## Висновки

Лінійна регресія показує лінійний зв'язок між незалежною змінною (вісь  $x$ ) та залежною змінною (вісь  $y$ ). Для розрахунку лінійної регресії найкращої відповідності використовується традиційна форма перетину нахилу. Лінія регресії може бути позитивною лінійною залежністю або негативною лінійною залежністю

Мета алгоритму лінійної регресії – отримати найкращі значення для  $x$  і  $y$ , щоб знайти лінію найкращої відповідності, а лінія найкращої відповідності повинна мати найменшу помилку.

Алгоритм лінійної регресії надає потужний статистичний метод для пошуку взаємозв'язку між змінними. Також він не потребує доопрацювання, однак обмежується лише лінійними відносинами між змінними.

Слід зауважити, що лінійна регресія забезпечує найкращу точність прогнозів для лінійних відносин, тоді як вона мало чутлива до викидів і враховує лише середнє значення залежної змінної.

## Література

1. Introduction to Machine Learning Algorithms: Linear Regression [Електронний ресурс]. - Режим доступу: <https://www.machinelearningmastery.ru/introduction-to-machine-learning-algorithms-linear-regression-14c4e325882a/>
2. A Guide to the Linear Regression Algorithm [Електронний ресурс]. - Режим доступу: <https://towardsdatascience.com/introduction-to-machine-learning-algorithms-linear-regression-14c4e325882a>
3. Article «5 Machine Learning Techniques for Sales Forecasting» [Електронний ресурс]. - Режим доступу: <https://towardsdatascience.com/5-machine-learning-techniques-for-sales-forecasting-598e4984b109> – Дата доступу: травень 2020.
4. Bohdan M. Pavlyshenko. "Machine-Learning Models with using LinearRegression Algorithms" [Текст] - SoftServe, Inc., 2017. - стор. 24-27.

Магістрант Несук О.О., к.т.н. доцент Потапова К.Р.,  
асистент Кучмій О.О.

Національний технічний університет України  
"Київський політехнічний інститут імені Ігоря Сікорського"

## СПОСІБ ПРОГРАМНО ПРИСКОРЕНОГО ВІЗУАЛЬНОГО ПОДАННЯ ІНФОРМАЦІЇ

### Abstract

Oleksandr Nesuk, student; Kateryna Potapova, assoc. prof., PhD;  
Kuchmii Oksana, assistant

#### *A Way Of Software Accelerated Visual Presentation of Information*

*This article presents a way of rendering graphical objects using inverted rendering method. This deals with overdraw problems that the standard rendering methods have. By rendering images in the backwards order and trimming out areas that already possess drawn objects. This method is effective when the program is drawing large amounts of objects every frame.*

### Вступ

Наразі досить велике значення набуває питання раціонального використання графічного ресурсу обчислювальних систем. Стандартні алгоритми рендерингу зображень вже не у повній мірі відповідають сучасним вимогам з огляду на велику кількість і складність операцій із графічними об'єктами.

Є проблема марної витрати графічного ресурсу внаслідок появи множини пікселів, які вповільнюють частоту зміни кадрів. У цій роботі розглядається зворотний алгоритм рендерингу, завдяки якому мінімізується кількість операцій із графічними об'єктами без втрати візуальної складової "картинки".

### Постановка задачі

Наразі стандартний алгоритм рендерингу може уповільнювати себе при великій кількості перезаписів пікселів або *overdraw*. Це призводить до уповільнення частоти зміни кадрів. При візуалізації графічних об'єктів у зворотному порядку можна знизити кількість перезаписів стеку до мінімальних значень, у той же час отримуючи картинку ідентичну до такої, яку створить стандартний алгоритм.

## Термінологія

*Рендеринг* – процес отримання зображення за допомогою комп'ютерної програми.

*Піксель* – найдрібніша одиниця цифрового зображення в растровій графіці.

*Стандартний алгоритм рендерингу зображень* – алгоритм представлення зображень, при якому об'єкти нашаровуються один на одного, створюючи деяку композицію.

*Зворотний алгоритм рендерингу зображень* – алгоритм представлення зображень, при якому об'єкти накладаються на простір починаючи із найближчого. Кожен об'єкт обрізається по контуру вже представленого зображення.

## Матеріали та методи

За *стандартного алгоритму* нашаровування декількох об'єктів призводить до перезапису інформації у стеку, тобто *overdraw*. На Рис.1 показано зображення та області де графічна інформація була замінена. Темніші області зображення з правої сторони вказують на втрату ефективності через *overdraw*. Це знижує швидкість виведення кадрів. При більшій кількості об'єктів, які перекривають одне одного, час витрачений на компонування кожного наступного кадру збільшується.

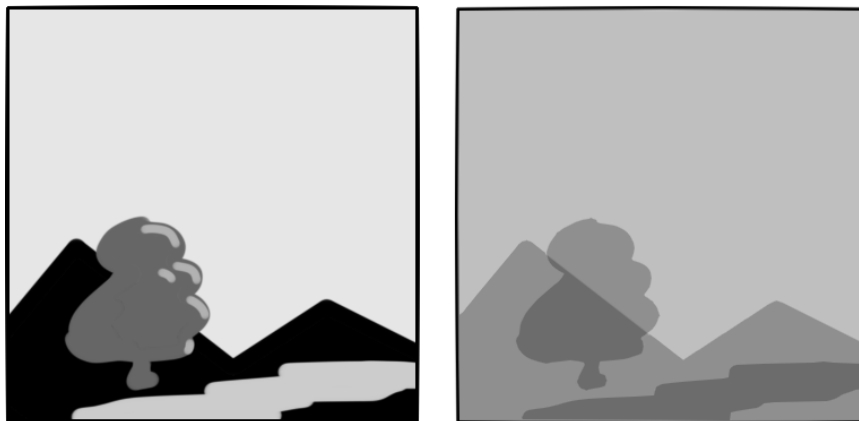


Рис. 1. Зображення (зліва), та області в яких проходить перезапис графічної інформації (справа)

*Зворотний алгоритм* передбачає обрізання частини об'єктів, які перекриваються вже виведеним зображенням. Це дозволяє виводити графічну складову швидше за стандартний алгоритм, бо виконується менша кількість операцій по зміні піксельної інформації у стеку.

Порівнявши цей алгоритм із стандартним, ми бачимо, що у стандартному алгоритмі сума всіх пікселів, які вводяться у стек, вперше

раз або у разі перезапису, дорівнює  $\sum_0^n PX_i$ , тоді як у зворотному ця сума становить  $\sum_0^n P_i$ , де:

$$P_i \leq PX_i \quad (1)$$

Важливо зауважити, що під час роботи зворотний алгоритм витрачає більше операцій на запис пікселя у стек через перевірку чи є поточне місце пікселя вільним.

Для здійснення моделювання використано мову C++ та бібліотеку *freelut*. Код, представлений на Рис.2, виконує обробку зображення для його подальшого представлення у зворотному порядку.

```
for(int y=0; y<pic[i].h; y++){
    bool flag=false;
    for(int x=0; x<pic[i].w; x++){
        xsh=pic[i].x+x;
        ysh=pic[i].y+y;
        while(shift[xsh][ysh]>0){
            x+=shift[xsh][ysh];
            xsh=pic[i].x+x;
            if(x>=pic[i].w){
                flag=true;
                break;
            }
        }
        if(flag){
            break;
        }
        draw_alpha(xsh, ysh, i);
        shift[xsh][ysh]=pic[i].w-x;
    }
}
```

Рис. 2. Фрагмент коду, який обрізає зображення об'єкта по контуру вже виведеної частини картинки

Для ефективної роботи алгоритму, усі об'єкти заносяться в буфер у порядку близькості кожного з них до камери. Після цього виводиться перший об'єкт. Одразу за ним, представляється кожен наступний об'єкт, дані зображення якого оброблюються, для недопущення перезапису клітинки пам'яті у стеку повного зображення. Алгоритм обходить таким чином кожен піксель зображення об'єкта, та записує у стек тільки ті елементи зображення які не будуть перезаписувати інформацію у стеку зображення.

Після закінчення проходу по всьому буферу, повний кадр виводиться на екран. Для підвищення ефективності перевірки, у матрицю цілих ненульових чисел, сторони якої відповідають розмірам програмного простору, заносимо значення зміщення, що дозволить перемістити лічильник поточного піксель об'єкта на вільне місце у стеку зображення.

## Результати

Отримані результати роботи зворотного алгоритму (Рис. 3) показали ефективну роботу алгоритму як при малій кількості об'єктів у програмному полі, так і при великій їх кількості.

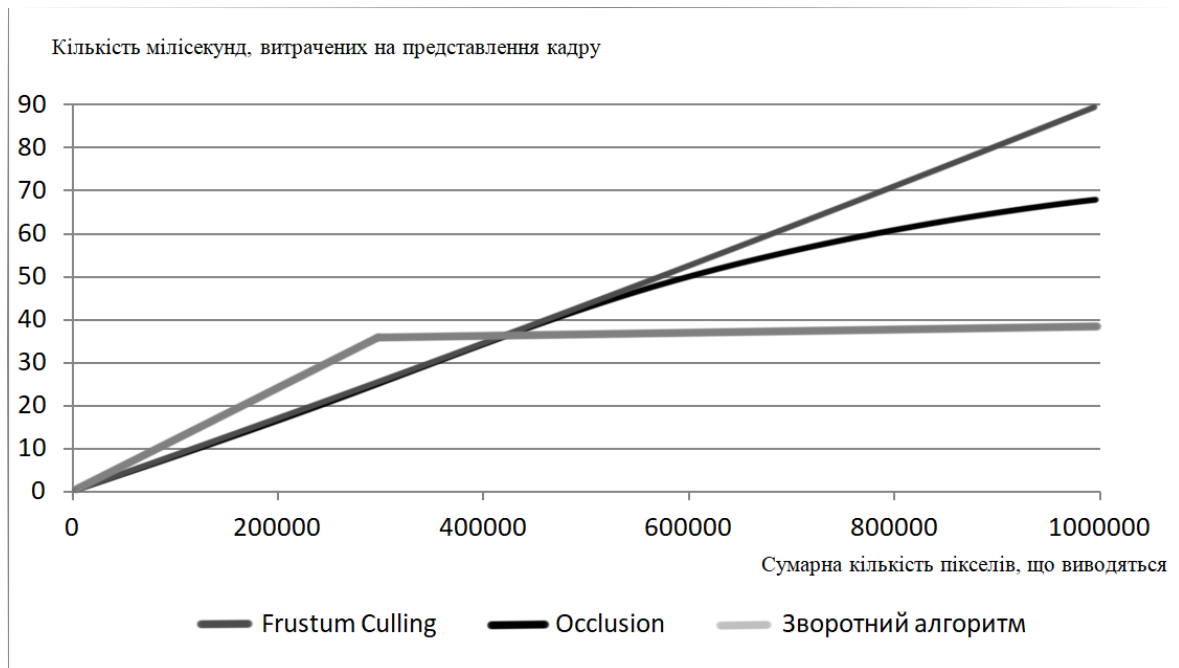


Рис. 3. Порівняльний графік зворотного алгоритму та стандартного з точки зору використання ресурсів за різної кількості виведених пікселів

Зворотний алгоритм має значну перевагу над стандартним з точки зору витраченого часу на проведення операції зміни кадру, при умові, що кількість перезаписаних пікселів стандартним алгоритмом була більша за ~30% від їх сумарної кількості на площині дисплея.

## Висновки

Підсумовуючи вищенаведене, зворотний алгоритм є значно більш ефективним і швидким за стандартні алгоритми. Завдяки його використанню може підвищуватись швидкодія і відбуватись раціональне використання графічного ресурсу системи. З огляду на це, зворотний алгоритм є доцільним для використання щодо програм, які виводять велику кількість об'єктів, що перекривають частини інших.

В залежності від поточного стану та параметрів є можливість змінювати алгоритм рендерингу під час роботи.



## Література

1. *Mathematics for Computer Graphics* (Undergraduate Topics in Computer Science) – <https://www.amazon.com/Mathematics-Computer-Graphics-Undergraduate-Science/dp/1447173341/>
2. *Mathematics for 3D Game Programming and Computer Graphics*, Third Edition / Eric Loney – <https://www.amazon.com/Mathematics-Programming-Computer-Graphics-Third/dp/1435458869/>
3. *Occlusion Culling*: URL: <http://121.196.216.106/UnityDoc/Manual/OcclusionCulling.html>
4. *GLUT documentation*: URL: <https://www.opengl.org/resources/libraries/glut/spec3/node1.html>
5. *OpenGL documentation* URL: <https://www.opengl.org/documentation/>

**К.т.н., доцент Орлова М.М., студентка Баглай І.Ю.**

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

## **ЕФЕКТИВНИЙ АЛГОРИТМ ДЛЯ ЗАДАЧІ УПАКОВКИ 3D-ПРЯМОКУТНИХ БЛОКІВ**

### **Abstract**

**Mariya Orlova, assoc. prof., PhD; Bahlai Ivanna, student**  
***An Efficient Algorithm for 3D-Rectangular Box Packing***

*Getting highest occupancy rate of capacity of a container is very important for the companies, which deals in shipping or has shipping as a part of their main activities. They have to fit 3D-boxes in container with optimum or nearest to optimum placement in order to ship more products with a minimum cost. The problem of fitting the boxes which is different from or the same to each other into a big container in optimum level, is called 3-dimensional packing problem. In this problem, the main objective is to minimize used container volume or wasted container space. This provides the reduction of costs in shipments with the use minimum number of containers.*

### **Вступ**

Проблема оптимального заповнення контейнера, з точки зору відсотку не заповненого простору, що залишиться після упаковки, є важливою для компаній, що займаються перевезенням. Їх завдання полягає у розміщенні 3D-блоків у контейнери оптимально, щоб в результаті доставити більше товарів, використовуючи при цьому мінімальну кількість контейнерів, що забезпечує зниження витрат на перевезення.

Проблема розміщення блоків, що можуть бути однаковими або різними за розмірами, у велику ємність оптимальним чином, називається задачею тривимірної упаковки, головною метою якої є мінімізація невикористаного (не заповненого) простору контейнера. Проблема 3D-упаковки є NP-складною задачею. Якщо кількість блоків збільшується то час необхідний для упаковки блоків у контейнер зростає недетерміновано поліноміально [1].

Проблема 3D-упаковки, зазвичай, вирішується за допомогою методів оптимізації та евристики. Деякі дослідження передбачають необмежену місткість контейнера і намагаються вирішити проблему 3D-упаковки за

допомогою однакових або різних за розміром блоків [2]. Деякі алгоритми упаковки враховують інші додаткові параметри блоків, такі як їх цінність, і вони мають більший пріоритет при їх розміщенні у контейнер [3]. Крім того, деякі алгоритми вважають один з вимірів необмеженим, а інші – фіксованим [3].

### **Постановка задачі**

Метою роботи є аналіз оптимального алгоритму для задачі упаковки 3D-блоків у контейнер та оцінка його ефективності за параметром мінімізації не заповненого простору в контейнері, що залишиться після упаковки.

### **Опис роботи алгоритму LAFF**

Проаналізовано найбільш поширений алгоритм Largest Area First-Fit (LAFF), який використовує евристику, що спочатку розміщує блоки з найбільшими сторонами (ширина і глибина) для мінімізації висоти від дна контейнера. Вважається, що висота контейнера є необмеженою, а його ширина і глибина вибирається враховуючи розміри блоків, які необхідно розмістити в ньому. Алгоритм працює наступним чином.

Спершу визначається ширина і глибина контейнера. Враховуючи набір блоків різного розміру, обчислюється ширина та глибина контейнера, знаходячи два найдовші розміри цих блоків. Ширина і глибина контейнера визначаються на початку алгоритму і залишаються фіксованими протягом усього виконання алгоритму. Вважається, що висота необмежена. Висота збільшується під час роботи алгоритму. Тобто, ширина ( $a_k$ ) та глибина ( $b_k$ ) контейнера визначаються шляхом вибору першого та другого найдовшого краю заданих блоків ( $a_i$ ,  $b_i$  та  $c_i$ ). За ширину ( $a_k$ ) приймають найдовшу сторону, а за глибину ( $b_k$ ) – другу найдовшу сторону. Отже,  $a_k$  і  $b_k$  ніколи не змінюються. Після визначення значень ширини та глибини контейнера, ці блоки можна помістити в контейнер. У алгоритмі використовуються два способи розміщення.

Перший спосіб виділяє простір для блоку, що збільшує висоту контейнера. Другий спосіб розміщення виділяє простір для решти блоків, якщо блок поміщається у вільний простір навколо розміщеного блоку, не переповнюючи висоту розміщеного блоку. У першому способі для розміщення вибираються блоки з найбільшою площею поверхні (добутком ширини і глибини), а потім у вибраних блоках здійснюється пошук, щоб знайти поле з мінімальною висотою з усіх вибраних блоків. Потім блок з

мінімальною висотою поміщають у контейнер. Найбільша поверхня вибраного блока повинна бути паралельною дну контейнера (рис. 1).

При реалізації другого способу розміщення намагаються виділити простір для решти блоків навколо того, що був розміщений за допомогою першого способу. Тобто, якщо навколо розміщеного блоку залишається деякий простір, як показано на рис. 1, намагаються заповнити цей простір за допомогою другого способу розміщення. При цьому враховуючи, що розміщений блок має розміри  $a_i$ ,  $b_i$  та  $c_i$ , контейнер матиме два порожніх простори з розмірами  $((a_k - a_i), b_k, c_i)$  та  $(a_k, (b_k - b_i), c_i)$  навколо вже розміщеного блоку. Отже, якщо є в наявності кілька блоків, що можна помістити у цей порожній простір, то вставляється блок з максимальним площею, як показано на рис. 2.

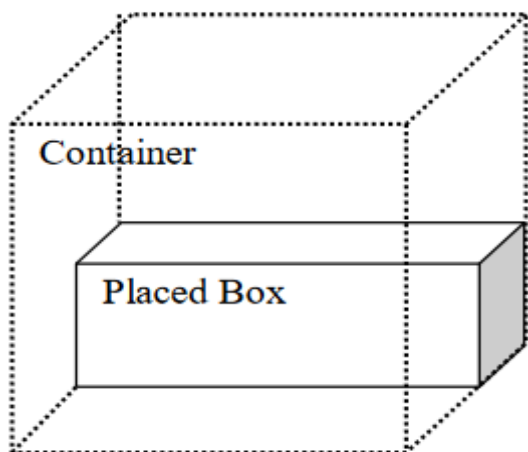


Рис. 1. Перший спосіб розміщення блоків

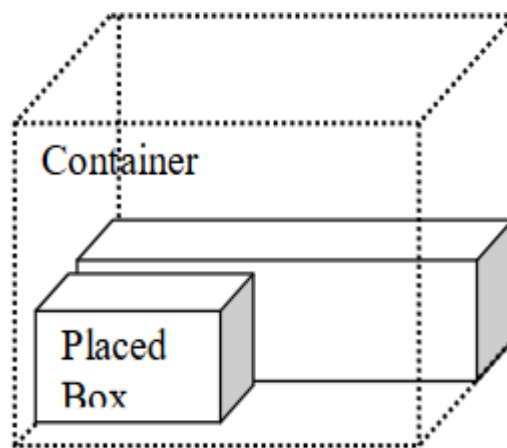


Рис. 2. Другий спосіб розміщення блоків

Ця ітерація триває до тих пір, поки є блоки, що можуть поміститися у вільний простір, або поки не розміщено всі блоки. У другому способі розміщення, якщо навколо розміщеного блоку немає простору, повертаються до першого способу розміщення.

Щоразу, коли є блоки, які ще не розміщені, необхідно переходити до першого способу розміщення і так до тих пір, поки не будуть розміщені всі блоки. В кінці роботи алгоритму формується можливе рішення, показане на рис. 3.

## Оцінка складності алгоритму

В роботі проаналізовано складність алгоритму за найгірших умов, коли  $n$  - кількість усіх блоків, розміщених в контейнері,  $a_k$  - кількість різних типів

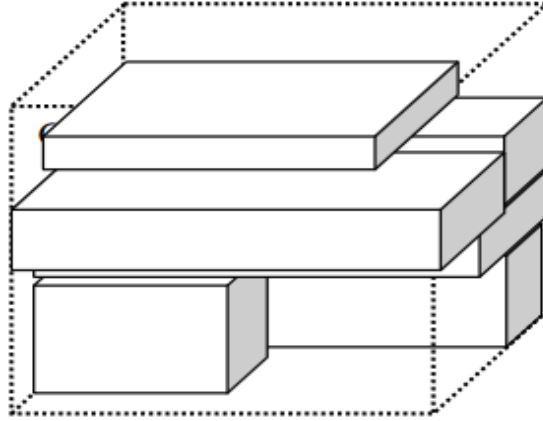


Рис 3. Можливе розміщення блоків у контейнері

блоків, тобто мають різні параметри. При цьому складність реалізації алгоритму дорівнює  $O(n^2)$ .

У таблиці 1 наведено результати тестування для різних  $k$  типів та кількості  $n$  блоків, що у сумі становлять  $V1$  об'єм, і розміщених у контейнері об'ємом  $V2$ .

Табл. 1. Результати тестів

Кількість різних типів блоків, $k$	Загальна кількість блоків, $n$	Об'єм усіх розміщених блоків, $V1$	Об'єм побудованого контейнера, $V2$	Відсоток не заповненого простору, %
1	10	8400	8400	0
1	20	12480	12480	0
2	5	2547	2610	2.41
2	10	6252	6480	3.52
2	15	25554	25920	1.41

2	20	49032	49320	0.58
5	5	3910	5100	23.3
5	10	11359	13680	16.97
5	20	22596	25840	12.55
10	10	13419	19200	30.11
10	20	21694	27740	21.80
10	30	12854	16800	23.49

Програма написана мовою програмування C та запускається на комп'ютері з такою конфігурацією: Intel Core Duo 1.73 GHZ CPU 1536 MB Ram.

### **Висновки**

Як видно з табл. 1, при збільшенні типів блоків збільшиться і об'єм невикористаного простору, оскільки помістити блоки різного розміру в контейнер важче, ніж блоки одного розміру. Запропонований алгоритм, як видно з табл.1, є ефективним алгоритмом, оскільки відсоток незаповненого простору знаходиться в прийнятних межах. В якості майбутньої роботи можна розглянути інші параметри блоків, такі як вага, її розподіл в контейнері, використання декількох контейнерів, порядок відвантаження блоків.

### **Література:**

1. *J. A. George and D.F. Robinson*, "A heuristic for packing boxes into a container." *Computer and Operations Research* - 1980 - 147-156.
2. *S. Martello, D., Pisinger, D., Vigo*, "Three dimensional bin packing problem", Technical Report DEIS-OR-97-6, Bologna University. *B. K. A. NGoi, M. L.* - 1997
3. *O., Fareo, D., Pisinger, M., Zachriassen*, Guided Local Search for the Three-dimensional Bin Packing Problem. Technical Report, Dept. of Computer Science, University of Copenhagen, Copenhagen, DK. - 1999 - 99-13

**К.т.н, доцент Орлова М. М., магістрант Коркішко А. О.**

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

## **АНАЛІЗ ТЕХНОЛОГІЇ БЛОКЧЕЙН, З ТОЧКИ ЗОРУ БЕЗПЕЧНОГО ПРОВЕДЕННЯ ЦИФРОВИХ ОПЕРАЦІЙ**

### **Abstract**

**Maria M. Orlova, assoc. prof., PhD; Anastasiia Korkishko, student**  
*Analysis of blockchain technology, in terms of secure digital operations*

*The aim of this work is to analyze the blockchain technology, in terms of ensuring the reliability of information transmission and formulating on this basis recommendations for the safe conduct of digital operations.*

### **Вступ**

Сучасне життя все більш пов'язане з інформаційними технологіями. Цифрова валюта (криптовалюта) все більше використовується в різних сферах діяльності людей. Більшість банківських систем підвищують свою безпеку із року в рік для того, щоб запобігти злочинним втручанням хакерів та виникненню зовнішніх або внутрішніх негативних факторів. Криптовалюта також розвивається стрімкими темпами, тому внесла свій вклад у розвиток безпеки, що сприяло створенню децентралізованої платіжної системи, яка не має будь-якого адміністратора та працює повністю в автоматичному режимі.

Саме тому технологію блокчейн все частіше розглядають як тип програмного забезпечення для покращення та безпеки реалізації транзакцій фінансових операцій. Використання цієї технології надає можливість покращити взаємодію між користувачами або цілими компаніями, радикально знижуючи ризики втрати криптовалюти або інформації.

### **Постановка задачі**

Метою даної роботи є аналіз технології блокчейн, з точки зору забезпечення надійності передачі інформації та формулювання на цій основі рекомендацій щодо безпечного проведення цифрових операцій.

## **Аналіз технології блокчейн**

Блокчейн – це база даних, яка працює за технологією розподіленої бази, яку часто називають книгою DLT (Distributed Ledger Technology), що робить історію будь-якого цифрового активу незмінною та прозорою за допомогою децентралізації та криптографічного хешування та управляється декількома учасниками [1].

Головною відмінністю блокчейну від звичайних баз даних - спосіб структуризації даних. Блокчейн збирає інформацію в групи (блоки), які зберігають певний набір інформації. Блоки мають певну ємність зберігання. При повному заповненні блок зв'язується з попереднім, цим самим створюючи ланцюг даних, точніше ланцюг блоків. Тобто, при цьому вся нова інформація, яка йде за тільки-но доданим блоком, компіюється в новостворений блок, який потім також буде доданий в ланцюг. База даних структурує свої дані в таблиці, тоді як блокчейн структурує свої дані у фрагменти (блоки), які пов'язані один з одним. Цим самим можна ствердити, що всі блокчейни є базами даних, проте не всі бази даних є блокчейнами.

Блокчейн складається з трьох важливих концепцій: блоків, вузлів та майнерів [2].

### **1. Блоки**

Кожен ланцюжок складається з декількох блоків, і кожен блок має три основні елементи:

- дані в блоці;
- 32-розрядне ціле число, яке є одноразовим, і генерується випадковим чином, коли створюється блок, який потім генерує хеш заголовку блоку;
- хеш - це 256-розрядне число, яке повинне починатися з великої кількості нулів (тобто бути надзвичайно малим).

Коли створюється перший блок ланцюга, одноразовий код генерує криптографічний хеш. Дані в блоці вважаються підписаними і назавжди прив'язані до одноразового числа та хешу.

### **2. Майнери**

Майнери створюють нові блоки в ланцюжку за допомогою процесу, який називається майнінгом.

У блокчейні кожен блок має свій унікальний одноразовий номер і хеш, але також посилається на хеш попереднього блоку в ланцюжку, тому майнінг блоку нелегкий, особливо у великих ланцюгах.

Майнери використовують спеціальне програмне забезпечення для вирішення неймовірно складної математичної проблеми пошуку Nonce (number only used once), який генерує прийнятий хеш. Оскільки одноразове значення має лише 32 біти, а хеш — 256, існує приблизно



чотири мільярди можливих комбінацій попсе-хеш, які необхідно видобути, перш ніж буде знайдено потрібну. Коли це відбувається, кажуть, що майнери знайшли "золотий попсе", і їх блок додається до ланцюга.

### 3. Вузли

Однією з найважливіших концепцій технології блокчейн є децентралізація. Жоден комп'ютер чи організація не можуть володіти ланцюгом. Натомість це розподілена книга через вузли, підключені до ланцюжка. Вузли можуть бути будь-якими електронними пристроями, які зберігають копії блокчейну та підтримують роботу мережі.

Кожен вузол має свою власну копію блокчейна, і мережа повинна алгоритмічно схвалити будь-який нововидобутий блок для оновлення, довіри та верифікації ланцюжка. Оскільки блокчейни прозорі, кожен дію в реєстрі можна легко перевірити та переглянути. Кожному учаснику надається унікальний буквено-цифровий ідентифікаційний номер, що відображає їх транзакції.

### **Переваги та недоліки технології**

1. Основною перевагою блокчейну так і залишається децентралізованість [3]. Централізовані системи мають перелік недоліків, які призводять до погіршення їх безпеки, в порівнянні з блокчейном:

- дані всі знаходяться в одному місці, що робить їх більш вразливими до атак хакерів;
- якщо централізований об'єкт відключиться від системи, то ніхто не зможе дістатися до даних, якими він володів;
- якщо централізований об'єкт – пошкоджений, то дані також можуть змінитися.

Всі ці проблеми блокчейн уникає, оскільки всі учасники мережі володіють однаковою інформацією, та комунікація між учасниками системи може відбуватися напряму, а не обов'язково лише через сервер.

### 2. Відкритість та конфіденційність

Імена користувачів системи захищені, проте учасники можуть побачити всі транзакції та час їх проведення. Інші фінансові системи не дають такої відкритості всіх операцій.

### 3. Незмінність

Технологія блокчейн вирішує питання безпеки та довіри кількома способами [1]. Нові блоки завжди зберігаються лінійно та хронологічно. Тобто вони завжди додаються до «кінця» блокчейна. Після того, як блок був доданий до кінця блокчейну, дуже важко (а іноді і неможливо) повернутися назад і змінити вміст блоку, якщо більшість не досягла консенсусу щодо цього. Це тому, що кожен блок містить свій власний хеш

разом з хешем блоку перед ним, а також згадану раніше позначку часу. Хеш-коди створюються математичною функцією, яка перетворює цифрову інформацію в рядок цифр і букв. Якщо цю інформацію будь-яким чином редагувати, хеш-код також змінюється, що є дуже важливим для безпечного зберігання даних. При спробі учасника змінити свою власну єдину копію блокчейну, вона більше не буде співпадати з копією всіх інших. При подальшому пересиланні копій між учасниками блокчейн помітить не співпадіння, та відкине змінену копію, як нелегітимну. Лише при одночасному контролюванні та зміненні 51% копій, нова копія зможе стати основною. Проте це вимагає великої кількості ресурсів, оскільки необхідно змінити всі блоки (вони матимуть різні часові позначки та хеш-коди).

### **Висновки**

Проведений аналіз технології блокчейн, як технології для безпечного проведення цифрових операцій, показав, що вона складається з трьох важливих концепцій: блоків, вузлів та майнерів. Основними перевагами блокчейну можна назвати: децентралізованість даної системи, абсолютну відкритість транзакцій, конфіденційність учасників та незмінність попередніх операцій. Саме тому можна стверджувати, що блокчейн має досить високий рівень безпеки проведення цифрових операцій.

### **Література**

1. Кріс Даннен. Представляємо Ethereum і Solidity. — Бруклін, Нью-Йорк, США: 2017. — 197 с.
2. Айєр К., Даннен К. Побудова ігор за допомогою смарт-контрактів Ethereum. Бруклін, Нью-Йорк, США: 2018. — 281 с.
3. Тромп, Джон. Фінансова криптографія та безпека даних: BITCOIN Бруклін, Нью-Йорк, США: 2015. — 52 с.

**К.т.н, доцент Орлова М. М., магістрант Сміун І. В.**

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

## **АНАЛІЗ ВИДІВ ТЕХНОЛОГІЇ BLOCKCHAIN ТА ЇХ ЗАСТОСУВАННЯ У СФЕРІ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ**

### **Abstract**

**Maria M. Orlova, assoc. prof., PhD; Inna Smiun, student**  
*Analysis of blockchain types and their application in the field of information security*

*This work is concerned with the study of information security threats and blockchain technology. Studied and analyzed Blockchain technology as a way to ensure information security. The introduction of Blockchain as a reliable and secure data protection system is proposed. Made the analysis of disadvantages and advantages of technology for information security.*

### **Вступ**

З стрімким розвитком технологій, сучасне життя стає все більш залежним від комп'ютерних технологій, а виведення з ладу комп'ютерної мережі або ж програмного забезпечення несе за собою безліч негативних наслідків. У сфері інформаційних технологій, проблема забезпечення інформаційної безпеки є однією з найголовніших. Кожного дня цінна інформація стає все вразливішою. У зв'язку з цим, на даному етапі розвитку технологій, усі сфери життя суспільства потребують реформ та впровадження нових сучасних та інноваційних методів для захисту інформації.

Існує багато способів забезпечення цілісності інформаційної безпеки, але більшість з них мають ряд недоліків. Технологія blockchain уже набула широкого використання у криптографії, вона забезпечує прозору та безпечну передачу інформації та запобігає маніпуляції даними. Інформаційна безпека стає все ближче до адаптації технології blockchain, такий метод дозволить забезпечити безпеку цифрової «особистості» користувача, безпечну передачу інформації [2].

## **Постановка задачі**

Метою даної роботи є аналітичний огляд технології blockchain, як метода захисту інформації, а також аналіз його видів та їх особливостей з точки зору інформаційної безпеки.

## **Термінологія**

Blockchain – це особливий тип захищеної розподіленої, децентралізованої бази даних, яка без централізованого сховища постійно розширяється та створює безперервний, незмінний ланцюг блоків, пов'язаних між собою.

Децентралізована база даних – база даних, що немає централізованого сховища. Всі дані розподілені між вузлами мережі. Якщо на будь-якому з комп'ютерів щось додається, редагується або видаляється, це стає відомим всім з'єднаним вузлам.

Інформаційна безпека – це способи боротьби з несанкціонованим доступом, використанням, розкриттям, спотворенням, змінам, дослідженням, записам або ж знищенням інформації.

## **Опис технології**

Blockchain є одним з видів розподіленого класу зберігання і синхронізації даних. Кожен вузол розподіленої системи робить записи в своїй версії реєстру, незалежно від інших вузлів і синхронізується з ними в рамках тимчасової мережі. Особливістю технології як виду розподіленого реєстру є те, що записи з'єднуються в інкрементальний ланцюжок блоків з використанням криптографічних алгоритмів. Після того, як дані були опубліковані у мережі, використовуючи надійну техніку відмітки часу і посилання на попередній блок, вважається, що вже неможливо видалити запис, чи додати у нього якісь зміни [1].

Крім самих записів (або транзакцій) та ідентифікатора блоку, у блок включаються хеш-суми поточного та попереднього блоків. Вони є результатом обчислення криптографічних хеш-функцій. Хеш-функції в blockchain, у поєднанні з його розподіленою архітектурою, забезпечують незмінність і незворотність всього ланцюжка блоків і транзакцій [1].

Крім хеш-функцій важливу роль в blockchain виконує набір математичних правил і функцій, званий алгоритмом консенсусу. Основним завданням алгоритму консенсусу є генерація і синхронізація ланцюга блоків у всіх учасників мережі. База даних blockchain зберігається у необмеженої кількості учасників блокчейн-мережі, які заздалегідь невідомі і можуть підключатися або відключатися в будь-який момент. Алгоритм консенсусу забезпечує досягнення спільної угоди про поточний стан бази даних усіма учасниками мережі. Робота по угрупованню нових

транзакцій в блоки і обчислення хеш-функцій для захисту ланцюга блоків виконується також вузлами консенсусу [1].

Поєднання властивостей розподіленого реєстру з блочною структурою даних, заснованою на криптографічному зв'язку, дозволяє blockchain ефективно реалізовувати ключові аспекти інформаційної безпеки – цілісність, конфіденційність, доступність інформації [3].

Існує декілька видів технології blockchain. Ланцюжки блоків мають різну доступність до реєстру даних - саме за цим параметром блокчейн поділяють на види. Деякі з них відкриті та загальнодоступні, а деякі приватні та доступні лише людям, яким надано дозвіл на їх використання. Основними видами вважають публічний, приватний та гібридний блокчейн. У залежності від видів, blockchain забезпечує різні аспекти інформаційної безпеки [2].

Публічний blockchain розподілений та децентралізований. Загалом можна описати публічний blockchain трьома характеристиками – незмінність, прозорість та децентралізованість. Однак традиційна модель децентралізованої публічної блокчейн-мережі, що забезпечує прозорість і стійкість до цензури, в силу своєї архітектури та ідеології не дозволяє забезпечити третій аспект інформаційної безпеки – конфіденційність даних.

Публічні блокчейн-мережі використовуються як інфраструктура для електронних платіжних систем. Досі продовжуються спроби адаптувати публічний блокчейн під корпоративні потреби, але для більшості державних та бізнес-замовників, публічний blockchain з його незалежністю та відкритістю даних, як і раніше, є невідповідною концепцією [].

Архітектура, криптографічні функції та алгоритми консенсусу децентралізованої блокчейн-мережі можуть бути потенційно пошкоджені, тим самим порушуючи базовий принцип blockchain – незмінність даних. Існує ряд можливих атак на інфраструктуру публічних блокчейн-мереж. Багато хто з них існують поки що в теорії та обговорюються як концептуальні.

Однією з таких атак є атака 51%, яка може відбутись, якщо одному об'єкту вдасться контролювати більше 50% потужності хешування мережі, що в підсумку дозволить порушити її роботу, шляхом навмисного виключення або зміни порядку транзакцій [2]. Не дивлячись на те, що це теоретично це можливо, атака 51% ніколи не була успішною.

Також публічний blockchain може постраждати від атаки Сівіли. Це вид атаки в однорангових мережах, коли зловмисник наповнює мережу підконтрольними йому вузлами, до яких підключаються вузли інших учасників [2].

Через ряд вище перерахованих можливих проблем публічного blockchain, а також через проблеми масштабованості, з'явилася модель приватного blockchain. Багато з описаних вище загроз також можуть бути актуальними і для приватних blockchain, але корпоративні сценарії мають власні особливості забезпечення інформаційної безпеки.

Приватний blockchain в першу чергу відрізняється моделлю забезпечення доступу до мережі, при якій право вносити зміни до реєстру є у строго визначених учасників. Крім того, у такому blockchain зазвичай обмежений доступ на читання записів реєстру. Така мережа має оператора, і вона вже не може бути децентралізованою, тільки розподіленою. Проте приватний blockchain дозволяє забезпечувати конфіденційність записів, так як тепер доступ надається згідно всіх правил забезпечення безпеки. Таким чином забезпечується третій аспект інформаційної безпеки – конфіденційність даних. Такі мережі набувають все більшого поширення як інфраструктура для корпоративних і державних завдань [4].

Blockchain у цій системі забезпечує захист даних від будь-яких несанкціонованих змін та унеможливорює дублювання інформації. Очевидно, що приватний blockchain забезпечує кращий контроль над інфраструктурою з боку організації чи групи компаній. Модель загроз для рішень, що працюють на приватному blockchain, вже не включає ряд атак, актуальних для публічного blockchain. Так, незважаючи на порівняно малий розмір розподіленої блокчейн-мережі, атака 51% або атака Сівілі стають важкими для виконання – всі вузли знаходяться під контролем у довірених зонах [3]. Гіпотетично зловмисник може отримати доступ і до підконтрольного вузла, але в такому разі вартість атаки багаторазово зростає через необхідність подолання корпоративного периметра різних організацій та засобів ІБ, що використовуються.

Існує модель гібридного blockchain, що поєднує обидва підходи. У такій моделі записи з приватної мережі або їх метадані можуть додатково зберігатися в публічному blockchain, забезпечуючи додаткову відмовостійкість всього реєстру. Гібридний блокчейн можна описати як blockchain, який намагається використати найкращі сторони як приватних, так і публічних blockchain. Гібридний blockchain пропонує такі функції як цілісність, прозорість та безпека. Члени гібридного blockchain приймають рішення щодо того, хто може брати участь у blockchain та які транзакції будуть публічними. В ідеалі гібридний blockchain надаватиме контрольований доступ і свободу одночасно.

Незважаючи на те, що транзакції в гібридному blockchain не є публічними, за необхідності вони можуть перевірятись. Кожна транзакція, яка відбувається в гібридному blockchain, може бути приватною і водночас відкритою для перевірки, якщо це необхідно. Але найголовнішу роль

роботі блокчейна грає незмінність: кожна транзакція записується один раз і може бути змінена, і саме такий вид може використовуватись практично у всіх сферах для захисту інформації.

## **Висновки**

У результаті проведеного аналізу існуючих видів blockchain, зроблено висновок про те, що всі вони мають ряд значних переваг для забезпечення інформаційної безпеки, серед яких можна виділити: цілісність, відстеження, стійкість, унікальність та незмінність інформації.

Досліджено, що найдоцільнішим видом для захисту інформації являється гібридний blockchain, у якому ланцюги блоків поєднують переваги обох видів технології, намагаючись при цьому обмежити недоліки, та завдяки чому забезпечують усі аспекти інформаційної безпеки. Гібридний blockchain працює в закритій мережі, перешкоджає витоку інформації, є захищеним від атаки 51% та атаки Сівіли. Крім того, незважаючи на те, що приватні блокчейни найкраще підходять для питань, пов'язаних із конфіденційністю, вони є обмеженими, коли справа доходить до взаємодії із зовнішнім світом, а гібридна модель може забезпечити такий доступ. Враховуючи вище наведено інформацію, гібридний blockchain є безпечним та надійним способом для захисту інформаційної безпеки.

## **Література**

1. Дон Тепскотт, Алекс Тепскотт. «Блокчейн-революція». Пер.з англ. Ю. Григоренко, Г. Лелів. Львів: Видавництво «Літопис», 2019. – С. 492
2. Киреенко, А. Е. Современные проблемы в области информационной безопасности: классические угрозы, методы и средства их предотвращения / А. Е. Киреенко. — [Електронний ресурс]. – Режим доступу: <https://moluch.ru/archive/38/4365/>.
3. Анин Б. Ю. Защита компьютерной информации / Б. Ю. Анин – Санкт-Петербург: БХВ-Санкт-Петербург, 2000. – 384 с.
4. Александр Табернакулов, Ян Койфманн. Блокчейн на практике. — М.: Альпина Паблишер, 2019. — 264 с.

УДК 004.054

К.т.н., доцент Павловський В.І., магістрант Бойко В.В.

Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”

## ОПИС ОСНОВНИХ СПОСОБІВ ДЕТЕКЦІЇ ПЛАГІАРИЗМУ

### Abstract

**Volodymyr I. Pavlovskyi, assoc. prof., PhD; Vladyslav Boiko, student**

*Description of the main techniques of plagiarism detection*

*This paper contains information about most common techniques and describe most difference between basic implemented programs. Problematic of such software is analyzed and this paper starts research in the sphere of the clone detection for the purpose of future improvement.*

### Вступ

Під плагіаризмом в програмному кодї розуміють часткове або повне дублювання фрагментів коду програми. Під терміном дублювання можна розглядати як ідентичні фрагменти коду в рамках однієї програми, так і плагіаризм у різних програмах.

Детекція плагіаризму завжди була і залишається актуальною задачею в розробці програмного забезпечення, як інструмент метричного аналізу і рефакторингу.

На сьогодні існує велика кількість програмних продуктів, які допомагають ідентифікувати плагіаризм у програмному кодї. Значна кількість інтегрованих середовищ розробки містить у собі відповідні розширення для полегшення розробки, підтримки і впровадження програмних продуктів, тому що накопичування великої кількості дублюючого програмного коду може призводити до виникнення значної кількості проблем, оскільки при детекції помилок в певному фрагменті коду, всі дублюючі фрагменти також повинні бути модифіковані, що може значно ускладнити впровадження і підтримку програмного продукту. Завдяки детекції плагіаризму дублюючий код може абстрагуватися і, як результат, загальний розмір програмного коду може бути зменшений. Крім того детекція плагіаризму в програмному кодї може використовуватися для



рефакторингу проєктів, зменшення кодової бази та пошуку порушення авторських прав.

### **Постановка задачі**

Провести аналіз основних технік – сукупність прийомів та способів, які використовуються в детекції плагіаризму в програмному коді та визначити напрями їх поліпшення. Оскільки детекція плагіаризму - це складний процес, котрий можна розбити на декілька етапів, то кожен з них може бути реалізований з використанням різних способів і технологій.

### **Аналіз технік, що використовуються для детекції плагіаризму**

Техніки детекції плагіаризму в програмному коді безпосередньо пов'язані з поняттям програмних клонів. На сьогодні розрізняють такі типи програмних клонів.

Тип 1 (Семантичні клони): семантично подібні фрагменти коду, хоча синтаксично вони можуть бути різні.

Тип 2 (Близькі клони): фрагменти коду, які мають велику частину ідентичного коду, але також містять додані/видалені твердження, різні ідентифікатори, літерали або типи.

Тип 3 (Параметризовані клони): частково ідентичні фрагменти коду, які можуть відрізнитися іменами змінних, типами або літералами.

Тип 4 (Точні клони): ідентичні фрагменти коду без урахування коментарів, пустих рядків і вайтспейсів.

Нижче проведено аналіз технік, які можуть ідентифікувати всі типи програмних клонів.

### **Техніки детекції плагіаризму**

Для якісного порівняння описаних нижче технік вводиться поняття *recall* як відношення кількості знайдених коректних клонів до загальної кількості клонів у вхідному програмному комплексі (1).

$recall = \frac{KK}{IK} * 100\%$  (1) , де *KK* — коректні клони, які дійсно являються плагіаризмом в програмному коді, *IK* — усі існуючі клони.

Також використовується поняття коректність (якість) знайдених результатів *precision*, представлено як відношення (2)

$precision = \frac{KK}{ЗК} * 100\%$  (2) , де *KK* — коректні клони, які дійсно являються плагіаризмом в програмному коді, *ЗК* — усі знайдені клони.

Чим вищі показники *recall* і *precision*, тим кращі результати показує певна техніка в задачах детекції плагіаризму. Але для повної оцінки технік такого типу зазвичай використовують *F-measure* (*F-mіра*), що є середнім гармонійним значень *precision* і *recall*.

### **Порівняння тексту програм**

Базовий варіант даної техніки реалізується шляхом посимвольного порівняння тексту двох програм. За допомогою даної техніки можна знайти клони тільки четвертого типу. Ця техніка досить проста в реалізації і може легко масштабуватися, не зважаючи на мову вхідної програми. Можливості даної техніки доволі обмежені, тому в базовому варіанті не використовується. Існує певна кількість модифікацій даної техніки.

Як приклад можна навести програмне забезпечення Dup [1]. Це модифікація техніки порівняння тексту. Модифікація полягає у нормалізації вхідного коду на етапі лексичного аналізу шляхом видалення коментарів і вайтспесів. Потім виконується хешування кожної лінії коду. Далі хеш-коди порівнюються і результати агрегуються. Існують модифікації даної техніки, які не використовують хешування, а безпосередньо порівнюють співставлення рядків.

Також можна відзначити техніку SDD [2], яка може визначати не тільки точні клони, а й частково ідентичні. Основою даної техніки є алгоритм Nearest Neighbor Distance, що значно розширює кількість потенційно опрацьованого коду.

Переваги даної техніки: універсальність та простота реалізації.

Проблеми даної техніки: обмежена кількість клонів, які можуть ідентифікуватись.

Оскільки дана техніка здатна знаходити тільки клони типу 4, то значення recall доволі низьке незалежно від реалізації, а значення precision має непогані результати. Міра F-measure варіюється в межах 6%-22% [1] залежно від реалізації.

### **Порівняння токенів вхідного коду**

На етапі лексичного аналізу виконується видалення коментарів і вайтспесів. Потім токени змінних замінюються на один універсальний токен, для того щоб ігнорувати різницю при порівнянні. Порівняння виконується вже шляхом матчингу груп токенів. Дана техніка суміжна з технікою порівняння тексту програм, але вона може бути модифікована і комбінована з іншими техніками для отримання кращих результатів. У базовій реалізації можливості даного підходу обмежуються детекцією клонів типу 4.

До представників програмного забезпечення такого типу можна віднести CCFinder. Воно базується на техніках описаних вище, але на додачу використовується комбінація з технікою, яка застосовує порівняння AST (Abstract Syntax Tree) програмного коду.

Також варто відзначити програмне забезпечення FRISC [3]. Воно також базується на техніці описаній вище, але після виконання лексичного

аналізу групи токенів формуються відносно символів '{' та '}' для виділення конкретних блоків програмного коду. На основі сформованих груп виконується хешування і створення структурного хеш-коду, що дозволяє розширити типи клонів, які можуть ідентифікуватися: тип 2, тип 3, тип 4.

Переваги даної техніки: простота реалізації.

Проблеми даної техніки: відсутність універсальності, обмежена кількість клонів, які можуть ідентифікуватися.

Оскільки даний підхід дозволяє визначати декілька типів клонів, то і значення recall і precision доволі непогані, але дуже залежать від конкретної реалізації. Тому F-measure даної техніки варіюється від 17% до 66% [5].

### **Порівняння абстрактних синтаксичних дерев вхідного коду**

Дана техніка є найбільш популярною і має широкі можливості для використання. За допомогою даного підходу можуть бути ідентифіковані клони типу 2 - 4. Основою даної техніки є побудова абстрактних синтаксичних дерев і їх порівняння з використанням алгоритмів матчингу.

Найвідомішим на даний момент є програмне забезпечення Koschke [4]. Його значною перевагою є лінійна складність виконання, але якщо порівнювати дану техніку з техніками порівняння тексту програм і порівняння токенів вхідного коду, то складність реалізації на порядок вища.

Переваги даної техніки: велика варіативність реалізації, можливість інтеграції з іншим програмним забезпеченням, висока швидкодія.

Проблеми даної техніки: складність реалізації, обмежена кількість клонів, які можуть бути ідентифіковані.

У базовому варіанті дана техніка має досить погані результати, оскільки обмежена тільки аналізом синтаксичних конструкцій, тому зазвичай програмне забезпечення даного типу використовується в комбінації з іншим програмним забезпеченням, яке реалізує іншу техніку. Значення F-measure даного програмного забезпечення варіюється від 8% до 32% [4].

### **Аналіз семантики вхідного коду**

Можна виділити дві під-техніки цієї техніки: аналіз графів залежності і гібридний аналіз.

### **Аналіз графів залежності**

Аналіз графів залежності виконується на основі ізоморфних підграфів PDG (Program Dependency Graph) [5].

Переваги даної техніки: можливість інтеграції з іншим програмним забезпеченням.

Проблеми даної техніки: складність реалізації, низька швидкодія та не універсальність.

Програмне забезпечення вказане вище має значення F-measure в рамках від 95% до 98% [1], що є близьким до ідеального результату, але результати роботи обмежуються тільки детекцією клонів типу 3.

### **Гібридний аналіз вхідного коду**

Гібридний аналіз виконується на основі комбінації декількох технік.

Прикладом даного програмного забезпечення є Funago [6]. Дане програмне забезпечення поєднує в собі аналіз абстрактних синтаксичних дерев і порівняння тексту вхідних програм. Поєднання даних технік дозволяє зменшити кількість співставлень на абстрактних деревах, що в свою чергу зменшує складність реалізації програмного забезпечення, але негативно впливає на швидкодію.

Переваги даної техніки: велика варіативність реалізації, можливість інтеграції з іншим програмним забезпеченням, можливість ідентифікації всіх типів клонів.

Проблеми даної техніки залежать від конкретної реалізації.

Оцінка ефективності даного підходу безпосередньо залежить від конкретної реалізації.

### **Висновки**

У даній статті проведена класифікація та опис основних технік і технологій, детекції плагіаризму в програмному коді. Висвітлені основні проблеми і недоліки існуючих підходів з метою їх подальшого аналізу і покращення. Наразі можна стверджувати, що не існує задовільної техніки, якщо говорити про точність, можливість масштабування, надійність і швидкодію, яка б повністю задовольнила практичні потреби розробників програмного забезпечення. Тому актуальною залишається задача розробки нових підходів до детекції плагіаризму в програмному коді, які б дозволили уникнути обмежень існуючих. Одним з таких підходів є подальша комбінація існуючих технік детекції плагіаризму, шляхом застосування конкретної техніки для детекції клонів певного типу. Оскільки існуючі представники даного програмного забезпечення мають ряд недоліків, зокрема відсутність універсальності, низька швидкодія, складність реалізації, то це дає підстави для розробки нового, ефективнішого програмного забезпечення даного типу.

## Література

1. Koschke, R., Falke, R. and Frenzel, P. (2006), Clone detection using abstract syntax suffix trees. Reverse Engineering, WCRE'06. 13th Working Conference on. IEEE, October 23-27, 2006 Benevento, Italy, pp. 3-8.
2. Wahler, V., Seipel, D. (2004), "Clone Detection in Source Code by Frequent Itemset Techniques", *SCAM*, Vol. 4, pp. 128-35
3. Funaro, Marco, (2010), A hybrid approach (syntactic and textual) to clone detection. Proceedings of the 4th International Workshop on Software Clones. ACM, 2010.

**К.т.н., доцент Павловський В.І., студент Рекеда В.В.**

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

## **ВЕБ-СЕРВІС ПІДВИЩЕННЯ РОЗДІЛЬНОЇ ЗДАТНОСТІ ЗОБРАЖЕНЬ З ВИКОРИСТАННЯМ SR- АЛГОРИТМІВ**

### **Abstract**

**Volodymyr I. Pavlovskiy, assoc. prof., PhD; Volodymyr Rekada, student**  
*Web service for improving image resolution using SR-algorithms*

*This paper describes the usage of Super Resolution (SR) algorithms for improving image resolution. The paper presents one of modifications of SR-algorithms which called "Super Resolution from a Single Image".*

### **Вступ**

Завдання покращення роздільної здатності зображень є важливим для різних практичних застосувань. Серед таких застосувань можна виділити обробку та аналіз медичних зображень, супутникових фотознімків, обробку даних з пристроїв відеоспостереження і т.д. Ці застосування вимагають високої якості зображень, тому за останні два десятиліття було запропоновано багато методів для підвищення роздільної здатності зображень.

На сьогоднішній день існує два основних підходи, які застосовуються при вирішенні цієї задачі. Перший спосіб – апаратний. Для цього необхідно використовувати фотоапарати з високоякісною матрицею. Мінусом даного підходу є необхідність використання дорогого спеціалізованого обладнання. Інший спосіб – використання програмних засобів, які реалізують різні алгоритми підвищення якості зображень. Такі алгоритми отримали назву SR-алгоритми, які дозволяють якісно збільшити роздільну здатність вихідного зображення, що виходить за межі фізичної роздільної здатності цифрового сенсора, який записав зображення. Перевагою даного підходу є зниження вимог до апаратної складової.

## Постановка задачі

Суть задачі полягає у створенні веб-сервісу підвищення роздільної здатності зображення з відкритим API, який для обробки зображення не потребує додаткових вхідних даних.

## Термінологія

*SR (Super-Resolution) алгоритми* – сімейство алгоритмів, які призначені для покращення роздільної здатності зображення з одного або кількох вхідних зображень з низькою роздільною здатністю. SR-алгоритми можна поділити на два сімейства методів: класичні SR-алгоритми та SR-алгоритми, які базуються на послідовності навчальних прикладів.

*YCbCr* – колірна модель, яка використовується у цифрових зображеннях та відео, де *Y (luminance)* – компонента, яка відповідає за яскравість, *Cb (chrominance-blue)* – компонента синього кольору, *Cr (chrominance-red)* – компонента червоного кольору.

*RGB* – адитивна колірна модель, в якій використовуються червоне, зелене та синє світло для синтезу певного кольору.

## Опис веб-сервісу та аналоги

При реалізації програмного засобу (ПЗ) підвищення роздільної здатності зображення було обрано формат веб-сервісу. Головною перевагою такого формату є наявність API, завдяки якому можливо використовувати функціонал сервісу в інших ПЗ, наприклад, веб-додатках, класичних додатках чи у чат-ботах. Серед недоліків такого формату ПЗ є неможливість його використання без підключення до мережі Інтернет.

Нині існує кілька програмних засобів, які дозволяють підвищити роздільну здатність зображення. В таблиці 1 наведено порівняння кількох ПЗ, які дозволяють підвищити роздільну здатність зображення.

Таблиця 1

Порівняння програмних засобів покращення роздільної здатності зображень

Назва	Тип	Платформа	Тип ліцензії	Наявність API
Adobe Photoshop	Класичний додаток	Windows, macOS	Умовно безкоштовне ПЗ	Ні
letsenhance.io	Веб-додаток	Будь-яка	Умовно безкоштовне ПЗ	Так
SmartDeblur	Класичний додаток	Windows	Умовно безкоштовне ПЗ	Ні

Основним недоліком цих сервісів є їх умовна безкоштовність. Такий тип ліцензії накладає певні обмеження на використання ПЗ, серед яких

пробний період, обмежена кількість зображень, які можливо опрацювати, а також нанесення водяних знаків на оброблені зображення. Крім того, не всі ці сервіси мають власні API, що унеможлиблює їх використання в інших програмних засобах. З цього можна зробити висновок, що у вільному доступі практично відсутні ПЗ підвищення роздільної здатності зображення, які б не мали ліцензійних обмежень та надавали власні API для їх використання в якості веб-сервісу.

Таким чином задача створення веб-сервісу підвищення роздільної здатності зображень з відкритим API є актуальною.

Основним модулем сервісу виступає модуль обробки зображення. Така обробка реалізується за допомогою сімейства SR-алгоритмів.

### **Вибір та опис модифікації SR-алгоритму**

Існує два основних сімейства SR-алгоритмів: класичні SR-алгоритми та SR-алгоритми, які базуються на послідовності навчальних прикладів.

В класичних алгоритмах використовується кілька вхідних зображень з низькою роздільною здатністю. Головна ідея – знайти у вхідних зображеннях повторювану інформацію, а потім об'єднати її для формування зображення з високою роздільною здатністю. Класичні SR-алгоритми розділяють на 4 класи: алгоритми з використанням регуляризації, алгоритми, що базуються на спектральному поданні, інтерполяційні алгоритми, баєсівські алгоритми [1].

При використанні SR-алгоритмів, які базуються на послідовності навчальних прикладів, існує база даних, в якій попарно зберігаються зображення з високою та низькою роздільною здатністю. Ідея полягає в розбитті вхідного зображення на окремі фрагменти та пошук схожих фрагментів серед наявних в базі даних [2].

В даній роботі буде використано модифікацію алгоритму, яка отримала назву Super Resolution from a Single Image. Ця модифікація поєднує в собі переваги класичних SR-алгоритмів та SR-алгоритмів, які базуються на послідовності навчальних прикладів. Зображення з низькою роздільною здатністю конвертується в зображення з високою роздільною здатністю без будь-якої іншої додаткової вхідної інформації.

В означеній модифікації SR-алгоритму використовується така властивість зображення, що невеликі фрагменти вихідного зображення мають тенденцію до надлишкового повторення в рамках цього ж зображення, до того ж таке повторення зберігається і в отриманому зображенні з підвищеною роздільною здатністю у різних масштабах [3].

Послідовність кроків алгоритму:

1. Виконати перетворення вхідного зображення з моделі RGB в YCbCr.



2. Виконати поділ зображення, яке було отримано на першому кроці, на три кадра, де кожен новий кадр відповідає одному з каналів YCbCr-моделі.

3. Для кадрів, що відповідають каналам Cb і Cr, виконати операцію збільшення роздільної здатності до цільового значення, використовуючи метод бікубічної інтерполяції.

4. Для кадру, який відповідає каналу Y, побудувати піраміду зображень.

4.1. Для вхідного кадру низької роздільної здатності  $L = l_0$ , побудувати послідовність кадрів нижчої роздільної здатності  $l_{-1}, \dots, l_{-n}$ , де  $n$  – рівень піраміди. При побудові кадрів нижчої роздільної здатності використовується оператор розмиття Гауса. Масштаб кожного наступного кадру в піраміді розраховується як добуток масштабу попереднього кадру на коефіцієнт  $\alpha$ , де  $\alpha = 2^{1/3}$ [4].

4.2. Для кадру  $L = l_0$ , побудувати послідовність порожніх кадрів вищого дозволу  $l_1, \dots, l_n$ , де  $l_n = H$ , використовуючи аналогічне масштабування.

5. Відновити зображення на вершині піраміди.

5.1. Для кожного пікселя вихідного зображення та області, яка його оточує (розміром  $5 \times 5$  пікселів), виконати пошук схожих ділянок в усіх зменшених копіях вихідного зображення. Для пошуку використовується алгоритм Approximate Nearest Neighbor search.

5.2. При знаходженні схожої ділянки на зображенні меншого масштабу рівня  $-l$  зберігається її образ з початкового зображення і копіюється у відповідне місце в зображенні рівня  $l^2$ .

5.3. Для кожного пікселя верхнього зображення піраміди та області, яка його оточує (розміром  $10 \times 10$  пікселів), виконати пошук ділянки, яка передувала відновленій на тій же позиції, але на зображеннях меншого масштабу. Для знайденої області, використовуючи алгоритм загаданий в кроці 5.1, знайти  $k$  схожих областей на тому ж рівні піраміди. Области, які були знайдені на цьому кроці формують базу зображень для використання класичного SR-алгоритму, в якому для підвищення роздільної здатності потрібно кілька зображень. Використовуючи класичний SR-алгоритм та знайдені області, відновити вихідну ділянку на цільовому зображенні.

6. Перевірити, чи досягнуте цільове значення роздільної здатності. Якщо ні, то виконати крок 4 по відношенню до кадру на вершині піраміди. Якщо так, то перейти до наступного кроку.

7. Вирівняти роздільну здатність отриманого кадру для отримання цільової роздільної здатності.
8. Виконати об'єднання кадрів, які відповідають каналам YCbCr.
9. Виконати перетворення отриманого зображення з моделі YCbCr в модель RGB.

## Висновки

У роботі розглянуто програмний засіб підвищення роздільної здатності зображення. Цей засіб створено у вигляді веб-сервісу з відкритим API, що дозволяє користуватись ним і в інших додатках. Основним компонентом сервісу є модуль обробки зображення, в якому використовується SR-алгоритм.

Також в роботі проведено та обґрунтовано вибір модифікації SR-алгоритму для використання в створюваному веб-сервісі. Це модифікація Super Resolution from a Single Image, яка базується на використанні класичних SR-алгоритмів. Її перевагою цієї модифікації в порівнянні зі звичайними класичними SR-алгоритмами є те, що для отримання нового зображення з підвищеною роздільною здатністю достатньо лише одного вхідного зображення.

## Література

1. *M. Irani and S. Peleg.* Proving resolution by image registration. CVGIP, (3), 1991.
2. *W. T. Freeman, T. R. Jones, and E. C. Pasztor.* Example-based super-resolution. Comp. Graph. Appl., (2), 2002.
3. *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.
4. *C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi.* В Photo-realistic single image super-resolution using a generative adversarial network. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), November 2017.

**К.т.н, доцент Павловський В.І., магістрант Савосько О.М.**

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

## **ВИЯВЛЕННЯ ШКІДЛИВОГО ТРАФІКУ ЗА ВИКОРИСТАННЯМ ГЛИБИННОГО НАВЧАННЯ**

### **Abstract**

**Volodymyr I. Pavlovskiy, assoc. prof., PhD; Oleksandra Savosko, student**  
*Malicious traffic detection*

*This work proposes a data processing technique that divides data streams into data stream segments so that the model can improve throughput per time unit to achieve detection efficiency. For this kind of data, a malicious traffic detection model with a hierarchical attention mechanism is also proposed and named HAGRU (Hierarchical Attention Gated Recurrent Unit). Introduced an attention mechanism to focus on harmful streams in a data stream segment that can intelligently use limited computing resources and compare the proposed model with the current state of the method on the datasets.*

### **Вступ**

Безперервний розвиток комп'ютерних мереж суттєво змінює спосіб життя та роботи людей. Враховуючи поступове посилення поточної ситуації з безпекою мереж, трафік зловмисних атак переполює все мережеве середовище, а поточна модель виявлення шкідливого трафіку недостатньо ефективна.

Характер мережевого трафіку є багатовимірним і зашумленим, що не дозволяє отримувати від статичних даних трафіку корисну й якісну інформацію про особливості трафіку.

У даній роботі пропонується метод обробки даних потоку, який поділяє ці дані на сегменти, щоб модель виявлення шкідливого трафіку могла покращити пропускну здатність за одиницю часу й забезпечити ефективність виявлення мережевих втручань. Виконано порівняльний аналіз запропонованої моделі з поточним станом методу на основі використання рекурентних нейронних мереж. Експериментальні результати показують, що запропонована модель добре працює за різними показниками оцінки і може покращити ефективність розпізнавання категорій втручань з меншою кількістю вибірок, коли дані незбалансовані.

## **Постановка задачі**

Сучасні системи виявлення комп'ютерних атак не забезпечують належний рівень захисту комп'ютерних систем, а їх методи мають ряд недоліків. Найточніший на сьогодні метод, що ґрунтується на сигнатурному аналізі, добре функціонує при виявленні вже відомих комп'ютерних атак, але не придатний для виявлення нових типів втручань.

В роботі розглядається та досліджується впровадження глибокої нейронної мережі з ієрархічним механізмом уваги, як метод виявлення шкідливого трафіку. Основною задачею є визначення основних переваг та недоліків запропонованої моделі виявлення шкідливого трафіку, на основі аналізу існуючих рішень.

## **Термінологія**

Глибинне навчання (deep learning machine) - галузь машинного навчання, що ґрунтується на наборі алгоритмів, які намагаються моделювати високорівневі абстракції в даних, застосовуючи глибинний граф із декількома обробними шарами, що побудовані з кількох лінійних або нелінійних перетворень.

Рекурентна нейронна мережа - клас штучних нейронних мереж, у якому з'єднання між вузлами утворюють граф орієнтований у часі.

Багатошаровий перцептрон - нейронна мережа з прямим зв'язком, яка відображає набір вхідних векторів у вихідні.

## **Опис технології**

В даний час існує багато програмних продуктів, спрямованих на вирішення такого актуального завдання, як виявлення шкідливого трафіку [1].

Завдання виявлення вторгнень в обчислювальні мережі зазвичай вирішуються із застосуванням: експертних систем; штучних нейронних мереж; нечітких систем; генетичних алгоритмів. Найбільш часто в системах виявлення вторгнень застосовуються експертні системи. Даний факт можна пояснити тим, що сигнатурний метод аналізу мережевого трафіку є найбільш швидким і не вимагає великих обчислювальних потужностей. Найбільшим недоліком експертної системи як системи виявлення вторгнень є її нездатність в принципі виявляти нові види атак. Крім того, відомо безліч технологій обходу систем виявлення вторгнень на основі експертних систем.

Перспективним напрямом виявлення аномального трафіку є використання глибоких нейронних мереж.

Глибокі нейронні мережі, завдяки багатшаровій архітектурі дозволяють обробляти і аналізувати великий обсяг даних, а також моделювати когнітивні процеси в різних областях.

Перший шар мережі може отримати низькорівневі ознаки, другий шар – ознаки більш високого рівня і т.д.

Представлена в даній роботі модель заснована на ефективній, надійній глибокій рекурентній нейронній мережі.

Зображена на рис. 1 ієрархічна модель уваги, розділена на п'ять частин. А саме, вхідний шар, частина перетворення ознак, частина блоку пам'яті із двонаправленим стробуванням, частина ієрархії та частина виходу багаторівневого перцептрону.

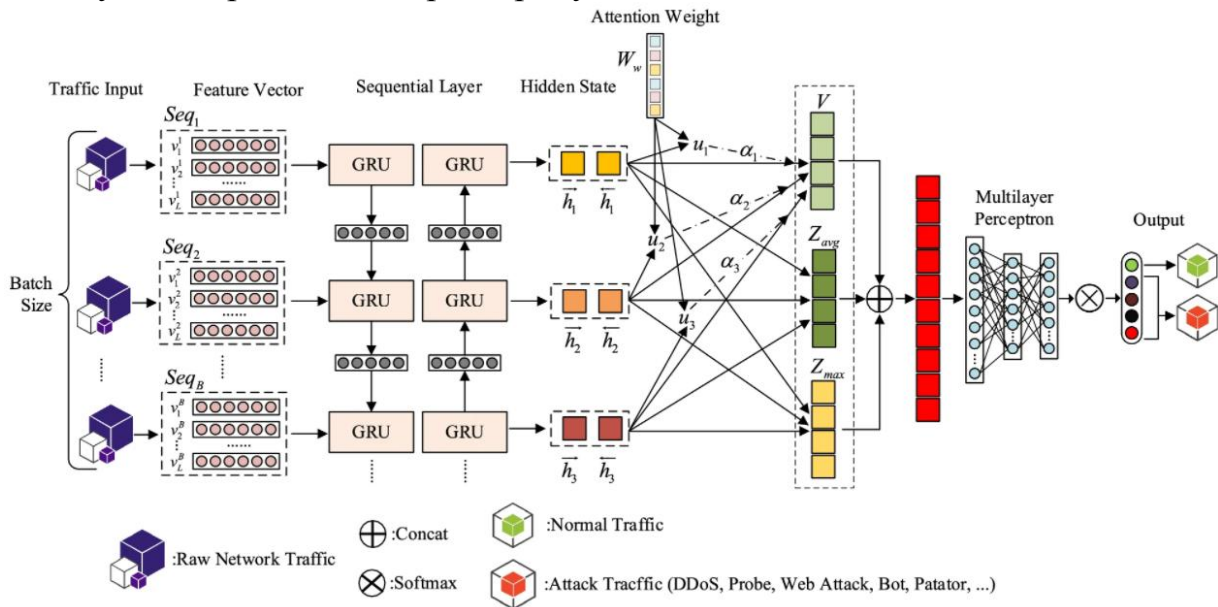


Рис. 1. Конструкція глибокої рекурентної нейронної мережі

Попередня обробка даних, які вводяться в модель, показана на рис. 2.

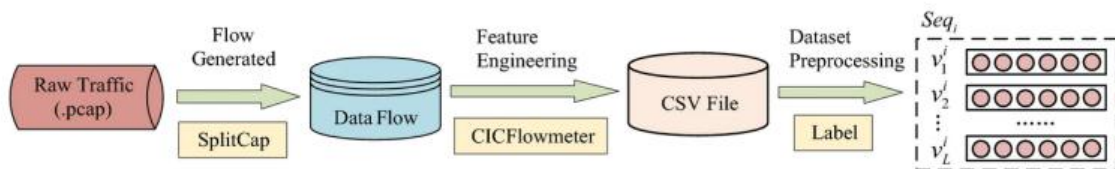


Рис. 2. Обробка даних трафіку

Відповідно до сегмента потоку даних, застосовуються операції трьох рівнів, а саме ієрархія механізму уваги, ієрархія максимального об'єднання та ієрархія середнього об'єднання. Результати цих операцій поєднуються, щоб отримати більш багаті функції трафіку, що полегшує моделі можливість ідентифікувати шкідливий трафік.

Рівень механізму уваги дозволяє моделі підтримувати свою обчислювальну здатність постійною, щоб підвищити продуктивність

моделі та краще ідентифікувати шкідливий і звичайний трафік. В даній моделі окрім використання механізму уваги для виділення важливих функцій, також використовуються функції максимального об'єднання та середнє об'єднання. Багатошаровий перцептрон лінійно перетворює особливості ієрархічного злиття і, нарешті, виводить категорію трафіку.

### **Переваги та недоліки технології глибоких нейромереж в виявленні вторгнень в комп'ютерні мережі**

До основних переваг систем виявлення вторгнень на основі нейромереж слід віднести:

Перша - це гнучкість, яку надає ця мережа. Особливо важлива здатність мережі обробляти дані з декількох джерел в нелінійній формі.

Суттєвою перевагою цього підходу є швидкість нейронних мереж.

Найбільш важливою перевагою цієї мережі є її здатність "навчатися" ознакам атак і визначати випадки, які не характерні для атак, що спостерігалися раніше.

Недоліки систем виявлення вторгнень на основі нейромереж:

Перший: вимоги до навчання нейронної мережі. Адже здатність штучної нейронної мережі до ідентифікації ознак вторгнення повністю залежить від правильного навчання системи.

Ще одним недоліком є природа "чорної скриньки" нейронних мереж. Вони адаптують свій аналіз даних у відповідь на отримане навчання, що відрізняє їх від експертних систем, які мають жорстко закодовані правила для аналізу подій.

### **Дослідження моделі**

Для підтвердження доцільності застосування запропонованої моделі використані три різних набори даних NSL-KDD, CIC-IDS2017 і CES-CIC-IDS2018. В експерименті обробляється частотна вибірка даних, дисбаланс даних і навчальний та тестовий набори, на які поділяється кожен набір даних у співвідношення 8:2. Запропонована модель оцінюється за частотою виявлення та хибнопозитивною частотою. Для порівняння виявлення шкідливого трафіку обрано шість сучасних методів. В якості показників оцінки обрано DR, FPR і  $F$ -score. Кількість показників оцінки знаходиться в межах [0-1], чим вище DR та  $F$ -score, тим краще модель працює з такими даними. Чим нижче значення FPR, тим кращі результати. Також було обрано оптимальну довжину сегмента потоку. Вплив довжини сегмента потоку на модель HAGRU зображений на рис. 3. Після розгляду комплексних індексів «Точність», «Швидкість виявлення», «FPR» та « $F$ -score», коли довжина сегмента потоку становить 384, запропонована

модель HGRU працює найкраще. Тому в моделі розмір сегменту потоку даних становить 384.

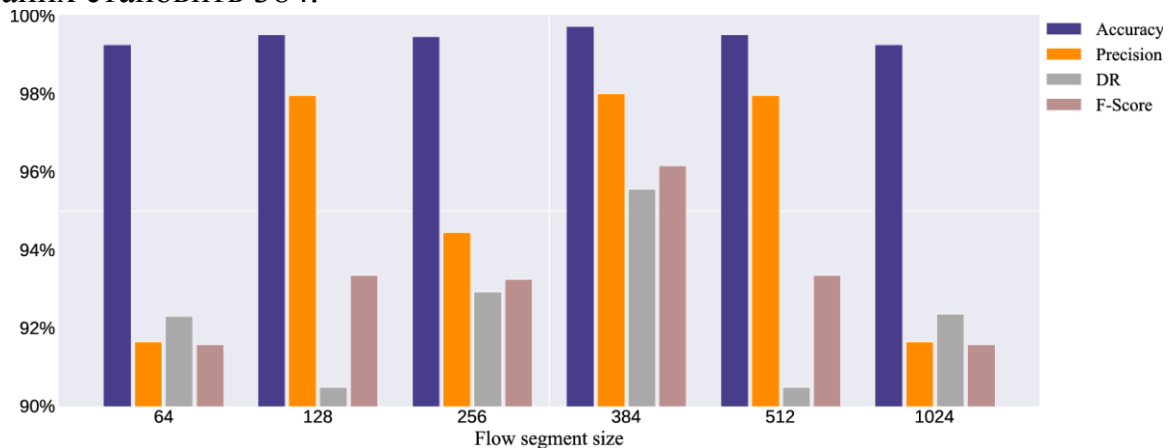


Рис. 3. Порівняльний аналіз результатів роботи моделі

## Висновки

Якщо розглядати одну категорію даних, запропонована модель може добре визначити категорії, порівняно з традиційними моделями, вона може розглядати більше типів мережевих атак. Основна причина полягає в тому, що HAGRU використовує механізм уваги та ієрархії, його три рівні можуть отримати велике різноманіття характеристик потоку навіть із невеликої вибірки даних, він може добре ідентифікувати трафік у разі дисбалансу даних.

## Література

1. Multi-Layer Integrated Anomaly Intrusion Detection System for Mobile Adhoc Networks [Електронний ресурс].2007. Режим доступу до ресурсу: [http://xanadu.cs.sjsu.edu/~drtylin/classes/cs157A/Project/PDFfiles/CS157B\\_Team14/4\\_Eric\\_Nam%20-%20IEEE%20Website/4156645.pdf](http://xanadu.cs.sjsu.edu/~drtylin/classes/cs157A/Project/PDFfiles/CS157B_Team14/4_Eric_Nam%20-%20IEEE%20Website/4156645.pdf).
2. IDES: An Intelligent System for Detecting Intruders [Електронний ресурс]. – 1990. – Режим доступу до ресурсу: [https://www.researchgate.net/profile/Teresa\\_Lunt/publication/242383334\\_Ides\\_an\\_intelligent\\_system\\_for\\_detecting\\_intruders/links/552daca0cf29b22c9c4f95f/Ides-anintelligent-system-for-detecting-intruders.pdf](https://www.researchgate.net/profile/Teresa_Lunt/publication/242383334_Ides_an_intelligent_system_for_detecting_intruders/links/552daca0cf29b22c9c4f95f/Ides-anintelligent-system-for-detecting-intruders.pdf)
3. Detecting Intruders in Computer Systems [Електронний ресурс]. – 1993. – Режим доступу до ресурсу: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.43.7289>

**К.т.н., доцент Петрашенко А. В. студент Коровій О.С.**

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

## **СПОСІБ РОЗПІЗНАВАННЯ ТОНАЛЬНОГО ЗАБАРВЛЕННЯ ТЕКСТІВ НА ОСНОВІ ПРОЦЕСУ "ДИСТИЛЯЦІЇ ЗНАНЬ"**

### **Abstract**

**Andrii V. Petrashenko, assoc. prof., PhD; Korovii Oleksandr, student**  
*Method of sentiment analysis of texts based on "distillation knowledge" process*

*This paper describes a method application of "distillation knowledge" for sentiment analysis. For our research we used two types of different architecture of neural networks for natural language processing: BERT and FastText. Combination of these two neural networks (BERT as a teacher and FastText as a learner) allowed us to achieve a maximum performance with high accuracy for BERT – 0.822, for FastText – 0.79 in sentiment analysis task by using the method of "distillation knowledge". It is showed how to minimize resources not losing a lot of accuracy, but speeding up the text sentiment recognition.*

### **Вступ**

Зі зростанням складності задачі обробки природньої мови та складністю нейронної мережі, збільшуються вимоги до потужностей комп'ютера. Для навчання глибоких нейронних мереж (кількість нейронів перевищує 100 мільйонів) потрібно використовувати хмарні технології, а саме, оренду цілого кластера з високопотужними GPU.

Використання глибоких нейронних мереж відкриває для нас можливість вирішення всіх потрібних задач в області обробки природньої мови з високою точністю та повнотою, але це потребує великі ресурси, які потрібно використовувати як при навчанні нейронної мережі так і при подальшому її використанні [1].

### **Постановка задачі**

Задача полягає у використанні процесу "дистиляції знань" для збільшення швидкодії при вирішенні задачі тонального забарвлення тексту при обробці природньої мови, але зі зменшенням точності.



## Термінологія

*Нейронна мережа* – це математична модель, яка емітує природній процес навчання нейронів мозку.

*BERT* – це глибока нейронна мережа архітектури Bidirectional Encoder Representations from Transformers.

*FastText* – архітектура з мінімальним розміром нейронної мережі при вирішенні задач класифікації

*State-of-the-art* – останній етап розробки продукту, що включає новітні технології, ідеї та функції.

## Опис процесу «дистиляції знань»

Для вирішення даної проблеми із ресурсами можна застосувати **дистиляцію знань**. Даний підхід дозволяє натренувати неглибоку нейронну мережу (кількість параметрів < 1 мільйон), використовуючи знання, які були вивчені глибокою нейронною мережею (кількість параметрів більше 100 мільйонів).

Спочатку навчається глибока нейронна мережа, на вхідних даних, і досягає максимально можливої точності, далі за допомогою цієї глибокої нейронної мережі (вчитель), генеруємо синтетичні дані, для навчання неглибокої нейронної мережі (учень). Це потрібно для того щоб уникнути ефекту перенавчання у неглибоких нейронних мереж, при недостатній кількості вхідних даних для навчання, неглибока нейронна мережа може завчити їх, та показувати 100% точність, але коли на вхід цієї мережі подати дані, які не були включені до навчальної вибірки, то нейронна мережа може дуже сильно помилятися. «Учень» навчається на синтетичних даних які згенерував «вчитель» і на даних на яких навчався сам вчитель, щоб охопити якомога більший об'єм даних для навчання. При цьому втрачається декілька пунктів точності. А іноді втрати точності немає, залежить від вхідних даних. Але в будь-якому разі при використанні даної нейронної мережі використання ресурсів зменшується в декілька десятків разів [2].

Робота складається з навчання глибокої нейронної мережі архітектури Bidirectional Encoder Representations from Transformers (BERT) [3], яка має близько 300 млн параметрів для навчання, і не глибокої нейронної мережі архітектури FastText [4], яка має близько 1 млн параметрів. Задача на якій будемо використовувати дистиляцію знання – це тональне забарвлення тексту. В даній роботі під текстами вважатимуться – новини. Використання пари BERT + FastText дозволило досягнути високого результату при використанні методу дистиляції тексту. BERT було обрано, так як це State-of-the-art мовна модель, для вирішення

задач природньої обробки мови. FastText було обрано, так як вона має архітектуру мінімального розміру.

### Результати

Як видно з рисунка 1, влучність (precision) найбільша для нейтрального класу, при тому повнота (recall) найкраще виражена у позитивного класу. Якщо розглядати значення f1-функції (це середнє значення між влучністю та повнотою), то можна сказати що глибока нейронна мережа архітектури BERT найкраще визначає позитивні тексти, потім негативні і проблем завдають їй нейтральні новин, вона там найбільше помиляється.

	precision	recall	f1-score
negative	0.81	0.85	0.83
neutral	0.83	0.74	0.78
positive	0.82	0.92	0.87
accuracy			0.82
macro avg	0.82	0.84	0.83
weighted avg	0.82	0.82	0.82

Рис. 1. Значення по різних метрикам відносно кожного класу для моделі BERT

Виходячи з результатів, що на зображенні на рисунку 2, точність моделі FastText на синтетичних даних складає 0.79081. Влучність найкраща для негативного класу, а от повнота – для нейтрального класу, f1-значення для нейтрального класу є найбільшим і складає 0.86, а от для позитивного класу – 0.53, що є доволі поганим результатом, це свідчить про те, що нейронна мережа буде найчастіше помилятися із визначенням позитивних новин.

	precision	recall	f1-score
negative	0.81	0.61	0.70
neutral	0.79	0.93	0.86
positive	0.73	0.42	0.53
accuracy			0.79
macro avg	0.78	0.65	0.69
weighted avg	0.79	0.79	0.78

Рис. 2. Значення по різних метрикам відносно кожного класу для моделі Fasttext

### Висновки

В роботі було проведено дослідження використання методу «дистиляції знань» для вирішення задачі тонального забарвлення тексту. Було використано дві архітектури нейронних мереж, модель BERT у ролі

вчителя та модель Fasttext – у ролі учня. По результатам модель вчителя показала точність 0.822, а модель учня – 0.79, різниця складає 0.022 пункти, що є хорошим результатом. Модель BERT має кращу влучність та повноту, для позитивних та негативних класів. Тобто сплутування позитивного тексту із негативним – мінімальне. Модель Fasttext в цьому аспекті робить більше помилок.

Модель BERT показує кращий результат за рахунок «розуміння тексту на рівні контексту», а не тільки на наявність слів. Модель Fasttext орієнтується лише на слова.

Якщо порівняти продуктивність, цих двох моделей то модель Fasttext є в 5 раз швидша, ніж модель BERT. Використовуючи лише ресурси CPU, модель Fasttext показала швидкодію близько 1000 текстів за секунду. Модель BERT, використовуючи ресурси GPU, показала швидкодію близько 200 текстів за секунду.

Тобто використання методу «дистиляції знань» дозволяє підвищити швидкодію визначення тонального забарвлення тексту, але при цьому втрачається точність.

## Література

1. *Goodfellow, I., Bengio, Y. and Courville, A.*, 2016. Deep learning. Cambridge (EE. UU.): MIT Press.
2. *Hinton, G., Vinyals, O. and Dean, J.*, 2021. Distilling the Knowledge in a Neural Network. [online] arXiv.org. URL: <https://arxiv.org/abs/1503.02531>
3. *Devlin, J., Chang, M., Lee, K. and Toutanova, K.*, 2021. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. [online] arXiv.org. URL: <https://arxiv.org/abs/1810.04805>
4. *Joulin, A., Grave, E., Bojanowski, P. and Mikolov, T.*, 2021. Bag of Tricks for Efficient Text Classification. [online] arXiv.org. URL: <https://arxiv.org/abs/1607.01759>

**К.т.н, доцент Петрашенко А.В., магістрант Рябоконт Т.О.**

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

## **ПРОГРАМНІ ЗАСОБИ ФОРМУВАННЯ ТА ОБРОБКИ БАЗИ ДАНИХ СЛОВОСПОЛУЧЕНЬ УКРАЇНСЬКОЇ МОВИ**

### **Abstract**

**Andrii V. Petrashenko, assoc. prof., PhD; Tetiana Riabokon, student**  
*The software system for generation and processing of a database of collocations of  
the Ukrainian language*

*This paper is devoted to the description of creating a database of collocations and identifying the most effective methods of collocation extraction from the text. Existing researches of statistical methods of collocation extraction from text data were analyzed. The architecture of automated generation of the database of collocations and possible ways of its acceleration is also described*

### **Вступ**

Аналіз словосполучень є важливим розділом NLP досліджень. Вміння аналізувати, класифікувати та знаходити словосполучення дає можливість оперувати контекстом та змістом, що закладені у речення, а не окремими словами. Це допомагає значно вдосконалити системи, що працюють з натуральними мовами. Існує багато алгоритмів та підходів, які дозволяють аналізувати окремі слова, але аналіз та пошук групи слів, що зв'язані між собою є більш складним завданням.

Словосполучення важливі для ряду застосувань: генерація природної мови – щоб переконатися, що вихідні дані звучать природно і уникнути помилок; обчислювальна лексикографія – для автоматичного визначення важливих словосполучень, які мають потрапити до словника та корпусні лінгвістичні дослідження, наприклад, вивчення суспільних та культурних явищ через мову.

### **Постановка задачі**

Дане дослідження проводиться з практичною метою – за результатами аналізу текстів та знаходження в них всіх словосполучень, які вдасться виділити за допомогою обраного алгоритму – побудувати словник

словосполучень української мови. Створена база словосполучень дозволить шукати слова, які найчастіше вживаються разом, також можна переглянути різні тип словосполучень та речення-приклади для того, щоб зрозуміти контекст, в якому найчастіше вживається шукане слово. Такий словник може бути корисним, як для використання людьми, наприклад, у навчанні, так і для потреб NLP, адже кількість україномовних ресурсів, які можна використати для досліджень у даній сфері дуже обмежена.

### **Опис розроблюваної системи**

Подібні системи вже існують для інших мов, вони також вміють працювати зі словосполученнями, але особливість створеної системи в тому, що вона автоматично оновлюється використовуючи декілька джерел, кількість яких буде зростати з часом. Тобто даний словник не обмежений в кількості джерел, він весь час поповнюється новими текстами різних жанрів та авторів.

Розроблювана база даних словосполучень наповнюється за допомогою автоматичного оброблення текстових даних статистичними методами пошуку словосполучень. Для пошуку словосполучень з тексту можна використовувати різні методи навчання асоціативних правил. Наприклад, застосовують метод точкової взаємної інформації, t-критерій Стюдента, індекс Соренсена, логарифмічна правдоподібність для обчислення ступеня близькості між складовими словосполучення у текстовому корпусі.

Відомості про t-критерій можна знайти у багатьох підручниках загальної статистики. Зокрема найвідоміша праця це «Statistical methods» Дж. Снедекора та В. Кокрена [1].

Серед багатьох загальних методів, представлених в роботі К. Маннинга та Г. Шютце 1999, найкращі результати можна досягти шляхом виділення словосполучення на основі як мовної, так і статистичної моделі [2]. А Т. Данінг (1993) вказав на слабкість методу взаємної інформації і показав, що логарифмічна правдоподібність є більш ефективним методом при виявленні одномовних словосполучень, особливо коли їх кількість дуже мала [3].

Через те що система повинна оброблювати велику кількість даних, а алгоритми нормалізації слів та виділення словосполучень працюють недостатньо швидко, була використана система Apache Spark. Вона дозволяє швидше обробляти набори великих даних, розділяючи роботу на частини та розподіляючи ці фрагменти між обчислювальними ресурсами. [4] Таким чином можна обробляти велику кількість текстів за порівняно невелику кількість часу та постійно оновлювати словник.

Виділення словосполучень з тексту є ключовою частиною системи, для неї використаний один з алгоритмів пошуку, що найкраще підходить для роботи з текстами українською мовою.

В даному дослідженні для визначення найефективнішої міри асоціації був використаний підхід, описаний Евертом [5], суть якого полягає в виділенні невеликої випадкової вибірки позитивних і негативних прикладів. Аналіз був проведений на текстовому корпусі UA-GEC. Результати для біграм наведені на Рис 1. були отримані після застосування POS-фільтра та фільтра за списком стоп-слів. На рисунку видно, що всі перевірені показники працюють краще, ніж простий підрахунок частоти, що виправдовує використання мір асоціації. Також видно, що метод точкової взаємної інформації (PMI) працює найкраще, за ним слідують хі-квадрат та логістична правдоподібність, а найгірше себе показав t-критерій Стьюдента.

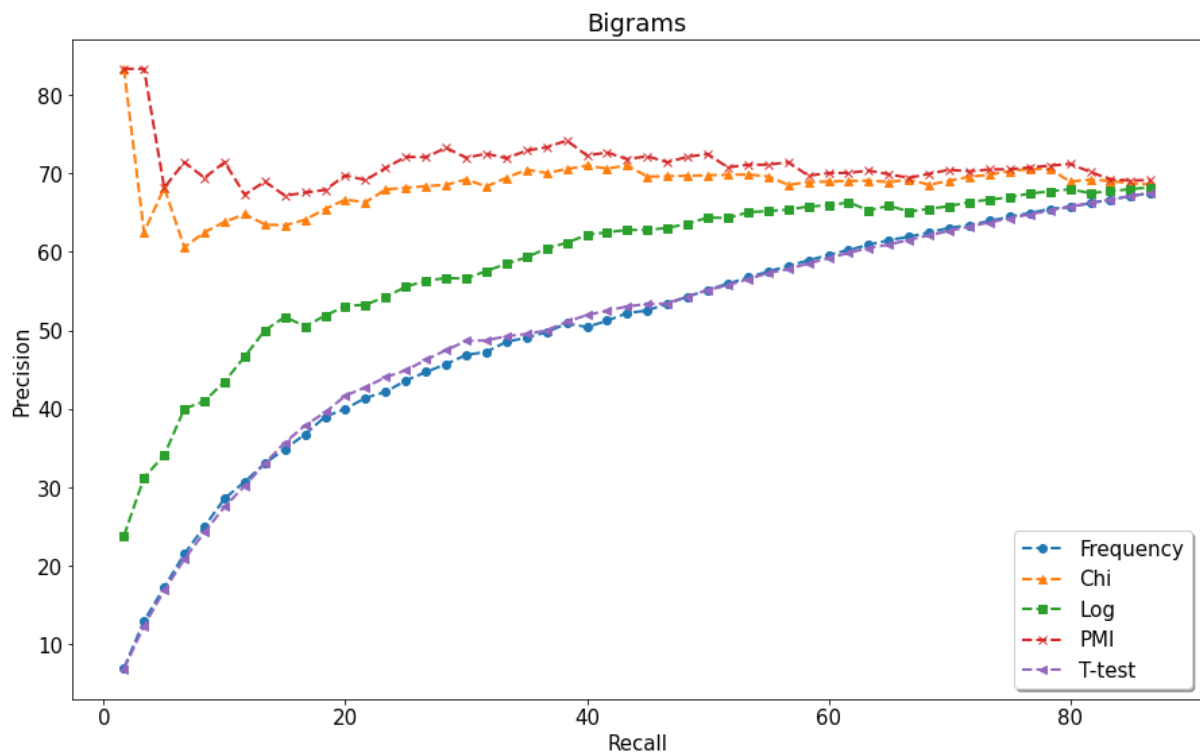


Рис.1. Результати обчислення влучності та повноти для біграм

Ще одною важливою частиною даної системи є пошук. Описана база словосполучень буде корисною тільки якщо можна буде зручно отримати із неї слова, що найчастіше вживаються із шуканим словом та приклади вживання даного слова. Ця частина системи реалізована за допомогою Elasticsearch, тому що він дозволяє швидко зберігати великі обсяги даних та шукати інформацію в них. Він дозволяє легко писати складні запити для пошуку за будь-якими критеріями, шукати під час введення тексту, а

також збирати статистику за великими обсягами даних, і що не менш важливо він добре масштабується. [6]

Розроблювана система має такий алгоритм роботи:

1. завантаження текстів в систему,
2. попередня обробка текстових даних (нормалізація, фільтр за стоп-словами та ін.),
3. застосування методу або декількох методів знаходження колокацій,
4. фільтрація колокації,
5. індексування даних та завантаження до пошукової системи.

При чому виконання пунктів 2–4 саме прискорені за допомогою Spark, а в якості системи пошуку використаний Elasticsearch.

Отже, розроблювана система вирізняється серед інших подібних автоматизованим наповненням словника та прискореним аналізом текстів, на основі яких саме будується база словосполучень.

## **Висновки**

Даний аналіз був проведений для побудови статистичної моделі для подальшого використання разом із іншими засобами обробки природної мови з метою подальшого використання для побудови бази словосполучень української мови. Розроблювана система постійно оновлюється, джерела для пошуку словосполучень будуть додаватися, а знайдені словосполучення вже автоматично потраплятимуть до словника.

Основною перевагою такого словника є відносна автономність, тобто його не потрібно наповнювати людям, наповнення відбуватиметься автоматично. Дана система також має пошук найчастіше вживаних слів разом зі словом, що ввів у систему користувач. Даний словник може бути використаний як допоміжний матеріал для вивчення української мови та як ресурс для досліджень у сфері NLP.

Майбутні дослідження будуть зосереджені на пошуку словосполучень, що складаються більше ніж з трьох слів, або є досить незалежними, щоб бути знайденими за допомогою простих статистичних методів.

## **Література**

1. Snedecor, George Waddel, and William G. Cochran. 1989. Statistical methods. Ames: Iowa State University Press. 8th edition. 53 с.
2. Manning C. and Schütze H., 1999. Foundations of Statistical Natural Language Processing. Cambridge: MIT Press.

3. Dunning T., 1993. Accurate methods for the statistics of surprise and coincidence. Computational Linguistics.
4. Apache Spark - Unified Analytics Engine for Big Data. URL: <https://spark.apache.org/>
5. S. Evert, B. Krenn, Using small random samples for the manual evaluation of statistical evaluation measures. Computer speech and language, 19: pp. 450–466; 2005.
6. Free and Open Search: The Creators of Elasticsearch, ELK & Kibana | Elastic. URL: <https://www.elastic.co/>



**К.т.н., доцент Петрашенко А.В., магістрант Бідяк М.А.**

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

## **ПРОЦЕС ВИЯВЛЕННЯ ТА ЗАПОБІГАННЯ ІНФОРМАЦІЙНИМ АТАКАМ ЗА ДОПОМОГОЮ МАШИННОГО НАВЧАННЯ**

### **Abstract**

*Andriy V. Petrashenko, assoc. prof., PhD; Mykhailo Bidiak, student*

*The process of detecting and preventing information attacks using machine learning*

*This paper concerns the task of detecting and preventing information attacks, especially DDoS attacks in the cloud. Distributed Denial of Service (DDoS) attacks have been the major threats to the Internet and can bring great loss to companies and governments. Therefore, the problem of protection against such attacks is acute. Thus, in this work, we consider the process of protection and counteraction to attacks using machine learning.*

### **Вступ**

Комп'ютерні мережі в сучасну епоху стали основою багатьох установ, включаючи урядові, дослідницькі та оборонні організації. З розвитком інтернету, хмарних обчислень, інтернету речей, сучасних мереж, постало завдання захисту таких ресурсів. Основні проблеми безпеки класифікуються як загрози цілісності, доступності та конфіденційності. Інформація, яка зберігається й обробляється, має бути конфіденційна та цілісна, а ресурси, які використовуються, мають бути доступні.

DDoS-атака є основною загрозою доступності ресурсів, оскільки вона намагається запобігти нормальному трафіку між клієнтом і сервером. DDoS-атаки стали звичною практикою в суто конкурентній боротьбі. Піддаватися атакам на замовлення можуть офіційні сайти компаній конкурентів. Іноді хакери займаються прямим шантажем приватних компаній, вимагаючи гроші за те, щоб не атакувати їх сайти [1].

Процес виявлення інформаційної атаки можна виконувати вручну, перевіряючи параметри мережевих пакетів, але при цьому страждає швидкість та точність виявлення. Також, існують автоматизовані алгоритми, з наперед визначеною поведінкою мережі під час атаки, які використовуються у промислових масштабах, але вони не захищатимуть

від досі не виявлених, нових атак. Тому гостро постає задача точно та швидкого виявлення інформаційних атак.

### **Постановка задачі**

Задача полягає в побудові та визначенні процесу виявлення та запобігання інформаційним атакам (в основному DDoS-атакам) в хмарних середовищах таким чином, щоб розроблена система була надійною та точною, але при цьому мінімально вплинула на швидкість ресурсу, на якому буде використовуватись.

### **Термінологія**

*Машинне навчання* – це підгалузь штучного інтелекту в галузі інформатики, яка часто застосовує статистичні прийоми для надання комп'ютерам здатності поступово покращувати продуктивність у певній задачі, без того, щоби бути програмованими явно.

*DDoS-атака* – це напад на комп'ютерну систему з наміром зробити комп'ютерні ресурси недоступними користувачам, для яких комп'ютерна система була призначена. DDoS-атаки можуть являти собою величезні обсяги трафіку за короткий час, низькі обсяги трафіку протягом тривалого часу, величезні обсяги трафіку за тривалий час [2].

*Класифікатор* – це алгоритм або техніка, для машинного навчання, що використовується для групування або класифікації даних у різні категорії залежно від характеристик. Використовується для вивчення шаблонів наявних даних, а потім класифікації даних у різні групи відповідно до шаблонів.

### **Обґрунтування вибору інструментів**

Існують багато різних методів виявлення інформаційних атак. Ці методи класифікуються на три основні категорії: виявлення на основі підписів (SD), виявлення на основі аномалій (AD) та аналіз протоколів стану (SPA). Методи машинного навчання часто використовуються для виявлення аномалій [3]. Даний метод використовується завдяки своїй здатності виявляти нові, до цього не існуючі, атаки. Виявлення базується на визначенні поведінки мережі. Наперед задається поведінка мережі. Якщо поведінка не відповідає наперед визначеній – приймається рішення ініціації події у виявленні аномалії.

При виявленні DDoS-атак, найскладнішим є виявлення відхилень від нормального трафіку. Тому, варто використовувати метод виявлення на основі аномалій. За допомогою штучного інтелекту, можна зробити диференціацію нормальних та атакуючих пакетів, такий підхід мінімально

вплине на швидкість роботи всієї системи. Пакети, які класифікуються як атакуючі, не потрібні, вони мають відкидатись та не обробляться.

Для коректного визначення відхилень, визначимо параметри, які буде аналізувати штучний інтелект: кількість пакетів, середній розмір пакета, дисперсія часового інтервалу, дисперсія розміру пакета, кількість байтів, швидкість передачі пакетів та бітрейт.

Алгоритми машинного навчання [4-5] бувають чотирьох типів: контрольоване навчання, неконтрольоване навчання, напівконтрольоване навчання та навчання з посиленням. Алгоритми контрольованого навчання – навчаються за деякими навчальними даними, а оцінка здійснюється за допомогою тестових даних. Для виявлення інформаційних атак найчастіше використовується контрольоване навчання.

Для класифікації пакетів, потрібно використовувати спеціальний алгоритм-класифікатор. Таким чином, визначається, які пакети приходять, нормальні чи атакуючі, за заданими параметрами. Існують такі алгоритми класифікації для машинного навчання під наглядом [4-5]: Artificial neural networks (ANN), Gaussian Naïve Bays (GNB), K nearest neighbor (KNN), Random Forest (RF). Вибір алгоритму залежить від задачі, застосування та характеру наявного набору даних. Алгоритм класифікації потрібно підбирати під конкретну задачу, адже система має бути надійною та швидкою, тому потрібно враховувати такі параметри як, точність та час виявлення. Провівши дослідження, виявлено, що для схожих задач виявлення, використовують ANN або RF алгоритми, вони показують найкращі результати по швидкості та точності. Пропонуємо використовувати модель ANN для виявлення інформаційних атак.

### **Опис алгоритму виявлення та запобігання атаки**

Задача виявлення DDoS-атак у трафіку досить складна, адже складно відрізнити атакуючий трафік від нормально. На виявлення впливають багато параметрів. Для оптимізації, пропонується використати алгоритм Зозулі, за допомогою якого виконаємо оптимізацію ознак, які потім передамо в модель ANN. Це призведе до підвищення точності класифікації.

Таким чином, вибравши інструменти та визначивши за якими параметрами мережевого пакету буде відбуватись класифікація, можна побудувати систему захисту. Загальний алгоритм виявлення та запобігання інформаційним атакам може мати такий вигляд:

1. Прийняття мережевого пакету на обробку.
2. Отримання параметрів пакету для класифікації.
3. Нормалізація значень характеристик трафіку за допомогою алгоритму Зозулі.

4. Обробка даних алгоритмом машинного навчання класифікації.
5. У залежності від результату класифікації – пропуск мережевого пакету чи його обробка.

## **Висновки**

Інформаційні атаки – серйозна загроза безпеці. Вони спричиняють зупинку послуг онлайн-додатків, мережевих послуг та ресурсів. Під час DDoS-атаки користувачі чекають на послуги поки система залишається зайнятою вирішенням хибних запитів зловмисників. Важливо мати захист від такого роду атак, адже це можуть бути урядові, дослідницькі або оборонні ресурси, які мають працювати без перебоїв.

У цій роботі автори дослідили останні досягнення щодо атак на мережу, було визначено процес виявлення та запобігання інформаційним атакам за допомогою алгоритмів штучного інтелекту таким чином, щоб вплив на швидкість системи був мінімальний, а надійність ресурсів збільшилась. Машинне навчання – точний та практичний метод для ідентифікації DDoS-трафіку. Також, визначено параметри для класифікації мережевих пакетів.

У майбутньому, дану систему можна вдосконалити, розробивши модель з глибоким навчанням, яка буде навчатись у процесі класифікації.

## **Література**

1. *Дубов Д. В.* Забезпечення національних інтересів України в глобальному та національному кіберпросторах. Кіберпростір як новий вимір геополітичного суперництва./ Д. В. Дубов // Київ: Національний інститут стратегічних досліджень. – 2014. – 116 с.
2. *Hsieh C. J.* Detection DDoS attacks based on neural-network using Apache Spark / C. J. Hsieh, T. Y. Chan // International Conference on Applied System Innovation (ICASI). – 2016.
3. *Lane B.* Using machine learning for advanced anomaly detection and classification / B. Lane, M. Poole, M. Camp, J. Murray-Krezan // Advanced Maui Optical and Space Surveillance Technologies Conference. – 2016.
4. *Ayon D.* Machine learning algorithms: A review / D. Ayon // International Journal of Computer Science and Information Technologies. – 2016. – Vol. 7, №3. – P. 1174–1179.
5. *Bzdok D.* Statistics versus Machine Learning / D. Bzdok, N. Altman, M. Krzywinski // Nature Methods. – 2018 – №15 (4). – P. 233–234.

УДК 004.623

К.т.н., доцент Петрашенко А.П., студент Дідус А.В.

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

## АНАЛІЗ МЕТОДІВ СТВОРЕННЯ НАБОРУ ТЕКСТОВИХ ДАНИХ ІЗ ВІДКРИТИХ ДЖЕРЕЛ ДЛЯ ПОДАЛЬШОЇ ОБРОБКИ

### Abstract

**Andrii V. Petrashenko, assoc. prof., PhD; Andrii Didus, student**  
*Analysis of methods for creating text datasets from open sources for computing.*

*This paper concerns the analysis of methods for collecting text data from the Internet. We researched a few methods of collecting data. In the end, we created and use parsers for getting text data from open sources like news sites, messengers.*

### Вступ

Кожного дня кількість даних зростає в геометричній прогресії, особливо текстових, адже текст – це найпоширеніша форма передачі інформації людьми. Тож коли попередні десятиліття характеризувались генерацією даних в основному, то сьогодні з розвитком потужностей дозволяє нам почати їх обробляти та шукати нові способи їх структуризації, збору та організації.

Нові системи збору даних з всесвітньої мережі Інтернет, сучасні способи їх масштабування та бази даних, які розроблені одразу під зберігання текстових даних та алгоритми їх упорядкування, структуризації дозволяють підійти до аналізу текстових даних повністю по-новому заради отримання максимальної вигоди від подальшого їх використання.

Аналіз нових способів збору даних дозволяє швидко та коректно збирати дані з потрібних сайтів. А правильно збережений і зібраний – це один із ключів успішного підходу до обробки та аналізу текстів, зокрема генерації текстів та використанню даного набору в машинному навчанні для розпізнавання подробиць повідомлень у новинних повідомленнях.

### Постановка задачі

Задача полягає в аналізі та порівнянні існуючих підходів до збору даних із відкритих джерел, зберігання їх та структуризації для подальшого аналізу.

## Термінологія

*Кодувальник* – людина, яка кодує набір даних, створює його або розмічає.

*Парсер* – програма-робот для обходу веб-сторінок заради збору даних.

*Парсинг* – збір даних за допомогою зчитування їх із сайтів парсерами.

*Навчальна вибірка* – множина значень, що використовується для тренування моделі.

*Тестова вибірка* – множина значень, що використовується для тестування моделі.

*Пайплайн* – послідовність відокремлених стадій в програмі.

### Основні методи збору даних

Для створення текстового набору даних використовують багато підходів задля отримання найвищої якості вибірки, які важко розділити, адже дуже часто вони можуть поєднуватись та доповнювати один одного.

Так, можна виділити такі методи збору даних:

1. Ручний збір даних – це збір даних, який виконується людьми-волонтерами за певну плату або так званими кодувальниками. Такого виду вибірки можуть формуватись в основному у невеликих об'ємах (зазвичай до десяти тисяч) або компаніями з великим бюджетом, адже потребує великої кількості людських ресурсів, які є доволі цінними.
2. Автоматизований збір даних – це процес збору даних з використанням програмованих засобів, який є доволі підходящим заради створення великих наборів даних.

### Автоматизований збір даних

Оскільки, кінцевою метою нашого набору даних є аналіз тексту та застосування до нього методів машинного навчання, отже для формування його був використаний саме автоматизований збір даних [1]. В свою чергу, можна використати цей метод кількома способами:

1. За допомогою вже готових програм, які пишуть потрібну інформацію у файл. До таких програм належать parserhub, Octoparse. Такого виду програми використовуються зазвичай людьми без досвіду в створенні програмного забезпечення, оскільки не вимагають особливих навичок в програмуванні.
2. З використанням бібліотек з відкритим кодом, за допомогою яких створюються парсери. До таких бібліотек належать:

2. 1. Beautiful Soup – одна із найпопулярніших бібліотек для парсингу мовою Python. Основна ідея якої – це створення так званого «дерева» із html або xml файлу.
2. 2. Selenium – бібліотека, яка дозволяє користуватись сайтами так, ніби це робить людина через один із доступних вам браузерів – Chrome, Firefox, Edge.
2. 3. Scrapy – одна із кращих бібліотек, яка дозволяє створити "павук" для обходу посилань в декілька рядків коду, окрім цього, дозволяє створювати цілі пайплайни для збору та первинної обробки даних.

### Зберігання текстових даних

За останні роки з появою цілого розділу в інженерії програмного забезпечення як Big Data, було розроблено багато баз даних оптимізованих для зберігання даних у великих кількостях, зокрема: ArangoDB, Solr, ElasticSearch [3].

Проте текстові дані – це доволі специфічні дані, оскільки текст – це та структура, яку дуже важко зберігати розділено, адже це потребуватиме додаткових ресурсів при подальшому доступу до цих даних.

Після пошуку можливих варіантів та їх аналізу нами було обрана ElasticSearch – база даних, яка дуже гарно підходить для таких цілей, адже дозволяє робити маніпуляції над текстом прямо після інсталяції та дуже гарно оптимізує сам процес збереження і дозволяє отримати потрібний нам результат набагато швидше в порівнянні з іншими базами даних. ArangoDB програє у швидкості, Solr не дозволяє отримати хороший результат без детального налаштування, так як це можна зробити в Elasticsearch. ElasticSearch підтримує власну мову запитів та дозволяє отримувати дані за допомогою вбудованого API, що не потребує додаткових налаштувань в інфраструктурі.

### Результати

Так, для збору даних на потрібних нам сайтах було розроблено парсер та розгорнуто його в контейнеризаційній системі Docker разом із ElasticSearch, що дозволило ізолювати середовища збору даних та їх зберігання.

```
docker ps -a --no-trunc --filter name=telegram
```

	IMAGE	COMMAND	CREATED
793d3e2c4cfd629ba3580bb461f5388e387378aa1711fe89	adidus/parser:telegram	"python start.py"	37 hours
3ee4b382539418ab1383f6d7f2bc64c118bda13e6a6db74	adidus/parser:telegram	"python start.py"	37 hours
2a7cc1f94316f75b81a38c655fa086c5e0e4a1cf2eb7794a	adidus/parser:telegram	"python start.py"	37 hours
36391721b71292554f7a4432872d965010e86e1f8da7f09a	adidus/parser:telegram	"python start.py"	37 hours
ba1067f861b4c112426f8eb0ab5035881db87554a1135ceb	adidus/parser:telegram	"python start.py"	37 hours
320847ef8050aa15deb2cd74eb1fcc2bcd5e13c438b4c35	adidus/parser:telegram	"python start.py"	37 hours
9998fd344096fdca280d4bea6805690d5718a0fab64e86a	adidus/parser:telegram	"python start.py"	37 hours
3c5cb2d4eece06829da292d4197ff40bb94eeda5803f727f	adidus/parser:telegram	"python start.py"	37 hours
032c3ac8d3ea1206ec1503641163660d84e90dd91f3daf7	adidus/parser:telegram	"python start.py"	37 hours
35c87eddb27f4f8f120b8a81ac91ee552d2c74c034a54090	adidus/parser:telegram	"python start.py"	37 hours
008744dce172b83519c0b2478e3f5eeaeec74e3c5daea32	adidus/parser:telegram	"python start.py"	37 hours
34dd62ee5ae2526b07d571d85ae41b4e0caea6febee05cc6	adidus/parser:telegram	"python start.py"	37 hours

Рис. 1. Зображення з мережею контейнерів працюючих парсерів

Таким чином, було отримано оптимальну структуру з набором даних для подальшої обробки і використання її задля автоматичної генерації нових текстів.

## **Висновки**

При порівнянні кількох методів підходу до збору даних та їх зберігання було виявлено оптимальні для нас – це парсинг за допомогою власноруч запрограмованих парсерів з використанням бібліотек мовою Python та зберігання їх в базі даних ElasticSearch, яка ідеально підійде для наших цілей, а саме – обробка та генерація тексту. Така комбінація дозволяє гнучко конфігурувати збір даних для нашого набору даних та використовувати дану конфігурацію в майбутньому. Окрім цього, дане поєднання використовує сучасні рішення, що дозволяє імплементувати такого виду конфігурації в сучасні проекти [2].

## **Література**

1. *Aisyah Awalina, Jibrán Fawaid, Rifky Yunus Krisnabay, Novanto Yudistira*, 2021. Indonesia's Fake News Detection using Transformer Network. [online] arXiv.org. Available at: <https://arxiv.org/pdf/2107.06796.pdf>
2. *Devlin, J., Chang, M., Lee, K. and Toutanova, K.*, 2021. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. [online] arXiv.org. Available at: <https://arxiv.org/abs/1810.04805>
3. *Free and Open Search: The Creators of Elasticsearch, ELK & Kibana | Elastic*. URL: <https://www.elastic.co/>



УДК 004.05

К.т.н., доцент Потапова К.Р., магістрантка Празднікова М. О.

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

## МОДИФІКАЦІЯ АЛГОРИТМУ КАННІ ДЛЯ ОБЧИСЛЕННЯ МЕЖ ОБ'ЄКТІВ ЗОБРАЖЕНЬ

### Abstract

*Potapova Katherina, assoc. prof., PhD; Prazdnikova Marharyta, student  
Method of recognition of object boundaries  
by a modified Canny algorithm using derivatives*

*The article formulates the problem of pre-processing of pictures of the scene to identify the contours of the image. A modified algorithm is developed, which first upgrades the image and adds a black and white hue, which allows you to better highlight the contour.*

### Вступ

Техніки обробки зображень надзвичайно важливі для сучасної науки, особливо в прикладній її частині.

Ці методи вважаються одними з тих, які постійно розвиваються і поліпшуються. Обробка зображення розуміється не тільки як поліпшення зорового сприйняття картинки, а й класифікація об'єктів, яка виконується при аналізі графічної інформації.

Одна з основних операцій в обробці зображень – пошук кордонів об'єкта, розташованого на поверхні.

Пошук меж об'єктів в потоку чи на окремому зображенні використовується в багатьох галузях діяльності та буденних справах людини. Наприклад, в медицині за допомогою такої обробки можна визначити пухлину чи тріщину на рентгенівському знімку, визначити межі та сфокусувати прилад, що веде зйомку, на потрібній ділянці. Камери відеоспостереження за транспортом за допомогою таких технологій можуть визначити ділянку, на якій знаходиться номер машини, що порушила правила, для подальшої ідентифікації. Так як машина рухається, то в моменті стоп-кадру номер автівки і обличчя людини нечіткі. Отже, необхідно спершу визначити ту частину зображення, яку слід обробити щільніше, пом'якшити тон та покращити якість картинки. Також покращення визначення границь зображення може використовуватися для

ідентифікації конкретних написів на стінах або плакатах, що необхідно для комерційних цілей маркетингових фірм.

Найбільш популярні алгоритми для обробки зображень:

- 1) оператор Собеля;
- 2) оператор Лапласа;
- 3) алгоритм Канні;
- 4) оператор Кірша.

Задача модифікованого алгоритма Канні – знаходження та виділення необхідного об'єкта на зображенні та розрахунок позиції цього елемента відносно центра малюнка. Дану задачу для виділення необхідних елементів на зображенні допоможе вирішити сегментація.

Сегментація (якщо брати визначення для технологій) – це процес розділення набору ділянок, які є на малюнку та схожі за деякими ознаками. Також об'єднання цих ділянок (областей) у вихідне зображення. Сегментація необхідна для визначення інформації, що містить у собі картинка. Тобто, після розбиття фотокартки на ділянки можна визначити об'єкти, для котрих необхідно виділити границі. Для сегментації зображення існує багато методів.

Після сегментації необхідно визначити характеристики виділених об'єктів: глибина зображення в цих точках, контраст, колір, перепад пікселів.

### **Постановка задачі**

Створення модифікації алгоритму Канні використанням похідних.

### **Термінологія**

*Алгоритм Канні* – це алгоритм, котрий має у собі за мету виділення контурів предметів на зображенні.

*Межі зображення* – це простір на зображенні, в якому можна побачити найбільший перепад кольору.

### **Опис алгоритму**

План роботи алгоритма Канні.

1. Згладжування зображення. Тобто видаляються шуми, що є на картинці.
2. Пошук градієнтів. Де є найбільший перепад градієнтів, там знаходяться межі кордону (границі об'єкта).
3. Придушення НЕ-максимумів. Знаходяться локальні максимуми функції, та тільки вони визнаються як кордони.
4. Подвійна порогова фільтрація. Потенційні межі визначаються порогоми.

5. Трасування області неоднозначності. Якщо на зображенні є межі зображення, такі що стоять поруч з більш сильними кордонами такі границі згладжуються. Тобто більш сильні межі “придушують” їх.

Щоб перейти до задач, які має виконати алгоритм Канні необхідно зображення модифікувати. А саме – перетворити в відтінки сірого (чорно-біле). Алгоритм Канні може працювати тільки з таким чорно-білим зображенням.

Перетворення зображення з додаванням відтінків сірого проходить так. При обробці цифрових зображень ми маємо справу з кількома відтінками (градієнтами), які можуть бути як відтінками основного кольору, так і зовсім відсутнього на ділянці розглянутого зображення. Ідея запропонованого методу попередньої обробки зображень (ПОЗ) полягає в видаленні «непотрібних» градієнтів при переході від одного пікселя зображення до іншого. Перехід від одного «потрібного» пікселя до іншого буде вважатися лінією контуру, все інше – фоном, який ми й чистимо.

Будь-яке зображення з математичної точки зору можливо вважати двовимірною матрицею  $M(i,j)$ , де  $i$  - кількість пікселів по горизонталі,  $j$  - кількість пікселів по вертикалі.

Кожен елемент матриці  $m[i,j]$  – це цілочисленне значення функції відображення яскравості, в межах від 0 до 255 згідно кольорової моделі *RGB* (*red, green, blue*) – для кожного з кольорів (червоний, зелений, синій).

Для виділення контуру по-перше необхідно перевести зображення в монохромне (градації сірого кольору)  $G(i,j)$ . Для цього ми знайдемо середнє арифметичне всіх складових *RGB*:

$$\square_{\square,\square} = \frac{\square_{\square,\square} + \square_{\square,\square} + \square_{\square,\square}}{3} \quad (1)$$

Далі ми виконуємо означення контуру за формулою:

$$\square\square\square_{\square,\square} = \frac{1}{9} \sum_{-1 \leq \square \leq 1, -1 \leq \square \leq 1} \square_{\square,\square} + \square_{\square,\square} + \square, \quad (2)$$

де  $\square\square\square_{\square,\square} = \{\square_{\square} + \square = \square, \square + \square, -1 \leq \square \leq 1, -1 \leq \square \leq 1\}$  –

потужність множини, яка створюється калькуляцією кольорів пікселя за координатами  $(i,j)$  та кольорів пікселів його сусідів;  $AVG(i,j)$  – це середнє арифметичне елементів цієї множини.

$$B_{i,j} = \begin{cases} 1, & AVG_{i,j} \leq G_{i,j} + |CALC_{i,j}| \\ 0, & AVG_{i,j} > G_{i,j} + |CALC_{i,j}| \end{cases} \quad (3)$$

В результаті ми маємо бінарну матрицю  $R(i,j)$ , в якій значення 1 відповідає лінії контуру, а значення 0 – фоновому зображенню.

Якщо потужність множини  $|CALC(i,j)|$  більша за 4, вважається, що піксель  $(i,j)$  відноситься до шуму та його вага штучно встановлюється дорівнювачій 4, щоб завищити ймовірність його виключення з лінії

контур.

Накладення власних умов на потужність множини може призвести до різних результатів. Дану обставину можна застосувати, коли потрібно отримати більш специфічне рішення в задачах виділення контурів (туман на зображенні, обробка тексту, інше)

### Результати і обговорення

Проведено тестування алгоритмів обробки зображень та зведено визначено які гірше або краще працюють за такими критеріями: час роботи алгоритма, деталізація вихідного зображення, виділення контурів. Також можна розглянути які алгоритми як виділяють границі на зображенні (рис.1).

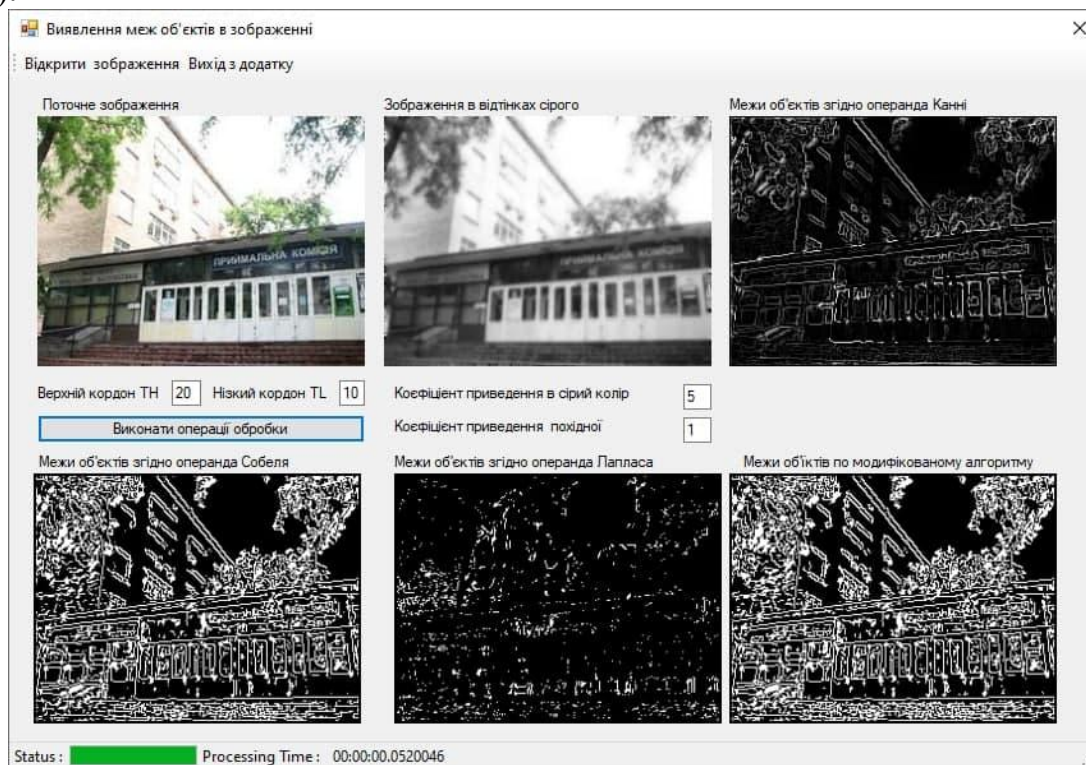


Рис. 1. Результати порівняння

Також за підібраними критеріями можна сказати, що найшвидше працює алгоритм Лапласа (але не чітко виділяє контури) із предиктором, а найповільніше Канні. Щоб його прискорити було вирішено зробити роботу алгоритму з потоками. За показниками якості роботи алгоритмів (деталізація та виділення контурів) результати кращі у методів Канні та Собеля.

Представлений мною алгоритм працює швидше за алгоритм Лапласа, а виділення контурів краще ніж у Собеля.

## Висновки

Так як існуючі алгоритми мають свої недоліки, то було вирішено зробити свій алгоритм, котрий надасть можливість краще виділяти границі та скоріше працювати. Результати показали, що на виході алгоритм добре обробляє фотознімки. Отже, цей алгоритм можна використовувати для обробки зображень (знаходження меж об'єктів) в потоці зображень.

## Література

1. *Canny J.* A computational approach to edge detection / J.A. Canny // IEEE Transactions on pattern analysis and machine intelligence. –Vol. 8, № 6, 1986. –P. 679–698.
2. *Brown University:* pb-lite Boundary Detection, part 1, part 2 [Електронний ресурс]. URL: <http://cs.brown.edu/courses/cs143/2011/results/proj2/vibhu> (дата звертання: 12.10.2021).
3. *Kreit E.* Biological versus electronic adaptive coloration: how can one inform the other? / E. Kreit, L.M. Mathger, R.T. Hanlon, P.B. Dennis, R.R. Naik, E.Forsythe, J. Heikenfeld // Journal of the Royal Society Interface. –Vol. 10 (78), 2013. – 13p.
4. *Kirsh R.A.* Computer Determination of the Constituent Structure of Biological Images / R.A.Kirsh // Computers and biomedical research. – Vol. 4, 1971. –P. 315-328.
5. *Otsu N. A.* Threshold selection method from gray-level histograms / N.A. Otsu // IEEE Trans. Systems. Man and Cybernetics. – Vol. SMC–9. №1. – 1979. –P. 62-66.

Магістрант Стадниченко А.В., ст. викладач Дробязко І.П.

Національний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»

## ПОРІВНЯЛЬНИЙ АНАЛІЗ СПОСОБІВ З'ЄДНАННЯ ПРИСТРОЇВ У СИСТЕМІ РОЗУМНИЙ ДІМ

### Abstract

**Anton V. Stadnychenko, student; Iryna P. Drobiazko, senior lecturer**  
**Comparative analysis of ways to connect devices in a smart home**

*With the increased availability of electricity and semiconductors, smart homes have grown in popularity. More and more buildings use these technologies to automate lighting, temperature, and power supply. Their use increases the energy efficiency of buildings and makes life easier for the people who are in them. For qualitative realization of a smart house it is necessary to approach responsibly to the choice of technologies for the organization of communication among devices. Each technology has its strengths and weaknesses and will not be relevant for every building. This paper presents a survey of such systems.*

### Вступ

За останні роки відбулося значне зростання популярності розумного дому. Розумний дім – це система датчиків і виконавчих приладів, здатних вирішувати певні завдання без участі людини. Розумний дім використовується для допомоги людині у повсякденному житті, здійснюючи автоматизацію процесу керування побутовими пристроями, засобами сигналізації та відеоспостереження, приладами для освітлення, регулювання температури приміщення тощо. За даними США, до 2023 року більше половини будинків будуть використовувати систему розумний дім, а витрати на неї у 2020 році становили 475 мільйонів доларів [1].

Розумний дім може містити велику кількість сенсорів та виконавчих пристроїв і для їх якісної взаємодії слід обрати найбільш придатний спосіб з'єднання, виходячи з особливостей цих пристроїв. На даний час існує багато технологій для об'єднання пристроїв розумного дому.

Кожна технологія має свої сильні та слабкі сторони і її використання є доцільним не для кожного будинку. Правильний вибір технології дозволить позбутися проблем при використанні систем розумного дому.

## **Постановка задачі**

Метою роботи є проведення порівняльного аналізу способів з'єднання пристроїв розумного дому за їх функціональними ознаками, а також формулювання рекомендацій щодо використання різних технологій при реалізації системи.

## **Способи з'єднання пристроїв**

Проаналізуємо найпопулярніші способи з'єднання пристроїв у системах розумного дому.

## **Стандарт KNX**

Стандарт KNX являє собою розвиток більш старого стандарту EIB. KNX – це однорангова, децентралізована мережа з керуванням подіями. Підключені до мережі пристрої обмінюються інформацією через спільний канал (шину). Інформація упаковується, і отриманий пакет передається каналом зв'язку від датчика до виконавчих механізмів. Ключовими відмінностями KNX є гарантована сумісність продуктів різних виробників та єдиний інструмент розробки [2].

До складу KNX входять сенсори, виконавчі пристрої, системні пристрої. Максимальна кількість пристроїв залежить від складності мережі і може становити від 256 до 57600 пристроїв. Для передачі даних можуть використовуватися різні середовища: вита пара, силова лінія, IP-мережа, радіоканал. В залежності від обраного середовища швидкість передачі може варіюватися від 1200 біт/с до 10 мбіт/с. Отже, через велику сумісність, хорошу масштабованість та швидкість, KNX можна використовувати для більшості систем розумного дому.

## **Стандарт CAN**

Інтерфейс CAN розроблено компанією Robert Bosch GmbH у 1980-х роках. У даний час він широко застосовується для автоматизації виробництва, в технології розумного дому та інших галузях. Інтерфейс CAN призначено для організації послідовних, високонадійних та дешевих каналів зв'язку у розподілених системах управління. Інтерфейс має високу завадостійкість завдяки корекції помилок. Блок даних у кадрі має розмір до 8 байтів. CAN має декілька механізмів захисту інформації від помилок; ймовірність пропуску помилки становить  $4,7 \cdot 10^{-11}$  [3]. Це значно підвищує обсяг службових даних у кадрі, що негативно впливає на загальну швидкість обміну даними. Швидкість передачі CAN варіюється від 50 кбіт/с до 1 мбіт/с в залежності від конфігурації шини. Отже, використана CAN є найдоречнішим у системах реального часу, де обсяг даних невеликий, а надійність є пріоритетом. Наприклад, у пожежних системах.

### **Стандарт 1-Wire**

1-Wire – протокол передачі та отримання даних по одному дроту. Найчастіше використовується у картках доступу та цифрових ключах. Через використання одного дроту, протокол дуже чутливий до завад та має низьку швидкість передачі даних (до 16.3 кбіт\с). Це робить його застосування у системах розумного дому дуже обмеженим (з простими датчиками або у якості ключів) [4].

### **Стандарт IrDA**

IrDA – набір промислових специфікацій бездротового об'єднання пристроїв з використанням інфрачервоного зв'язку. Даний стандарт був популярним для реалізації обміну інформації між мобільними пристроями та потім його витіснили Bluetooth та Wi-Fi. Основною перевагою IrDA станом на даний час є його низька вартість. Специфікація підтримує різні характеристики зв'язку:

- швидкість від 2.4 Кбіт до 1 Гбіт в секунду;
- діапазон відстані від 0.2 м до декількох метрів;
- довжина хвилі 850÷900 нм.

В залежності від цих характеристик змінюється й енергоспоживання пристроїв. Використання IrDA у розумних приміщеннях є недоцільним, оскільки пристрої можуть знаходитися у різних частинах приміщення, і тому за допомогою оптичного підходу з'єднати усі пристрої неможливо. Дану специфікацію можна використовувати лише у дистанційному пульті керування, що з'єднується з центральною станцією.

### **Стандарт EnOcean**

EnOcean – бездротова технологія поглинання енергії, що на даний момент використовується для автоматизації приміщень та у багатьох промислових галузях. Модулі, що створені на базі технології EnOcean, об'єднують мікроперетворювачі енергії, які мають ультранизьке енергоспоживання та забезпечують бездротовий зв'язок безбатарейними бездротовими датчиками, контролерами та перемикачами.

Головна ідея, на якій базується технологія EnOcean, полягає в простому спостереженні: у будь-якому місці, де необхідно за допомогою датчиків вимірювати певні параметри, відбувається перетворення енергії з однієї форми в іншу. Це може бути натискання на перемикач, коливання температури або зміна рівня освітленості. За будь-якої з таких подій виробляється достатньо енергії для передачі бездротового повідомлення. Це означає, що більшість компонентів може працювати без акумулятора.



Пакет даних для даного стандарту має довжину 14 байт і передається зі швидкістю 125 Кбіт в секунду. Обсяг необхідної для роботи електроенергії є незначним, оскільки радіочастотна енергія використовується для передачі між пристроями тільки першої частини даних.

Виходячи з цього, використання EnOcean є доцільним для простих датчиків та перемикачів, яким не потрібна передача потокової інформації та великого обсягу даних. Такі пристрої є екологічними та енергоефективними, що особливо важливо для великих приміщень [5].

### **Рекомендації щодо використання різних способів з'єднання пристроїв**

Результати дослідження способів з'єднання пристроїв у системах розумного дому, які представлено у Табл. 1, показують, що усі наведені вище способи з'єднання, окрім IrDA, сьогодні є актуальними для використання у розумному домі, проте кожен з них має власні особливості.

Таблиця 1

Порівняння різних способів з'єднання

Спосіб з'єднання	Швидкість передачі даних	Тип з'єднання	Потоковий обмін даними	Розмір даних у пакеті	Діапазон дії
KNX	0,0012÷10 мбіт/с	Дротовий та бездротовий	+	15 байт	150 метрів бездротовий 1000 метрів дротовий
CAN	0,05÷1 мбіт/с	Дротовий	+	8 байт	250 метрів 1000 метрів при мінімальній швидкості
1-Wire	0,016 мбіт/с	Бездротовий	-	8 біт	Прямий контакт
IrDA	0,002÷1000 мбіт/с	Бездротовий	-	16 байт	0,2÷2 метри
EnOcean	0,125 мбіт/с	Бездротовий	-	20 байт	15 метрів

Вибір конкретної технології залежить від характеристик та призначення приміщення, де буде використана певна система розумного дому. Найбільш доцільним є використання декількох способів з'єднання.

Так, стандарт 1-Wire підходить для створення цифрових ключів та карток доступу до приміщення. Стандарт EnOcean краще підходить для перемикачів та датчиків освітлення, оскільки не потребує додаткового живлення. CAN слід застосовувати там, де потрібен потоковий обмін невеликої кількості даних з високою надійністю, наприклад, у датчиках диму, затоплення та детекторах руху. Для інших пристроїв слід використовувати стандарт KNX через велику сумісність з іншими пристроями, високу швидкість передачі даних та можливість створювати великі мережі.

## **Висновки**

В роботі проведено аналіз існуючих рішень для з'єднання пристроїв систем розумного дому. Як результат аналізу, визначено особливості використання різних технологій та можливості їх застосування при реалізації системи.

Результати проведеного дослідження використано при створенні системи автоматизованого освітлення приміщень.

## **Література**

1. 20 Eye-Opening Smart Home Statistics // [Електронний ресурс]. Режим доступу: <https://techjury.net/blog/smart-home-statistics/#gref>
2. Технология KNX для систем автоматизации // [Електронний ресурс]. Режим доступу: <https://www.ixbt.com/home/knx-intro.shtml>
3. Преимущества и недостатки интерфейса CAN (Controller Area Network, локальная сеть контроллеров) // [Електронний ресурс]. Режим доступу: [https://revolution.allbest.ru/programming/00418443\\_0.html](https://revolution.allbest.ru/programming/00418443_0.html)
4. ПРОТОКОЛ 1-WIRE // [Електронний ресурс] Режим доступу: <https://avr.ru/beginner/understand/1wire>
5. Компоненты для систем EnOcean // [Електронний ресурс] Режим доступу: <https://wireless-e.ru/radiomoduli/komponenty-dlya-sistem-enosean/>
6. Инфракрасный протокол связи IrDA // [Електронний ресурс] Режим доступу: <https://www.ixbt.com/peripheral/irda.html>

**УДК 004.65**

**К.т.н., доцент Тарасенко-Клятченко О. В. , студент Олійник О. В.**

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

## **ВИКОРИСТАННЯ OLAP-ТЕХНОЛОГІЙ ДЛЯ БАГАТОВИМІРНОГО АНАЛІЗУ ДАНИХ**

### **Abstract**

**Oksana V. Tarasenko-Klyatchenko, assoc. prof., PhD; Oliinyk Oleksandr, student**  
*Usage of OLAP-technologies for multidimensional data analysis*

*This paper describes the usage of OLAP-technologies for multidimensional data analysis. The paper presents the results of a general analysis of OLAP technology and an example of building a hypercube. The results of the general analysis of OLAP technology and an example of construction of a hypercube were shown.*

### **Вступ**

У сфері програмного та апаратного забезпечення ведеться постійний розвиток та вдосконалення. На будь-якому підприємстві зараз використовуються сховища даних, бази даних та системи управління базами даних. Разом з удосконаленням інформаційно-комунікаційних технологій загалом удосконалюються і бази даних. Їх структура стає складнішою, а обсяг розв'язуваних задач більшим.

Наприклад, реляційні бази даних дозволяють подавати дані про предметну область за допомогою двовимірних таблиць, між якими встановлюється зв'язок. Вони дозволяють відобразити інформацію у найбільш доступній для розуміння користувача формі. Проте проблеми реляційної бази даних у тому, що її розробка є трудомісткою, а швидкість доступу до даних є низькою. Реляційні бази даних здатні бездоганно обробляти масиви даних, що мають невелику кількість вимірів, але вони не відповідають вимогам глибшого аналізу даних. Саме тому стає актуальною проблема застосування OLAP-технологій, що передбачають використання багатовимірних кубів, зокрема в бізнес сфері.

### **Постановка задачі**

Мета статті полягає в аналізі технології OLAP та її застосування для багатовимірної аналізу даних.

## **Механізм роботи OLAP-технологій**

Багатовимірна система управління базами даних – це тип управління базами даних, який ґрунтується на багатовимірному поданні баз даних. Багатомірні бази даних часто створюються з використанням вхідних даних вже існуючих реляційних баз даних. Така база даних здатна відповідати на запитання на кшталт «скільки товарів було продано в певному регіоні в певний місяць» і аналогічні питання, пов'язані з узагальненням та аналізом бізнес-операцій та тенденцій, тоді як реляційна база даних зазвичай отримує відповідь з використанням мови структурованих запитів. Багатовимірна система управління базами даних має здатність швидко обробляти дані в базі даних так, щоб відповідь могла бути згенерована миттєво.

OLAP (від англ. «online analytical processing») - аналітична обробка в реальному часі) - це технологія обробки, яка дозволяє користувачеві легко і вибірково витягувати і переглядати дані за різними параметрами, з різних точок зору [1]. Набір технологій OLAP включає такі функції як побудова динамічних звітів з різних аспектів, аналіз наявних даних, спостереження, оцінювання та прогнозування основних показників бізнес-процесів [2].

З точки зору концепції, багатовимірна база даних використовує ідею інформаційних кубів для того, щоб подання даних було більш зрозуміле користувачеві. Якщо ж у реляційних базах даних інформація представлена у вигляді взаємопов'язаних таблиць, то в базах даних OLAP інформація представлена у вигляді так званих OLAP-кубів.

В OLAP-кубах зберігаються відомості про взаємозв'язки та про ієрархію даних, що значно спрощує процедуру отримання необхідної інформації та створення звітів, оскільки місцезнаходження необхідних даних та їх взаємозв'язок з іншими даними відомо заздалегідь. Куби даних містять інформацію, відображену в попередньо розробленому вигляді. Це означає, що групування, фільтрація, сортування інформації здійснюється до введення даних. Саме тому вивід та вилучення інформації, що запитується користувачем, стає простою та швидкою дією [3].

## **Застосування OLAP-технологій у бізнес-рішеннях**

У сфері бізнесу є три основні цілі застосування OLAP-технологій: аналіз даних, планування бюджету, фінансова консолідація. Багатовимірна модель даних, можливість проводити аналіз великого обсягу даних та швидке оброблення запитів роблять OLAP-технології безальтернативним механізмом для аналізу продажів, маркетингових кампаній та інших завдань з великою кількістю вихідних даних [4].

Існує декілька різновидів OLAP-систем, одним з найпопулярніших є ROLAP. Relational OLAP – це реляційні OLAP-системи, для яких вхідні дані повинні зберігатися в реляційних БД.

Вхідні дані ROLAP-моделі перетворюються на багатовимірну модель через проміжний шар метаданих. Під час застосування ROLAP результати вибірок зберігаються в тій же РБД, де і їхні вхідні дані. Агреговані дані розташовуються в спеціально створених службових таблицях РБД.

Узагальнений алгоритм застосування технології ROLAP для здійснення багатовимірного аналізу РБД полягає у виконанні наступних кроків:

1. Отримання вхідного сховища даних, представленого у вигляді взаємопов'язаних таблиць РБД.
2. Визначення показників та рівня бажаної деталізації багатовимірного аналізу вхідних даних.
3. Формування багатовимірної OLAP-моделі із вхідної РБД через представлення її структури у вигляді моделі «зірка» або «сніжинка». Визначення структури, кількості та вмісту таблиць фактів і вимірів.
4. Формування структури запитів багатовимірного аналізу, застосування аналітичних операцій для розробленого гіперкубу.
5. Обробка отриманих даних. За необхідністю візуалізація отриманих даних за допомогою різних діаграм і графіків [5].

Бізнес-показники в OLAP-системах зберігаються, як було зазначено вище, не у двовірних таблицях, а у розрізах гіперкубів. Ці розрізи є основними бізнес-аспектами діяльності того чи іншого підприємства: товари, місія продажу, час продажу тощо. Розглянемо для наочності конкретний простий приклад тривимірного OLAP-куба (рис. 1). Інформація сортується за трьома параметрами: час, місто продажу та товари. Ці аспекти називаються вимірами. У кожного з цих вимірів є варіанти вибірки: вимір «місто», «час» та «товари». Варто зазначити, що структура OLAP-куба може бути складнішою. Наприклад, вимір «Час» може містити не тільки квартал, а й рік, тиждень, число, при цьому буде побудована спеціальна ієрархія: рік-місяць-тиждень-число.

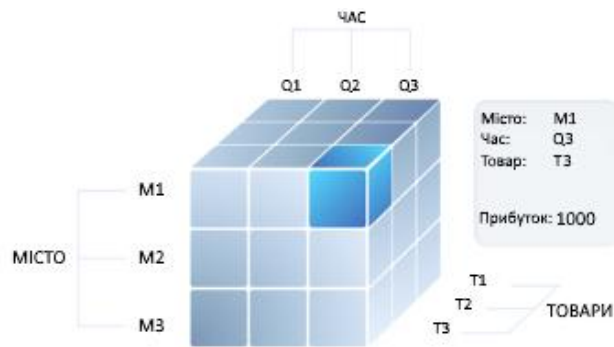


Рис. 1. Приклад OLAP-куба

## Висновки

У роботі представлено результати загального аналізу технології OLAP та приклад побудови гіперкубу. Використання OLAP-систем забезпечує стабільно високий рівень продуктивності та масштабованості, обслуговуючи великі обсяги даних, доступ до яких можуть отримати тисячі користувачів. З допомогою OLAP-технологій доступ до інформації здійснюється у реальному часі, обробка запитів не уповільнює процес аналізу, забезпечуючи його оперативність та ефективність. OLAP-система надає користувачам можливість проводити складний аналіз даних, що дозволяє краще зрозуміти принципи функціонування компанії та знайти способи покращення результатів її діяльності.

Аналітичні програми, побудовані на основі OLAP-технологій, дозволяють вирішувати цілий спектр аналітичних завдань, таких як аналіз виробництва, фінансовий аналіз, маркетингові дослідження, аналіз електронного бізнесу, аналіз трудових ресурсів. OLAP дає можливість створення аналітичних додатків, що охоплюють аналіз усього виробничого та фінансового циклу підприємства.

В той же час, одними із недоліків OLAP є підвищена чутливість до правильності схеми вхідної РБД, визначення індексів та зовнішніх ключів. Аналіз інших показників обробки гіперкубу є об'єктом подальших досліджень.

## Література

1. Chaudhuri, S., & Dayal, U. (1997). An overview of data warehousing and OLAP technology. *ACM Sigmod record*, 26(1), 65-74.
2. Kimball, R.; Ross, M. *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*, 3rd ed.; John Wiley & Sons: Hoboken, NJ, USA, 2013.

3. *Konikov, A., Kulikova, E., & Stifeeva, O.* (2018). Research of the possibilities of application of the Data Warehouse in the construction area. In *MATEC Web of Conferences*, 251(p. 03062). EDP Sciences.
4. *Mansmann, S., Neumuth, T., & Scholl, M. H.* (2007, September). OLAP technology for business process intelligence: Challenges and solutions. In *International Conference on Data Warehousing and Knowledge Discovery* (pp. 111-122). Springer, Berlin, Heidelberg.
5. *Бондаренко, А.В.* (2012). Метод преобразования реляционных данных в многомерные кубы и построение OLAP-отчетов. *Альманах современной науки и образования*. (6), 13–17

**К.т.н., доцент Тарасенко-Клятченко О. В. ,  
студент Бровдій Є. Ю.**

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

## **КЛІЄНТООРІЄНТОВАНІ СЕРВІСИ ДЛЯ РЕКЛАМИ НА ОСНОВІ OPENRTB CLICKHOUSE ТЕХНОЛОГІЇ**

### **Abstract**

**Oksana V. Tarasenko-Klyatchenko, assoc. prof., PhD; Brovdii Yevhenii, student**  
*Client-oriented service for advertising based on OpenRTB ClickHouse technology*

*This paper describes the usage of OpenRTB technology with modern ways of usage ClickHouse database for client-oriented services in advertising. The paper presents the results of a general analysis of OpenRTB technology and an example of usage ClickHouse database for serving advertising. The results of the general analysis of OpenRTB ClickHouse technology were shown.*

### **Вступ**

Ми живемо у світі, який швидко змінюється. Здатність виявляти та застосовувати важливу для бізнесу інформацію з петабайтових наборів даних у режимі реального часу зараз є ключовим фактором у багатьох компаніях. Цифровий маркетинг не є винятком. Насправді цифровий маркетинг зараз є одним з основних джерел використання платформ великих даних. Мережі призначення ставок у реальному часі (RTB) стимулюють високу конкуренцію між рекламодавцями, які вимагають різноманіття технологій та інновацій. У старі часи існувала приказка «Реклама — двигун торгівлі». Тепер можна сказати, що «Технологія — двигун реклами».

LifeStreet була однією з перших компаній у США, яка виявила ClickHouse, і це дозволило їй створити надійну та масштабовану аналітику для ефективної реклами.

Сьогодні для точної оцінки цінності показу оголошення потрібно отримати та структурувати в 100-1000 разів більше даних, ніж десять років тому.

Це тому, що десять років тому кількість даних, які використовувалися для прогнозування цінності показів, обмежувалася трафіком із веб-сайтів видавця, де ведеться цифрова рекламна кампанія,



або з першого покоління мереж видавців. Відтоді мережі призначення ставок у реальному часі (RTV) змінили націлювання на аудиторію та об'єднали видавців і рекламодавців з усього світу в програмному аукціоні RTV. Тепер тисячі рекламодавців роблять ставки за покази від десятків різних видавців у режимі реального часу, і маркетологи мають більше даних, ніж будь-коли, щоб зробити надійні оцінки вартості показу оголошення. Сучасні можливості показів реклами допомагають рекламодавцям зрозуміти ринок і краще оцінити вартість показу.

### **Постановка задачі**

Мета статті полягає в аналізі технології OpenRTV ClickHouse та її застосування в клієнтоорієнтованих сервісах для запуску реклами в мережі Інтернет.

### **Механізм роботи OpenRTV ClickHouse технології**

У традиційних базах даних дані зберігаються за рядками, а коли запитуються дані, потрібен доступ до повних рядків. Це працює для невеликих наборів даних, але стає надто дорогим і повільним для звітів з мільярдами рядків. І навпаки, стовпчасті бази даних зберігають дані за стовпцями, які дозволяють отримувати зі сховища лише ті стовпці, які необхідні для конкретного звіту. Ця архітектура звітності забезпечує набагато краще стиснення даних, зменшуючи обсяг даних, які потрібно обробити одним звітом у 10 000-1 000 000 разів.

Стовпчасті бази даних вперше з'явилися в 1998 році [1], але протягом наступного десятиліття були досить непопулярними – ера великих даних ще не настала. У 2010 році, коли деякі компанії вперше почали використовувати комерційні колонкові бази даних, це були одні з перших рекламних компаній, які використовували цю технологію. Але непомірні ліцензійні збори та недоліки продуктивності в реальному часі обмежували можливість масштабно виконувати аналітику в реальному часі, поки не виявили ClickHouse у 2016 році.

ClickHouse була першою системою керування базами даних, яка відповідала всім вимогам LifeStreet:

- Приймання даних у реальному часі із сотень рекламних серверів
- Дійсно швидкі спеціальні звіти. Вона значно перевершувала інші технології.
- Масштабованість. Швидке масштабування кластеру до 5 ПБ даних.
- Простота обслуговування. Один DBA може підтримувати багаторегіональні високодоступні кластери.
- Вартість володіння. ClickHouse є відкритим вихідним кодом, тому оплата лише за обладнання, і ClickHouse не вимагає нічого.

Спочатку ClickHouse був розроблений Яндексом, російською компанією, яку часто називають російською Google. Яндекс розробив ClickHouse, щоб створити аналітичний додаток для власної величезної рекламної мережі. Було вирішено зробити його платформою з відкритим кодом, дотримуючись тенденції, встановленої Yahoo, Google, Facebook та іншими інтернет-гігантами, які продовжують випускати багато продуктів з відкритим кодом.

Мабуть, найважливішою особливістю ClickHouse є швидкість запитів у порівнянні з іншими базами даних. Можливість робити запити до таблиць із мільярдами записів можна виконати за лічені секунди, що дає величезну перевагу самому бізнесу. Чим швидше виконуються запити, тим більше статистичних даних можна знайти. З попередніми рішеннями для баз даних маркетингові аналітики LifeStreet часто страждали від «захворювання, яке не варто пробувати». Якщо запит зайняв більше 15 хвилин — це погано [2].

Завдяки ClickHouse дослідження спеціальних даних стало простим і можливим. Сьогодні в LifeStreet ClickHouse використовується різними способами. В основному він використовується для зберігання даних RTB — кількох трильйонів рядків у величезному георозподіленому кластері. Цей кластер щодня обслуговує мільйони запитів і забезпечує звіти та інформаційні панелі, алгоритми машинного навчання та спеціальний дослідницький аналіз. Звіти LifeStreet визнані користувачами одними з найкращих на ринку — завдяки ClickHouse. ClickHouse також використовується для зберігання бюджетів кампаній у реальному часі, даних DMP та необроблених журналів. Використання однієї технології бази даних для кількох завдань зменшує витрати на інфраструктуру та команду. Зі змінами конфіденційності Apple iOS 14 цінність і використання даних зростають в геометричній прогресії.

### **Застосування OpenRTB ClickHouse у бізнес-рішеннях**

Тільки в США налічується понад 250 мільйонів активних користувачів Інтернету, і багато з них використовують кілька Інтернет-пристроїв. Якщо розглядати світ загалом, то ця цифра наближається до 3-4 мільярдів. Звичайно, не всі користувачі мають право брати участь у всіх маркетингових кампаніях, але для ефективного маркетингу рекламодавцям потрібна можливість знаходити та застосовувати важливу для бізнесу інформацію з петабайтових наборів даних у режимі реального часу, щоб визначити, які покази, швидше за все, позитивно реагують на оголошення. І саме система управління даними (DMP) відповідає за збір і зберігання кількох анонімізованих атрибутів користувача, таких як країна, мова тощо, щоб розрізнити різні моделі користувачів [3].

Ці дані зазвичай надаються рекламною мережею, але існують також спеціальні зовнішні системи, які надають додаткову інформацію, пов'язану з користувачем, наприклад, профілі покупців з інтернет-магазинів. Інший спосіб збору даних DMP – це самі маркетингові кампанії. За останні кілька років кількість даних, які обробляються та зберігаються, потроюється кожні 18-24 місяці, досягаючи 300 мільярдів записів на добу під високим водяним знаком [4].

300 мільярдів записів на день означає приблизно 3,5 мільйона записів на секунду. Це дуже багато даних. Кожен запит ставки створює запис із сотнями різних атрибутів, наданих мережею RTB та системою DMP. Розмір запису може становити кілька кілобайт. Ці записи необхідно зберігати в базі даних. Щоб поглянути на це в перспективі, типовий ноутбук має менше одного терабайта дискового простору і може зберігати дані менше ніж за годину.

Завдання полягає не тільки в тому, щоб швидко та надійно зберігати ці дані, але й у тому, щоб зробити їх доступними для подальшого аналізу людьми та алгоритмами машинного навчання. Ці дані можуть бути непотрібними, якщо їх ніхто не використає. Саме тому, ефективний маркетинг вимагає брати велику вибірку даних за різний час, щоб знайти моделі користувачів, створити профілі ефективності, зробити прогнози та оцінити цінність показів [5]. Це означає, що кілька сотень терабайт або петабайт даних повинні бути легко доступними для аналітиків ринку, інформаційних панелей і алгоритмів машинного навчання.

## **Висновки**

В роботі представлені приклади застосування нової сучасної OpenRTB технології на основі ClickHouse бази даних. Завдяки цій технології можна спроектувати клієнтоорієнтовані сервіси для реклами будь-якої складності. Про управління взаємовідносинами з клієнтами написано багато, але можна заощадити багато часу та грошей, усвідомивши – на ранньому етапі – що правильна робота – це більше, ніж просто інвестування в IT чи маркетингові кампанії, орієнтовані на клієнта. Саме тому надійність та стабільність в роботі мають першочергове значення в будь-яких клієнтоорієнтованих сервісах запуску реклами в мережі Інтернет.

## Література

1. Clemens Szyperski,. Component Software - Beyond Object-Oriented Programming – Second Edition. Addison-Wesley / ACM Press, 2002. ISBN 0-201-74572-0.
2. D. M. Yellin and R. E. Strom. Protocol Specifications and Component Adaptors. ACM Transaction on Programming Languages and Systems, 19(2), 1997.
3. F. Arbab. What Do You Mean Coordination? Bulletin of the Dutch Association for Theoretical Computer Science (NVTI). Marzo 1998.
4. Dev, H., & Mishra, S. K. (2011). Design of Data Cubes and Mining for Online Banking System. International Journal of Computer Applications, 20.
5. D. C. Luckham, J. J. Kenney, L. M. Augustin, J. Vera, D. Bryan, W. Mann, Specification and Analysis of System Architecture Using Rapide, IEEE Transaction on Software Engineering, Special Issue on Software Architecture, vol. 21, no. 4, April 1995.

**Д.т.н, професор Терейковський І.А., студент Кравчук О.В.**

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

## **АНАЛІЗ ВРАЗЛИВОСТЕЙ ПРОГРАМНОГО РІВНЯ ОПЕРАЦІЙНОЇ СИСТЕМИ ANDROID**

### **Abstract**

**D.Sc., Professor Tereikovskiy Ihor; Kravchuk Oleksandr, student**

***Analysis of vulnerabilities on the software level of the android operating system***

*This paper lists and divides into categories basing on the source of the problem the vulnerabilities on the software level of the Android operating system. It describes 5 groups of potential sources of threats such as unprofessional development, broken protection of software, insufficient access control, disclosure of confidential data and incorrect validation. Each of these groups broken down into a list of vulnerabilities and description of their causes.*

### **Вступ**

Зростання кількості додатків на платформі Android безпосередньо пов'язане з поширенням смартфонів на базі даної операційної системи. Згідно з даними, 84,8% світових поставок смартфонів у 2020 році займають мобільні пристрої оснащені саме Android [1].

В свою чергу, це спричинило зростання кількості випадків поширення шкідливого мобільного забезпечення. Згідно зі звітом органу безпеки мережі Інтернет за 2019 рік, у 2018 році в день блокувалося в середньому 10573 шкідливих мобільних додатків [2].

Дана ОС приваблює зловмисників через великі об'єми обігу конфіденційної інформації користувачів, до якої вони потенційно можуть отримати доступ. Через постійно мінливу природу екосистеми Android та так звану гонки озброєнь, пов'язану із виявленням та усуненням вразливих місць системи, важливо постійно досліджувати її на предмет безпеки та підтримки найвищого рівня конфіденційності інформації користувача.

Саме тому дуже важливим є завчасно виявити слабкі місця системи, котрі дають можливість зловмисникам отримувати контроль над даними та пристроєм користувача, проаналізувати їх та надати повну картину поточного рівня безпеки операційної системи задля подальшого її покращення.

## **Постановка задачі**

Задача полягає у виявленні та класифікації вразливостей програмного рівня операційної системи Android з метою їх подальшого аналізу та характеристики, що дасть основу для більш ефективного пошуку шляхів усунення наведених проблем.

## **Термінологія**

*ОС* – операційна система.

*Secure Hash Algorithm 1 (SHA1)* – алгоритм криптографічного хешування, що реалізує хеш-функцію, побудовану на ідеї функції стиснення.

*MD5* – 128-бітний алгоритм хешування, призначений для створення «відбитків» повідомлень довільної довжини.

*Сокет* — назва програмного інтерфейсу для забезпечення обміну даними між процесами. Процеси при такому обміні можуть виконуватися як на одному, так і на різних пристроях, пов'язаних між собою мережею.

*Android Debug Bridge* – це універсальний інструмент командного рядка, який дозволяє спілкуватися з пристроєм на базі Android [3].

*WebView* – системний компонент або додаток, який відповідає за відкриття веб-сторінок всередині інших додатків.

## **Аналіз вразливостей програмного рівня**

Розбираючи вразливості програмного рівня операційної системи Android можна спостерігати певні спорідненості у причині виникнення деяких з них. Спираючись на джерела слабких місць та проблеми у безпеці додатків було виділено 5 основних груп вразливостей та наведено наслідки, до яких вони призводять.

Некомпетентність розробників. Перш за все, вразливі зони можуть бути створені самими розробниками програмного рівня. Наприклад, відсутність електронного підпису додатку з використанням унікального ключа надає можливість без перешкод перепакувати додаток та створити над ним надбудову зі шкідливого коду. Наявність підпису є головним критерієм перевірки справжності програми. Крім цього, джерелами небезпеки є використання ненадійних або застарілих бібліотек. Такі бібліотеки схильні мати проблеми у виді різноманітних вразливостей, таких як витоки даних, недієві методи безпеки, тощо. Ще однією проблемою на рівні розробки є надання занадто широких дозволів для програми, навіть у випадках, коли частина з них не використовується. Це спричиняє загрозу неконтрольованого доступу до даних та функціоналу мобільного пристрою.

Недієвий захист мобільного пристрою. Насамперед – це слабкий криптографічний алгоритм. Основний набір алгоритмів криптографії можна розділити на симетричні, асиметричні та хеш-функції. Проблема зводиться до того, що кожна з цих категорій включає кілька алгоритмів, кожен з яких може мати різний рівень стійкості. Наприклад, використання слабких криптографічних хеш-функцій, таких як SHA1 або MD5, або відсутність випадковості в генерації ключів робить програму вразливою до витоку інформації. Ще однією проблемою даної групи є використання мобільним пристроєм несправжнього цифрового сертифікату. Цифровий сертифікат (ЦС) – це електронний документ, який допомагає суб'єктам безпечно обмінюватися інформацією через мережу Інтернет за допомогою інфраструктури відкритих ключів. ЦС містить дві основних складові: ідентифікатор і відкритий ключ шифрування. Ідентифікатор підтверджує справжність сертифікату в той час як відкритий ключ шифрування гарантує, що зашифровані дані зможе розшифрувати лише власник сертифіката. Використання несправжнього ЦС робить канали зв'язку через SSL/TLS небезпечними і вразливими до втрати інформації. Симптомом даної проблеми є наявність X509TrustManager або HostNameVerifier, який не виконує жодної перевірки на дійсність сертифікату. Тобто, не перевіряється оригінальність його підпису, а будь-які помилки сертифікації ігноруються.

Несправний контроль доступу. Це означає часткову або повну відсутність автентифікації при запитах від/до інших програм чи програмних модулів. Однією з проблемних для безпеки програми ситуацій є неявне делегування виконання певної операції (наприклад, проведення транзакції) для іншого додатку. При неявному делегуванні система Android сама, серед доступних програм, визначає додатки, які потенційно можуть виконати зазначені дії. Проблема даної ситуації в тому, що будь-який додаток, який оголошує, що може обслуговувати запитану операцію, потенційно має право на її виконання. Отже, якщо такий додаток є зловмисним, може виникнути загроза, яка називається навмисним захопленням. Інформація користувача, що передається при делегуванні, може бути піддана різноманітним маніпуляціям та витоку. Ще одним вразливим місцем є відкриті можливості на рівні Android Debug Bridge. Android Debug Bridge (ADB) — це універсальний інструмент, який забезпечує зв'язок із підключеним Android пристроєм. При використанні відкритого ADB програма локально спілкується з проксі-сервер на даному рівні через сокети TCP, котрі відкриті на тому ж пристрої, який відкриває ADB сервер. Це надає можливість шкідливим програмам вільно приєднуватися до даного сокету та проводити небезпечні для системи атаки.

Викриття конфіденційних даних. Дана вразливість характеризується, наприклад, використанням мобільним пристроєм ненадійних протоколів для передачі даних. Наприклад, розділ заголовка протоколів транспортування даних, таких як HTTP, містить пари ключ/значення для зберігання операційних параметрів, котра не шифрується при передачі [4]. Тому, при прослуховуванні мережі будь-хто легко отримати доступ до вкладених даних. Ще однією серйозною проблемою є те, що користувачі зазвичай покладаються на буфер обміну для копіювання та вставки даних у різні програми. Однак вміст буфера обміну є доступний для читання та запису всіма додатками ОС. Тому шкідливе програмне забезпечення може виконувати різноманітні атаки на вміст буфера обміну, що в свою чергу може спричинити наслідки, які варіюються від захоплення URL-адреси до ексфільтрації даних і вкраплення шкідливого коду.

Некоректна перевірка введених даних. Це призводить до таких проблем як, наприклад, проблеми з WebView. WebView є важливим компонентом, який дозволяє розробникам використовувати веб-технології, такі як HTML і JavaScript для доставки веб-вмісту в програму [5]. На відміну від веб-браузерів, таких як Chrome, FireFox, тощо, які розроблені відомими компаніями, WebView не проходить ліцензування та надає додатку неконтрольований доступ до мережі Інтернет. Як наслідок, додаток може завантажувати небезпечний веб-вміст, тобто без перевірки отриманого коду. Таким чином, зловмисник може занести шкідливий код через будь-який канал, який додаток використовує для отримання веб-контенту. Крім цього, вагомою проблемою є вразливість до SQL ін'єкцій. Програми, що маніпулюють даними, організовують їх збереження за допомогою баз даних. Через це програма може безпосередньо використовувати вхідні дані для побудови запиту, який виконуватиметься ядром бази даних. Таким чином, без належної перевірки вхідних запитів, зловмисник може вставити шкідливий код в оператори SQL, що надає йому можливість отримати повний доступ до внутрішньої інформації та контроль над нею.

## **Висновок**

Не зважаючи на всі досягнення у галузі безпеки програмного забезпечення, операційні системи досі страждають від проблем з безпекою та конфіденційністю. Забезпечення належного рівня захищеності в ОС Android, домінантній мобільній платформі, є важливим питанням, так як мобільні пристрої часто містять великі об'єми конфіденційних даних, і наявність вразливості у хоча б одному додатку може загрожувати безпеці мільярдів користувачів.



У підсумку, дана стаття наводить та розділяє на категорії, ґрунтуючись на джерелі проблеми, вразливості програмного рівня операційної системи Android. Таким чином було виведено 5 груп: некомпетентність розробників, неідеальний захист мобільного пристрою, несправний контроль доступу, викриття конфіденційних даних та некоректна перевірка введених даних. Кожна з них розбивається на перелік вразливостей, які вона об'єднує.

Результатом даної статті є класифікація та опис вразливостей програмного рівня операційної системи Android. Дана стаття збільшує рівень знань про слабкі місця додатків даної платформи, які призводять до втрати даних та відсутності конфіденційності інформації користувачів. Наведена класифікація та характеристика вразливостей в подальшому може бути використана з метою розробки більш ефективних рішень для усунення описаних проблем з безпекою додатків написаних для ОС Android.

## Література

1. Smartphone Market Share. Available online: <https://www.idc.com/promo/smartphone-market-share/os> (accessed on 14 October 2021).
2. Symantec, I. Internet Security Threat Report 2019. Available online: <https://www.phishingbox.com/downloads/Symantec-Security-Internet-Threat-Report-ISRT-2019.pdf> (accessed on 14 October 2021).
3. Google Developer. Android Debug Bridge (adb). Available online: <https://developer.android.com/studio/command-line/adb> (accessed on 17 October 2021).
4. How To: Use mitmproxy to read and modify HTTPS traffic. Available online: <https://blog.heckel.io/2013/07/01/how-to-use-mitmproxy-to-read-and-modify-https-traffic-of-your-phone/> (accessed on 23 October 2021).
5. Android Team. Building web apps in WebView. Available online: <https://developer.android.com/guide/webapps/webview> (accessed on 20 October 2021).

УДК 004.934.1

Д.т.н, професор Терейковський І.А., магістрант Остапчук С.І.

Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”

## АНАЛІЗ ПРОГРАМНИХ КОМПЛЕКСІВ ДЛЯ ПЕРЕКЛАДУ ЗВУКОВИХ ДОРІЖОК ВІДЕОФАЙЛІВ

### Abstract

*D.Sc., Professor Tereikovskiy Ihor and Student Ostapchuk Stepan  
Analysis of software complexes for translation of video file tracks*

*The article discusses the methods that will help to translate the video. The paper also offers its own method of translating a video file. Through the proposed program, you can efficiently and easily translate videos using an easy interface. The ways of further work on improving the program are also given.*

### Вступ

В даний час розвиток інформаційних технологій є не тільки найважливішою областю знань, але і механізмом функціонування та розвитку соціальних відносин. На сьогоднішній день практично кожна організація має власний web-сайт або web-додаток. Вони є інструментом для ознайомлення з тією чи іншою організацією.

В умовах великої кількості інформації, різної по типу та призначенню, весь світ шукає найрізноманітніші засоби для більш доступного споживання інформації в тому числі і відеоматеріалів. Багато корисного відеоматеріалу записано найрізноманітнішими мовами світу. Тому в даній статті розглянуті програми, які можуть допомогти користувачу перекладати звукові доріжки відео файлів , для кращого засвоєння інформації.

### Постановка задачі

Основні задачі, які повинен виконувати програмний комплекс:

- Вивантаження відео на сервер;
- Завантаження перекладеного відео;
- Запит на переклад відео, додавання мови з якої та на яку перекладати;
- Запит на список доступних мов для розпізнавання;
- Запит на список доступних мов для перекладу;

- Запит на список всіх серверів, для обробки відео;
- Запит на найменш завантажений сервер.

Застосунок повинен у повному обсязі надавати клієнтським додаткам інформацію про вільні сервери для обробки відео, список завантаженості всіх серверів, метод для завантаження та вивантаження відео, інформацію про списки доступних мов для розпізнавання та перекладу та найголовніший метод для перекладу звукових доріжок з вказаної мови оригіналу на мову перекладу. Додаток повинен легко масштабуватися, бути мультимовним, коректно працювати на будь-яких пристроях, бути простим та зрозумілим у використанні для широкої аудиторії.

### **Аналіз існуючих програм-аналогів**

В даний момент часу не існує програм аналогів, що б виконували функції даного застосунку, тобто що б автоматизовано в відеофайлі будь якого розширення перекладали звукову доріжку, з вказаної мови оригіналу на мову що обрав користувач для себе. Те що на даний момент не має програм аналогів передумовлено низкою проблем. В аудіо доріжках які перекладаються можливі зупинки в мовленні, а також в різних мовах слова звучать по різному, та мають різну довжину, а отже переклад може бути не коректний.

Зараз існують сервіси з розпізнавання аудіо доріжки, такі як:

- Google speech to text,
- Amazon speech to text,
- Talkatoo speech to text.

Сервіси для озвучування тексту:

- Google text to speech,
- Amazon text to speech,
- Azure text to speech.

Тобто сервіси що виконують окремі задачі, а не обробку одночасно, що розробляється в даному застосунку.

Google Speech-to-Text[1] дозволяє розробникам конвертувати аудіо в текст, застосовуючи потужні нейромережеві моделі в простому у користуванні API. API розпізнає понад 120 мов та варіантів для підтримки глобальної бази користувачів. Ви можете ввімкнути голосові команди та управління, переписати аудіо з центрів дзвінків тощо. Він може обробляти потокове або записане аудіо в режимі реального часу, використовуючи технологію машинного навчання Google.

Google Cloud Text to Speech перетворює текст у людську мову більш ніж 180 голосами на 30+ мовах та варіантах. Він застосовує новаторські дослідження синтезу мови (WaveNet) та потужних нейронних мереж Google для доставки високоякісного звуку. За допомогою цього простого у

користуванні API ви можете створювати реальні взаємодії зі своїми користувачами, які трансформують обслуговування клієнтів, взаємодію пристрою та інші програми.

Amazon speech to text[2] (Transcribe) використовує процес глибокого навчання, відомий під назвою автоматичне розпізнавання мови (ASR), для швидкого і безпомилкового перетворення мови в текст. Amazon Transcribe можна використовувати для розшифровки дзвінків в службу підтримки клієнтів, автоматизації створення субтитрів і генерування метаданих для мультимедійних ресурсів, щоб на їх основі отримати архів з можливістю повнотекстового пошуку.

Amazon Transcribe автоматично розставляє розділові знаки і застосовує форматування, завдяки чому отриманий результат за якістю майже не поступається результату ручної розшифровки, при тому, що час і витрати скорочуються.

Amazon speech to text (Polly) - це сервіс, який перетворює текст в природну мову, дозволяючи розробляти додатки і створювати принципово нові категорії продуктів з підтримкою мовних інтерфейсів. Завдяки просунутим технологіям глибокого навчання сервіс Polly для перетворення тексту в мову (TTS) дає можливість реалістично синтезувати людську мову. Сервіс включає в себе безліч різних максимально реалістичних голосів для різних мов, а також дає можливість створювати додатки з підтримкою мовних інтерфейсів для різних країн.

Talkatoo програмне забезпечення для мовлення, яке перетворює його в текст, заощаджуючи своїм клієнтам (зайнятим професіоналам, таким як лікарі, ветеринари та адвокати) час, щоб вони могли займатись своїми справами. Talkatoo працює в будь-якому текстовому полі так само, як клавіатура; Його можна використовувати в програмних системах управління текстом, Microsoft Office, Google Docs та в Інтернеті. Talkatoo скорочує час документування навпіл. Результат дуже точний, тому користувачі витрачають менше часу на виправлення своїх приміток.

Azure Text to Speech - перетворює текст в аудіо майже в режимі реального часу, дає можливості адаптувати його, змінювати швидкість мови, крок, гучність тощо.

## **Висновок**

У даній науковій роботі проведено аналіз програмних комплексів, які використовуються для перекладу звукових доріжок відео файлів. В ході пошуку було виявлено що на даний момент не існує програм, які б повністю перекладали відео-файли. За допомогою наведених в статті програм можна перетворити звукову доріжку в текст, перекласти на

потрібну мову, та перетворити переклад в звукову доріжку, але такий підхід є не зручний для користувача. Тому вирішено розробити програмний комплекс, який задовільнить вимоги по перекладу, та буде простий в використанні для кінцевого користувача.

## **Література**

1. Schwenk H. Training Neural Network Language Models On Very Large Corpora : праці наук. конф. Joint Conference HLT/EMNLP, 6—8 жовт. 2005, Ванкувер, Канада.
2. Schwenk H. Neural Network Language Models for Conversational Speech Recognition : праці. міжн. наук. конф. International Conference on Speech and Language, 4—8 жовт. 2004, Чеджу, Корея.

УДК 519.688

Д.т.н., проф. Терейковський І.А., студент Парфіненко Ю.М.

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

## АЛГОРИТМ ОБРОБКИ ВІДЕОПОТОКУ

### Abstract

Tereikovsciy Ighor, prof., DcS; Parfinenko Yuriy, student  
*Video processing algorithm*

*This article describes the principle of operation of the algorithm 25 frames. And also results of practical tests of its use are resulted. Different volunteers and different types of influence of this algorithm were used for greater accuracy. For comparative results, preliminary research results on this topic were considered*

### Вступ

Задача провести практичні тести алгоритму, так званого, 25-кадру та перевірити його вплив на добровольців. Даний алгоритм, як відомо, може впливати на підсвідомість людини та нав'язувати їй певні думки чи емоції.

Задачею експерименту було перевірити дійсність впливу даного алгоритму на підсвідомість людини, для цього були запропоновані різні зашифровані повідомлення чи зображення в відеоряді.

### Постановка задачі

Задача полягає в практичній реалізації та тестуванні так званого алгоритму 25-кадру, щоб перевірити його вплив на людей які будуть дивитися запрограмований відеоряд.

### Термінологія

*25-й кадр або сублімінальна реклама* — вигадана методика впливу на свідомість і підсвідомість людей за допомогою вставки в відеоряд прихованих повідомлень у вигляді додаткових кадрів. Автор методу Джеймс Вікері визнав, що результати експериментів, які нібито підтверджували наявність такого впливу на людей, були сфабриковані [5].

*Відеоопсихокорекція* - закодовані образи, сюжетні картинки та слова закладаються у відеоряд, який людина переглядає. Спеціальний алгоритм дозволяє вбудувати будь-яку картинку в будь-який більш менш тривалий відеоряд. Цей метод називається диспаратним уведенням. При цьому

жодними існуючими способами, крім спеціального комп'ютерного декодера, виявити це зображення не вдається. Несвідоме людини неминуче реагує інформацію, закладену в прихованій картинці, а усвідомлення її і навіть підозра у її присутності немає.

### **Опис роботи**

Ідея полягає в тому, що зір людини нібито здатний розрізнити не більше ніж 24 кадру в секунду. Тому чужорідний кадр, що показується менш ніж на 1/24 секунди, нібито минаючи свідомість, впливає відразу на підсвідомість. Насправді ж через підсвідомість проходить вся інформація, яка надходить в мозок, а потім для обробки цієї інформації, яка буде сприйнята як найбільш важлива, підключається свідомість. Таким чином відсіюється величезна кількість інформації, яка за тривалістю сприйняття може значно перевищувати 1/25 секунди (наприклад, звичайна телевізійна реклама), а отже, «прихована» реклама вже в будь-якому випадку менш продуктивна, ніж звичайна [2, 4].

Насправді 25-й кадр прихованим не є: кожен кадр відзначається оком спостерігача, але через інертність зору зливається з подібними і не виділяється людиною. На сьогоднішній день існують камери які можуть знімати з частотою більшою ніж 24 кадри в секунду, наприклад 30, 60 або навіть в 96 кадрів. Сучасні ж програми можуть видавати частоту кадрів більше сотні і для цього існують монітори з частотою оновлення понад 360 кадрів. Тому завдяки цьому ж ефекту помітити «зайвий» кадр не складає труднощів, він сприймається як щось лишнє у відеопотоці і відрізняється від основного. Можна навіть прочитати коротке слово, якщо воно набрано великим шрифтом і знайоме глядачеві — в цьому легко самостійно переконатися, використовуючи домашній комп'ютер і програму відеомонтажу.

Що стосується психологічного ефекту, то його наявність ще в 1958 році була офіційно спростована Американською психологічною асоціацією. Однак на сьогоднішній день існують люди, що продовжують вірити в наявність так званого 25-кадру, що в свою чергу збільшує тривогу в правлінні деяких країн. Зокрема навіть в Україні «25-й кадр» теж прийом заборонений. Зокрема Законом «Про рекламу» забороняється «використовувати засоби і технології, що діють на підсвідомість людини», і новою редакцією Закону «Про телебачення та радіомовлення», «забороняється використання в програмах і передачах на телебаченні та радіо прихованих уставок, які впливають на підсвідомість людини» [2].

За допомогою сьогоднішніх редакторів відео досягнути даного ефекту не є складним, оскільки вони дозволяють редагувати кожен кадр окремо. Існує велика різноманітність відеоредакторів, вони також

поділяються на різні категорії, а саме безплатні, платні, офлайн чи онлайн. Найбільш популярні це продукція компанії Adobe, але для даної роботи використовувався редактор Sony Vegas Pro.

Розглянемо роботу алгоритму на прикладі, для цього уважніше розберемось з роботою відеоредактору на рис. 1. На даному зображенні ми бачимо 2 паралельних відеоряду на один з них вставляємо необхідний нам кадр, а саме на місце 25 кадру та повторюємо його, або міняємо на наступний, протягом всього відео ряду. Зробити це дуже просто оскільки у нас є поточний час відеопотоку та нумерація поточного кадру, між якими можна переключатися та накладати будь-які ефекти. Для повноти ефекту можна убивати прозорість кадру, щоб він менше виділявся на фоні основного потоку відео. Для повноти експерименту було вирішено розділити все послання на деталі та розставити його протягом всього відео.

Були проведені й інші експерименти, що показали, що підпорогова стимуляція впливає на розпізнавання та запам'ятовування об'єктів, викликаючи різні специфічні та дифузні ефекти. Наприклад, багаторазово повторений підпороговий стимул випробувані оцінювали позитивніше, ніж одноразово продемонстрований.

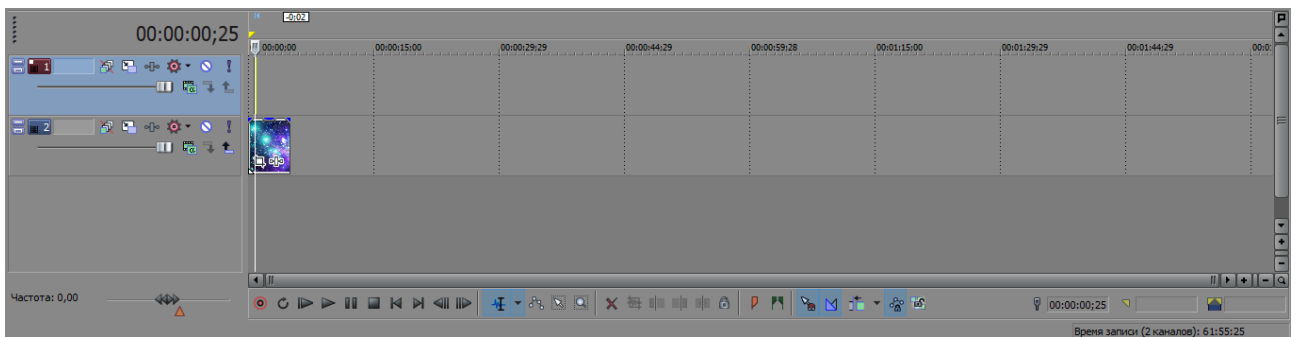


Рис.1

## Висновки

Результати даних експериментів дозволяють зробити такі висновки. Не змінюючи безпосередньо поведінки, підсвідомий вплив у візуальних засобах масової інформації може, як мінімум, збуджувати емоції та впливати на оцінки зовсім інших стимулів. У перспективі це може активно використовуватись, зокрема, комерційною та політичною рекламою, однак дана можливість заборонена в більшості країн світу.

Ця галузь досліджень розвивається сьогодні надзвичайно активно - за рахунок досягнень когнітивних дисциплін та нейронаук. І найближчими роками перед суспільством вже реально може виникнути проблема сублінальних впливів.



На відміну від експериментів, у рекламних кліпах підпорогові стимули накладаються на аудіовізуальний зміст телевізійного рекламного ролика або фільму, на якому зосереджена свідомо увага аудиторії. Якщо свідоме зосередження на чомусь одному перешкоджає впливу підпорогових стимулів, що одночасно пред'являються, то їх використання не дасть практичного ефекту. Це встановлено експериментами.

Було показано студентам-добровольцям двоххвилинний відеофільм. На екрані оберталися у вертикальній та горизонтальній площинах зображення у різних ракурсах барвисті види обставлених меблями кімнат. Глядачі не знали про те, що більш ніж у 12 місцях у фільм вставлені сублімінальні кадри, які демонструвалися протягом шістнадцятої частки секунди. Після показу фільму проводилися виміри тривожності. Ті, кому були показані криваві образи, повідомляли про більшу тривожність, ніж ті, кому показали нейтральні образи, а ті, кому дісталися веселі образи з мультфільмів, повідомляли про значно меншу тривожність, ніж навіть контрольна група.

## Література

1. Смирнов И., Безносюк Е., Журавлёв А. *Психотехнологии: Компьютерный психосемантический анализ и психокоррекция на неосознаваемом уровне.* — М., «Культура», 1995.
2. Закон України «Про телебачення і радіомовлення», ст. 6 п. 3.
3. Anthony R. Pratkanis. The Cargo-Cult Science of Subliminal Persuasion. The Committee for Skeptical Inquiry. Vol. 16.3, Spring 1992.
4. Євгеній Вотяков. Деякі результати дослідження питання про ефект «25-го кадру». // Вісник КрасГАСА, вип. 7, 2004.
5. 25-й кадр. URL: [https://uk.wikipedia.org/wiki/25-%D0%B9\\_%D0%BA%D0%B0%D0%B4%D1%80](https://uk.wikipedia.org/wiki/25-%D0%B9_%D0%BA%D0%B0%D0%B4%D1%80) (дата звернення 29.10.2021)

**К.т.н. доцент Тесленко О.К., магістрант Перегудов М.С.**

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

## **АНАЛІЗ ПІДХОДІВ ТА МЕТОДІВ АНТРОПОГЕННОГО ВПЛИВУ НА ПРОГРАМНІ КОМПЛЕКСИ**

### **Abstract**

*Oleksandr V. Teslenko, assoc. prof., PhD; Mykola Perehudov, student  
Analysis of approaches and methods about the anthropogenic impact  
on software packages*

*This paper considers approaches and methods of anthropogenic impact on software. Dependability is a measure of a systems reliability, availability, maintainability, safety and security. Fuzzing considered as one of the main automated methods to discover potential threats in the software execution flow.*

### **Вступ**

Більше чим за півстоліття після появи перших комп'ютерів теорія надійності сформувалась як класична наука і набула суттєвого розвитку стосовно технічних систем взагалі та апаратури КС зокрема. Створено потужний арсенал методів і засобів оцінки та забезпечення надійності, який пройшов всебічну апробацію, і продовжує удосконалюватись [1]. В той же час розвиток теорії надійності стосовно програмних засобів, мережових технологій та web-систем має скромніші результати [2]. Важливою причиною такого стану є значне зростання питомої ваги негативних антропогенних впливів (людський фактор) [3]. порівнюючи з негативними фізичними (техногенними, природними) впливами, які більш характерні для технічних систем. В результаті з розвитком інформаційних технологій все більша доля в відмовах КС на виконання необхідних сервісів припадає на негативні антропогенні впливи.

Проблема захисту від атак ґрунтується на інформаційній стійкості програмних систем, що використовуються шкідливим програмним забезпеченням.

Під інформаційною стійкістю в інформаційних технологіях мається на увазі гарантоване забезпечення сервісів і протидія їх використанню не за декларованим призначенням. Таке визначення є досить загальним і дає

підстави включати в себе цілий спектр проблем, починаючи від проблеми забезпечення надійності технічних (апаратних) засобів, надійності програмного забезпечення і до проблем протидії хакерським атакам (кіберзлочинність) та ін.[1].

Історично склалось так, що найбільше зусиль для забезпечення інформаційної стійкості було направлено для КС критичного застосування, пов'язаних з ризиками значних матеріальних або фінансових втрат, а також з людськими жертвами [4]. До таких систем належать, наприклад, енергоблоки АЕС, аерокосмічні системи та банківські системи. В англійській літературі для напрямку забезпечення інформаційної стійкості КС критичного застосування використовується термін "dependability". В вітчизняній науково-технічній літературі в цьому випадку використовується термін «гарантоздатність» [5]. Принципи побудови та реалізації гарантоздатних КС активно досліджувались на протязі останніх десятиліть. В базовій статті першого випуску спеціального журналу IEEE "Transactions On Dependable and Secure Computing" [6], було узагальнено результати розвитку гарантоздатності КС.

Маючи на меті вдосконалення інформаційної стійкості пошук критичних ділянок можливий через тестування програмного додатку. Зазвичай тестування програм виконується для перевірки відповідності поведінки сервісу до сценаріїв зазначених у специфікаціях.

### **Постановка задачі**

Задача полягає в аналізі існуючих методів пошуку антропогенного впливу на програмні комплекси.

### **Основна частина**

На основі проведеного аналізу можна виділити наступні методи дослідження інформаційної стійкості ПЗ:

#### **1. Ручний пошук**

Базується на пошуку функцій, вхідні дані для котрих можна підмінити з метою отримання незапланованого результату.

#### **2. Пошук по шаблонах**

Автоматизований метод пошуку типових не стійких сценаріїв на основі сигнатур. Такі сигнатури можуть бути доповнені набором евристичних правил.

#### **3. Fuzzing**

Автоматизований метод пошуку на основі динамічного синтезу вхідних даних. Вхідні дані можуть бути повністю або частково синтезовані:

- з використанням заздалегідь підготовлених даних
- зміною структурованих даних протоколу
- порушенням формату даних в бінарних протоколах
- повним перебором мутацій даних

Результатом антропогенного впливу на роботу програмного комплексу не завжди являє собою порушення безпеки. Порушенням безпеки є збій, що може призвести до будь-якої з наступних можливостей:

- віддаленого виконання коду;
- обходу політик безпеки;
- ініціювання відмови в обслуговуванні.

Що дозволяють впливати на інформаційно-технічний стан (ІТС) [7] КС спричиняючи перекручення інформації або несанкціонований доступ до неї (що не допускається за умовами її використання відповідно до технічного завдання). Це можна тлумачити як перехід системи у несправний, непрацездатний, частково працездатний або небезпечний стан залежно від обсягу та наслідків такого впливу.

## **Висновки**

Для автоматизованого пошуку методів антропогенного впливу на програмні комплекси доцільніше використовувати метод пошуку на основі динамічної зміни вхідних даних, адже різноманітність функцій окремих компонентів комплексу відрізняється в загальному випадку і не може бути детально досліджена вручну або за допомогою пошуку по сигнатурам.

## **Література**

1. Romankevich A.M., Romankevich V.A. Diagnosis of multiprocessor systems under failure of more than half processors // Automation and Remote Control. – 2017. – Vol. 78, Issue 9. – P. 1614 – 1618
2. Andrew Hoffman. Web Application Security: Exploitation and Countermeasures for Modern Web Applications. — Highway North, Sebastopol: O'Reilly Media Inc., 2020 — 291с.
3. Сергієнко А. М. Злонамірено створене апаратне забезпечення // Information Technology and Security. - 2012. - № 1. - С. 93-100
4. "Dependable Systems and Networks".www.dependability.org. Retrieved 2021-06-08
5. В.С. Харченко Гарантоздатність комп'ютерних систем: проблеми та результати // Авіаційно-космічна техніка та технологія.-2005. №7(23). – с.352-357.

6. A. Avizienis, J.C. Laprie, B. Randell, C. Landwehr. Basic Concepts and Taxonomy of Dependable and Secure Computing // IEEE Trans. On Dependable and Secure Computing. – 2004-Vol.1.-№1.-p.11-33.
7. Харченко В.С. Гарантоспособность и гарантоспособные системы: элементы методологии // Радіоелектронні і комп'ютерні системи. – 2006. №5. – С.7-19

УДК 519.1

Доцент Тесленко О.К., магістрант Ходоровський А.П.

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

## ШВИДКИЙ СПОСІБ ВИЗНАЧЕННЯ КЛІКИ НЕРОЗДІЛЕНОГО ЗАВАДОСТІЙКОГО КОДУ

### Abstract

**Oleksandr K. Teslenko, docent; Andryi P. Hodorovskyi, student**

#### *Quick method of determining clique undivided factor code*

*This paper discusses the solution for optimizing the search for undivided factor code. The solution is a module based on determining the next code by counting the number of units in the existing click column. The module responds to a clique and chooses the best way to find the top to save time.*

### Вступ

Завадостійкі коди – один з найбільш ефективних засобів забезпечення високої вірності як при зберіганні, так і при передачі дискретної інформації. Створено спеціальну теорію завадостійкості кодування, що швидко розвивається останнім часом. Під завадостійкими кодами[1] розуміють коди, що дозволяють виявляти або виявляти і виправляти помилки, які виникають у результаті впливу завад. Завадостійкість кодування забезпечується за рахунок введення надмірності в кодові комбінації, тобто за рахунок того, що не всі символи в кодових комбінаціях використовуються для передачі інформації.

Принцип виявлення помилок полягає в тому, що декодер визначає приналежність прийнятої комбінації коду. Якщо комбінація належить коду, то приймається рішення, що помилок немає. Якщо ж прийнята комбінація є забороненою, то приймається рішення про наявність помилки.

Але нероздільні коди не мають чіткого розділення кодової комбінації на інформаційні і перевірочні символи. До них відносяться, наприклад, коди з постійною вагою, коди Плоткіна і ін.

### **Постановка задачі**

Метою даної роботи є порівняння ефективності алгоритмів побудови максимальної кліки і алгоритму прямого перебору допущених вершин.

### **Термінологія**

**$V_n$**  – множина всіх  $n$ -розрядних двійкових чисел;

**Кодослово** – елемент множини  $V_n$  ;

**Відстань Хемінга між двома двійковими числами(кодословами)** – кількість різних двійкових розрядів(кількість одиниць після порозрядної операції XOR).

**XOR (виключне або)** – логічна операція результатом якої є істина лише тоді коли операнди відрізняються.

**Блоковий код  $C(t,n,d)$**  – підмножина із  $V_n$ , з кількістю елементів  $t$  та відстанню Хемінга між будь якими кодословами не меншою ніж  $d$ .

**Граф Хемінга** – неорієнтовний граф, вершини якого належать  $V_n$ , а ребра між вершинами існують при умові, що відстань Хемінга між вершинами не менша за задану( $d$ ).

**Кліка Графа Хемінга** – сукупність вершин, де між будь якими вершинами із кліки існує ребро.

**Повна кліка (повний код)** – кліка, до якої не можливо додати ні одного кодослова.

**Максимальна кліка (максимальний код)** – кліка з найбільшою кількістю кодослів для даного графа Хемінга.

### **Опис вхідних даних алгоритмів**

Для аналізу роботи алгоритмів потрібно використовувати не лише час роботи пошуку побудови максимальної кліки але і кількість вершин у максимальній кліці.

Кількість вершин у максимальній кліці може змінюватись в залежності від вибору першої вершини. Максимальна кількість вершин у графі Хемінга залежить від розрядності кодослів та від відстані між кодословами. Тобто чим більшої розрядності кодослова тим більше послідовностей і варіацій може існувати для кліки. Чим більшої відстані між кодословами нам потрібно дотримуватись тим менша кількість кодослів можлива між вершинами графа Хемінга.

Відстань між кодословами забезпечує завадостійкість у нероздільних кодах. Якщо дотримуватись відстані у мінімум 3 розряди, то кліка може виправити однорозрядну помилку, а якщо дотримуватись відстані більшої за 5, то і дворозрядні помилки.

Для шестирозрядних кодослів максимальною кількістю вершин у графі Хемінга буде 8 вершин, для 8 розрядних – 20, а для 9 розрядних кодослів – 33 вершини.

Ці дані потрібні для пришвидшення пошуку дозволених для додавання вершин у існуючу кліку.

### **Загальні правила для алгоритмів пошуку вершини**

Усі алгоритми повинні включати в себе певну послідовність обов'язкових кроків. До таких кроків належить:

- створення початкового коду потрібної розрядності;
- перевірка значення відстані Хемінга нового кодослова до кожної вершини кліки (тобто виконання операції XOR зі всіма вершинами графа Хемінга та кандидатом на додавання);
- зміна кодослова, якщо воно не підходить;
- додавання коду до кліки якщо він підходить.

Перший, другий та останній пункти не будуть суттєво відрізнятися між різними реалізаціями. А ось підбір та зміна це саме те місце, де і будуть відрізнятися різноманітні реалізації. В залежності від конкретної поставленої задачі ми можемо змінювати та доповнювати алгоритм додатковими умовами, котрі можуть прискорити або покращити якісно побудову максимальної кліки у графі Хемінга.

### **Опис алгоритму прямого пошуку вершини. Алгоритм Брона-Кербоша**

Алгоритм Брона-Кербоша[2] базується на постійному збільшенні не підходящої вершини на одиницю. Початок пошуку потрібної вершини завжди починається з нульового коду 00...00. Незалежно від того, котрі і скільки вершин є у графі Хемінга при невідповідності коду до кліки, алгоритм буде завжди збільшувати код на 1 і виконувати всі попередні кроки заново, допоки не знайде потрібну вершину, котру можна додати або допоки код знову не стане 00...00. «Обнулення» відбудеться наступної ітерації після того як всі розряди вже будуть мати значення 11...11. Це означатиме, що граф Хемінга вже повний і більше до нього вершин додати алгоритм не зможе.

### **Опис алгоритму пошуку вершини зважаючи на сумарну кількість одиниць в стовпці кліки**

Головною відмінністю є запам'ятовування кількості одиниць у графі Хемінга. Тобто до наведеного алгоритму вище додається ще один пункт в кінці: перерахування кількості одиниць у стовпцях кліки для забезпечення швидкого знаходження потрібної вершини. Пришвидшення змінює



початкову вершину в залежності від сумарного співвідношення кількості одиниць та нулів. Теоретично кількість одиниць та нулів у повному Графі Хемінга повинна буде рівною. Це правило дещо порушується при розрядностях кодослів більших за 8.

При такому підході до побудови графа Хемінга отримаємо такі результати між двома алгоритмами на рис.1.

<pre> N of topsNumber of cliq 23  4 24  6 25 11 26 17 27 89 28 82 29 117 30 110 31 53 32 17 33  6 Summary of tops 512 </pre> <p style="text-align: center;">а)</p>	<pre> N of topsNumber of cliq 26  6 27  2 28 48 29 51 30 76 31 205 32 124 Summary of tops 512 </pre> <p style="text-align: center;">б)</p>
--	--

Рис. 1. Порівняння кількості вершин у 9 розрядних кодословах – а) алгоритм пошуку вершини зважаючи на сумарну кількість одиниць в стовпці кліки; алгоритм Брона-Кербоша - б). Перший стовпець вказує на кількість вершин у кліці, другий вказує на кількість таких клік.

При 9 розрядних кодословах робимо висновок, що метод прямого перебору є більш якіснішим і стабільнішим ніж інший. Порівняємо алгоритми в затраченому часі на виконання на рис.2.

Для знаходження середнього часу виконання було вирішено запустити дані алгоритми по 20 разів, потім переглянути їхні результати, отримати сумарно затрачений час на виконання, потім знайти середнє значення за всі 20 повторень. Це дозволить зменшити вплив додаткових завад для знаходження середнього значення.

Передостанній рядок вказує на сумарно затрачений час на роботу алгоритмів, останній на середню роботу алгоритму, а перші двадцять на кожне із повторень.

1	7953952	1	1523645
2	7960164	2	1483917
3	8021927	3	1509758
4	7975305	4	1534126
5	7922013	5	1736752
6	7701674	6	1500933
7	7821502	7	1538276
8	7962546	8	1533267
9	7824375	9	1593413
10	7975642	10	1547708
11	7977611	11	1550877
12	8025301	12	1520075
13	7937382	13	1805364
14	7831654	14	1813812
15	7944182	15	1954702
16	7990751	16	1542960
17	7990234	17	1892439
18	7968843	18	1686879
19	7980903	19	1727323
20	7983196	20	1683734
21		21	
22	158749157	22	32679960
23	7937457	23	1633998

а)

б)

Рис. 2. Порівняння алгоритмів за часом виконання. а) – алгоритм Брона-Кербоша; алгоритм пошуку вершини через сумарну кількість одиниць в стовпці кліки – б).

Звідси випливає, що для алгоритму прямого пошуку вершини потрібно в майже 5 разів більше часу ніж для іншого алгоритму.

## Висновки

В роботі проведено порівняльний аналіз алгоритмів побудови клік з відстанню Хемінга 3 для 9-розрядних кодів. Розроблено програмне забезпечення для перевірки клік на повноту, для перевірки на дієздатність запропонованих алгоритмів. Запропонований алгоритм більш швидкий, але не гарантує в загальному випадку знаходження максимальної кліки.

Шляхами для вдосконалення даного програмного забезпечення є зменшення вибірки вершин на додавання, оптимізація структури програмного забезпечення.

## Література

1. Електронні системи: навчальний посібник / Й. Й. Білинський, К. В. Огороднік, М. Й. Юкиш. – Вінниця : ВНТУ, 2011. -208 с.
2. Johnston, H.C. Cliques of a graph-variations on the Bron-Kerbosch algorithm. International Journal of Computer and Information Sciences 5, 209–238 (1976).

**Магістрант Штефанович Г.М., старший викладач Коляда К.В.**

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

## **СПОСІБ ГЕНЕРАЦІЇ ТА ВИКЛИКУ SQL-ЗАПИТІВ НА ОСНОВІ REST API**

### **Abstract**

**Heorhii M. Shtefanovych, student; Kostiantyn V. Koliada, senior teacher**  
***SQL-query generation and execution method based on REST API***

*In this paper the problem of standard way of developing the database access layer was presented. Along with its solution was discussed why such approach of development and algorithm was designed. The possible pitfalls and problems are studied and main points for improvement are provided. The comparative analysis of method efficiency versus common way was fulfilled. The ways for further research are proposed as well.*

### **Вступ**

Коли йдеться про розробку будь-якого додатку, питання сховища даних завжди стоїть майже на першому місці, а особливо в тих додатках, які оперують великою їх кількістю. Відповідно, постає питання ефективного способу доступу до даних, і тут на допомогу приходять СУБД. І так як вони є повноцінною самостійною системою, кожна з них має інтерфейс доступу до її вмісту.

На сьогодні в Java уже розроблено багато інструментів для доступу до даних з коду додатку. Але, так як вкінці все зводиться до генерації та виклику необхідного для СУБД SQL-запиту, кожен із інструментів просто пропонує свій спосіб реалізації цих операцій.

У цій роботі пропонується новий метод генерації та виклику SQL-запитів. І, так як в програмуванні все стандартизується дедалі більше, то новий метод буде базуватися саме на визначених стандартах та домовленостях ІТ-спільноти.

### **Постановка задачі**

Задача полягає в створенні нового, простішого методу доступу до даних на базі уже існуючих ефективних технологій та стандартів таким чином, щоб використання цього методу було точно не складнішим, ніж існуючі. Просимо зазначити, що завданням не є зробити доступ до даних

швидшим, хоча в деяких випадках він може бути швидшим за стандартні засоби, завданням є зробити розробку швидшою, простішою і гнучкішою. Даний інструмент повинний бути надійним та зробити можливість змін у HTML-запитах додатків незалежнішими від розробників.

### **Термінологія**

*ORM (Object relational mapping)* – технологія програмування, яка зв'язує бази даних з концепціями об'єктно-орієнтованих мов програмування, створюючи «віртуальну об'єктну базу даних».

*REST (Representational state transfer)* – підхід до архітектури мережевих протоколів, які надають доступ до інформаційних ресурсів, це архітектурний стиль для розподілених гіпертекстових систем.

*API (application programming interface)* – набір визначень підпрограм, протоколів взаємодії та засобів для створення програмного забезпечення. Спрощено – це набір чітко визначених методів для взаємодії різних компонентів.

Відповідно, *REST API* – це програмний інтерфейс, який базується на архітектурному підході REST, який в свою чергу використовує мережу Інтернет та протокол HTTP. Головним і єдиним способом комунікації відокремлених один від одного модулів за допомогою такого програмного інтерфейсу є клієнт-серверна архітектура.

*SQL (Structured Query Language)* – декларативна мова програмування для взаємодії користувача з базами даних, що застосовується для формування запитів, оновлення і керування реляційними БД, створення схеми бази даних та її модифікації, системи контролю за доступом до бази даних.

*DTO (Data transfer object)* – один із шаблонів проектування, який використовують для передачі даних між підсистемами програми.

### **Опис алгоритму**

У мові програмування Java є фреймворк доступу до бази даних – Hibernate. Ми обрали саме його, так як на сьогодні це самий популярний інструмент в мові і його використовує більше половини проектів світу, а також він застосовується в інших популярних інструментах, таких як Spring Framework, і його популярність далі збільшується, що дає підстав вважати його стандартом [7]. Врешті-решт, він сам базується на специфікації JPA, від якої ми постараємось не відходити, тож у випадку необхідності можна буде замінити Hibernate на іншу ORM з мінімумом зусиль. Такий набір технологій використовується тому, що він дає можливість розробляти шар логіки доступу до даних незалежно від

вендору РСУБД, отже база даних для нас не має значення, головне щоб вона була SQL типу.

У стандартному підході обробки HTTP-запитів алгоритм дій досить простий:

1. Сервер отримує HTTP запит від клієнта у REST контролері, та конвертує цей запит у DTO;
2. Контролер у свою чергу викликає метод об'єкта (назвемо його Service) рівня бізнес-логіки, передаючи у нього уже конвертовані з запиту дані (DTO);
3. У викликаному методі Service-а викликається метод інтерфейсу доступу до бази даних із уже готовим, написаним попередньо SQL запитом;
4. Отримана відповідь від БД обробляється в методі Service-а відповідно до бізнес-логіки та повертається результатом у контролер;
5. Контролер у свою чергу повертає ці дані клієнту у вигляді HTTP-відповіді.

Схему цього підходу див. на Рис. 1

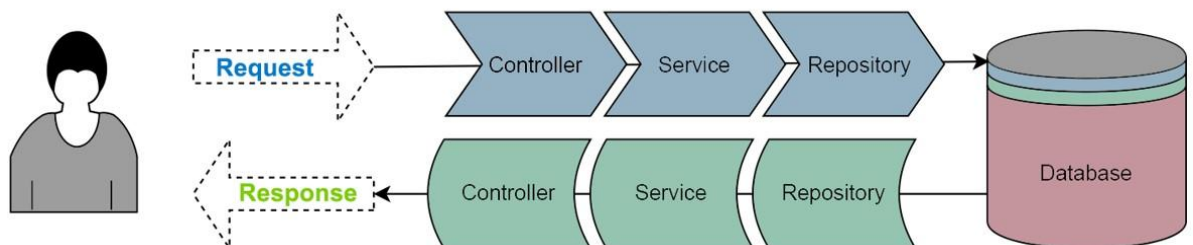


Рис.1. Спрощена схема шляху запиту клієнта

Отже, перейдемо до запропонованого алгоритму. Перші 2 пункти аналогічні пунктам з попереднього підходу, а далі маленька, на перший погляд, відмінність - у методі об'єкта Service викликається об'єкт (нехай RequestManager), який зчитує отриманий DTO, отримує від нього відповідну таблицю бази даних з усіма необхідними для SQL-запиту колонками. На основі даних з HTTP-запиту та відповідної таблиці в БД генерується необхідний SQL-запит, який у відповідь повертає оброблені дані, та повертає їх назад в RequestManager, який в свою чергу повертає їх в Service, після чого виконуються пункти 4 і 5 з попереднього підходу. Детальну схему роботи цього механізму див. рис. 2.

Проблема стандартного підходу захована в пункті 3 його опису. Її суть полягає в тому що запит до бази даних в будь якому вигляді має бути попередньо написаний. Такий метод є дуже ефективним коли йдеться про чітко сплановану розробку, в якій чітко визначена структура бази даних, вимоги всі точно узгоджені, програмний інтерфейс визначений і всі

впевнені, що нічого з цього всього не буде змінюватись з часом [5]. Проте, у сучасному світі команди розробників віддають перевагу гнучким методологіям розробки, наприклад AGILE. Такі методології передбачають часту зміну бізнес вимог до ПЗ, а отже і змін в програмному коді. В цьому випадку запити до БД буде потрібно писати кожного разу коли хоч щось із вищезазначеного в системі змінюється в більшості випадків. Також, дуже велика частина таких запитів це прості CRUD-операції, які в будь-якому випадку потрібно виконувати кожному додатку, а значить і прописувати їх потрібно також.

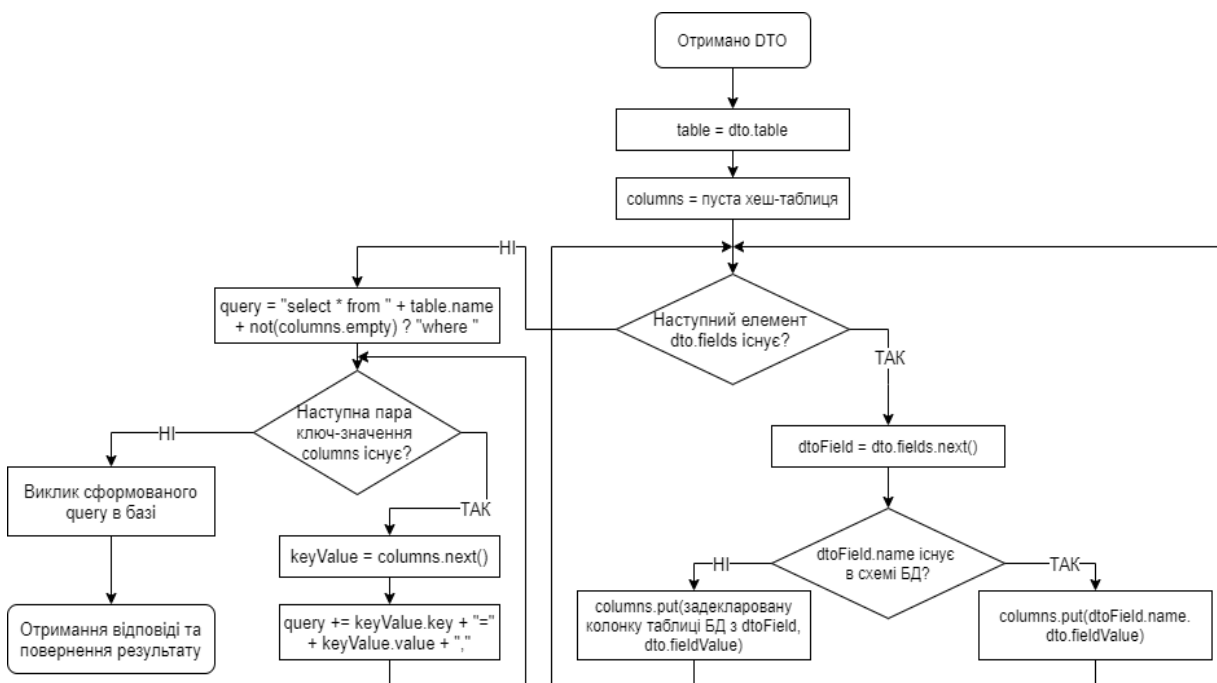


Рис.2. Детальна блок-схема роботи запропонованого алгоритму

Запропонований алгоритм відрізняється тим, що на нього можна перекласти велику частину запитів. Розробникам не потрібно кожного разу прописувати однакові CRUD операції для кожної сутності БД, достатньо всього лише викликати об'єкт, який виконує даний алгоритм, та виконує всю роботу. Також при зміні бізнес-вимог, розробнику не потрібно переписувати заново запити чи змінювати їх, достатньо лише декларативно указати в об'єкті DTO, які з його полів за які колонки таблиць відповідають, викликати алгоритм, і він все згенерує і викличе за Вас.

Звісно, не йдеться про складні запити, які вимагають достатнього вміння їх написання. Проте із заміною звичних, рутинних задач із написання CRUD-операцій даним алгоритмом, можна відняти у

розробників необхідність писати просту, шаблонну логіку, делегувавши її, чим збільшити кількість часу на розробку більш складних і цікавих речей.

## **Висновок**

Якщо порівнювати стандартний метод розробки та запропонований, можна виявити плюси і мінуси в обох. Дуже часто нам приходится розробляти простий інтерфейс доступу до серверних ресурсів, а особливо в мікросервісній архітектурі, і кожного разу приходится повторювати одні і ті ж дії, щоб написати цей інтерфейс, а також інтерфейс доступу до БД.

Запропонований метод дозволяє уникнути написання шаблонних операцій за допомогою автоматизації та збільшити кількість вільного часу розробника, який можна виділити на завдання більшої вартості. Також метод дозволяє будувати програмний інтерфейс який не буде вимагати внесення змін у код, якщо помінялися HTML запити. Сукупність цих факторів не може не зекономити хоч якийсь час.

У подальшому можна реалізувати можливість генерації більшої кількості операторів SQL-запитів, додати кешування запитів, загалом, оптимізувати споживання ресурсів.

## **Література**

1. *R. Martin*. Clean architecture. 2012
2. [https://docs.oracle.com/en/storage/tape-storage/sl4000/slkyp/data-transfer-objects-dtos.html#GUID-37C76672-A90F-4FF4-9754-35704BD8DB40](https://docs.oracle.com/en/storage/tape-storage/sl4000/sl4000/slkyp/data-transfer-objects-dtos.html#GUID-37C76672-A90F-4FF4-9754-35704BD8DB40)
3. <https://www.ibm.com/docs/en/db2-warehouse?topic=reference-sql>
4. <https://docs.oracle.com/en/what-is-the-purpose-of-a-data-access-layer>
5. <https://www.redhat.com/en/topics/api/what-is-a-rest-api>
6. <https://help.hcltechsw.com/commerce/7.0.0/com.ibm.commerce.developer.doc/concepts/csdbusinesslogicbase.html>
7. <https://spring.io/learn>

Ст. викладач Яценко В.І., студент Без'язичний М.В.

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

## ІСНУЮЧІ КВАНТОВІ АЛГОРИТМИ ТА РОЗВ'ЯЗУВАННЯ ЗАДАЧ КОМБІНАТОРИКИ ЗА ДОПОМОГОЮ КВАНТОВИХ КОМП'ЮТЕРІВ

### Abstract

**Vitaliy O. Yatsenko, senior lecturer; Maxim V. Beziazychniy, student**  
*Existing quantum algorithms and solution of combinatoric problems  
on quantum computers*

*This paper investigates and describes already existing quantum algorithms and researches solutions for combinatoric optimization problems with the help of quantum system (computers in this particular case).*

### Вступ

У сучасному світі з'являється все більше і більше інформації так що звичайні комп'ютери вже не поспішають у методах обробки таких об'ємів. Більше того, перед людством ще існує ряд математичних завдань, які здаються не вирішеними при спробах знайти відповідь, використовуючи поточні потужності обчислення, навіть настільки великі як потужності супер-комп'ютерів. Завдяки можливості виконувати дуже складні обчислення значно швидше, квантові комп'ютери здатні надати таке необхідне зростання продуктивності та швидкості.

### Постановка задачі

Метою даної роботи є дослідження та опис основних існуючих квантових алгоритмів та розв'язання комбінаторних задач на оптимізацію за допомогою квантових комп'ютерів.

### Термінологія

*Квантовий комп'ютер* – носій інформації, що використовує квантові об'єкти, а для обчислень квантові об'єкти повинні бути з'єднані в квантову систему.



*Кубіт* – це квантова частка, яка відрізняється від звичайного біта тим, що крім стандартних 0 і 1 кубіт може перебувати між нулем і одиницею – це називають суперпозицією.

*Квантовий оракул* – аналог чорного ящика, що реалізує деяку функцію, але отримує на вхід і видає квантові стани на виході.


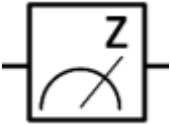

*Квантовий алгоритм* – класичний алгоритм, який задає послідовність унітарних операцій (гейтів або вентилів) із зазначенням, над якими саме кубітами їх треба здійснювати.

*Класи складності P і NP* – клас задач, що можна розв'язати звичайними (P) та недетермінованими (NP) алгоритмами за поліноміальний час.

*Клас складності BQP* – клас задач, що можна розв'язати квантовим комп'ютером за поліноміальний час.

### Види квантових алгоритмів

Розглянемо наведену таблицю квантових вентилів, які надалі будуть використовуватися для опису алгоритмів [1].

Назва	Елемент	
Pauli X		Повертає спін кубіта у протилежному напрямку. Наприклад, для зміни значення кубіта з $ 1\rangle$ на $ 0\rangle$ або навпаки
Замір результату системи		Вимірює значення кубіта та записує його у біт
Перетворення Адамара		$H_m = \frac{1}{\sqrt{2}} \begin{pmatrix} H_{m-1} & H_{m-1} \\ H_{m-1} & -H_{m-1} \end{pmatrix}$

### Алгоритм Дойча (1985) та алгоритм Дойча-Йожі (1992)

Уявімо досить тривіальну задачу: у нас є деяка булева функція. Потрібно за мінімальну кількість звернень до цієї функції визначити чи ця функція константна (на всі входи завжди видає або 0 або 1) або збалансована (для половини входів видає 0, а для іншої половини – 1).

Цю задачу можна вирішувати класичним чином або квантовим. У першому випадку нам знадобиться два запити до «виходів функції»: ми можемо подивитися відповідь на вхідний 0 і на 1, а потім однозначно визначити саму функцію та її тип. Другий варіант рішення – реалізація алгоритму Дойча (рис. 1). Ми конструємо для цієї функції спеціальний квантовий аналог, який називається квантовим оракулом.

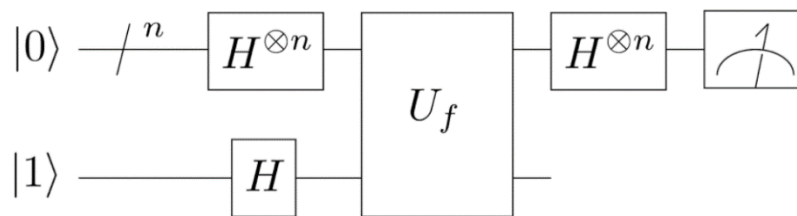


Рис. 1. Реалізація алгоритму Дойча

### Алгоритм Шора (1994)

Алгоритм Шора – це насправді деяка надбудова над іншим базовим алгоритмом – алгоритм пошуку періоду[2]. Ми знаємо функцію, її період слід знайти. І виявляється, що при класичному розв’язку, якщо період досить великий, то все, що ми можемо, – це перебирати варіанти доти, поки не знайдемо два аргументи, які дають одне й те саме значення. У квантовому випадку алгоритм будується інакше. Нам потрібен квантовий оракул, що реалізує цю функцію, а також квантове перетворення Фур'є – аналог дискретного (швидкого) перетворення Фур'є (FFT) (рис. 2). На вхід квантового оракула подаються два регістри кубітів: перший відповідає вхідним значенням функції, другий вихідним. У першому регістрі готується суперпозиція всіх можливих вхідних значень, другий регістр ініціалізується фіксованим станом, після чого на виході отримуємо квантову суперпозицію всіх можливих входів і відповідних їм виходів

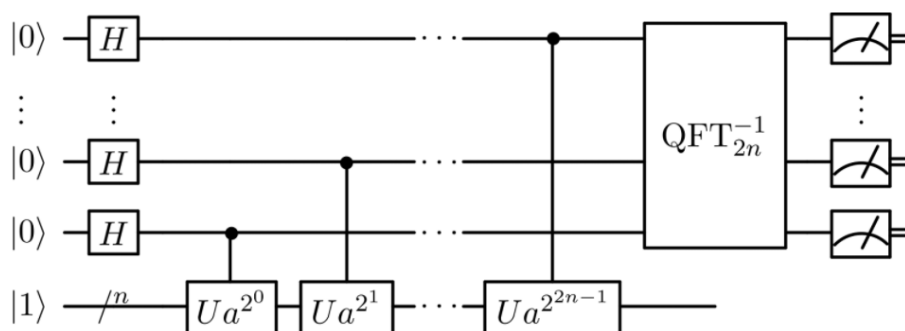


Рис. 2. Реалізація алгоритму Шора

Потім робимо вимір вихідного регістру, у результаті отримуємо деяке випадкове значення функції, а в іншому регістрі – суперпозицію всіх аргументів функції, відповідних отриманому значенню. Далі застосовуємо квантове перетворення Фур'є над першим регістром і в вимірюванні, що зчитує, отримуємо величину, пропорційну зворотному періоду функції. Повторюючи цю операцію кілька разів і використовуючи класичний алгоритм Евкліда для пошуку найбільшого спільного дільника, ми отримуємо період.

### Алгоритм Гровера (1996)

Мета алгоритму – пошук елемента в неупорядкованій базі даних. Можна вважати, що у нас є чорна скринька з якоюсь фіксованою кількістю входів, що дорівнює  $N$ , серед яких є лише один, на який чорна скринька верне 1. Завдання: найбільш оптимальним чином знайти цей елемент. Виявляється, що квантовий алгоритм вирішення такого завдання дає перевагу, оскільки в даному випадку замість порядку  $N$  запитів, які нам потрібні в класичному випадку, знадобиться лише корінь із  $N$  запитів. Якщо в класиці ми вирішували б це завдання, пробуючи один варіант за іншим, то в квантовому випадку алгоритм будується так, що ми створюємо деякий квантовий стан, що містить відразу всі можливі варіанти входів (рис. 3). Потім подаємо його на квантовий оракул, а з виходом робимо деякі перетворення. Потім знову подаємо результуючий стан в оракул і робимо деякі перетворення. Після порядку кореня з  $N$  ітерацій фінальний вимір майже точно покаже на потрібний вхід.

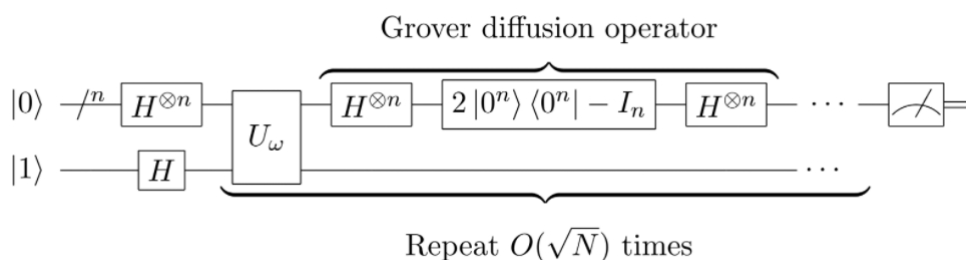


Рис. 3. Реалізація алгоритму Гровера

### Складність задач комбінаторної оптимізації

#### та перспективи квантових комп'ютерів для їх вирішення

Переглянемо деякі типові приклади задач комбінаторної оптимізації.

- Сортування. Дано  $N$  чисел, поверніть їх у порядку зростання.
- Проблема розкладання цілих чисел. Вам дано велике число  $M$ , і ви повинні вказати два цілі множники,  $p$  і  $q$ , таких, що  $M = pq$ .
- Проблема задовільності (SAT). Вам надано логічний вираз багатьох змінних  $x_i \in \{0, 1\}$ , наприклад,  $P(x_1, x_2, x_3, x_4) = x_1 \vee x_2 \wedge (x_3 \vee x_4)$ . Знайти вираження змінних які зроблять вираз істинним. Наприклад, у наведеному випадку  $X_i = 1$  є вірним рішенням

Якщо поліноміальний алгоритм знайде розв'язок одної з них, то у нас буде поліноміальний алгоритм для розв'язання всіх NP задач. Таким чином, стратегія вирішення будь-якої NP-задачі буде: (1) перевести нашу задачу у задачу SAT, (2) вирішити нову задачу SAT, і (3) перевести назад отримане рішення. Для фізиків дуже важливо знати, які задачі комбінаторної оптимізації є NP-повними, а які ні. Це важливо для того щоб

уникнути втрати дорогоцінного часу наївною спробою розв'язати їх за поліноміальний час. Насправді, ми також не очікуємо, що квантові комп'ютери ефективно розв'яжуть задачі, які є NP-повними. Це означало б, що NP міститься в BQP, і ми не віримо, що це так [3]. Проте ми очікуємо, що деякі проблеми, які є в NP і не є в P можуть бути ефективно розв'язані за допомогою квантового комп'ютера, тобто вони належать до BQP[4]. Для вирішення цих проблем квантові алгоритми можуть забезпечити експоненційну перевагу. Крім того можливо ми зможемо вирішити деякі NP-повні задачі на квантовому комп'ютері з квадратичною перевагою порівняно з класичним рішенням.

## Висновки

В ході роботи було розглянуто і описано основні квантові алгоритми та одні із можливих варіантів розв'язування основних задач комбінаторики за допомогою них. Обчислювально складні задачі комбінаторної оптимізації є нерозв'язними навіть для потужних сучасних обчислювальних систем і, ймовірно, залишаться такими, навіть враховуючи зростання обчислювальної потужності.

## Література

1. *Pradosh K. Roy*. Quantum Logic Gates. August 2020. [Електронний ресурс]. – Режим доступу: [https://www.researchgate.net/publication/343833536\\_Quantum\\_Logic\\_Gates](https://www.researchgate.net/publication/343833536_Quantum_Logic_Gates)
2. *Холєво А.С.* Введение в квантовую теорию информации. – М.: МЦНМО, 2002 – 128 с.
3. A complexity theorist's (non)apology [Електронний ресурс]. – Режим доступу: <https://www.scottaaronson.com/blog/?p=206>
4. *Emma Strubell*. An Introduction to Quantum Algorithms. 2011

УДК 004.623

д.т.н., професор Яценко В.О., магістрант Герасимчук Д.М.

Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”

## ОГЛЯД ТЕХНОЛОГІЙ ОПТИМІЗАЦІЇ ДОКУМЕНТООБІГУ НА ПІДПРИЄМСТВІ

### Abstract

**Vitaliy O. Yatsenko, prof., PhD; Herasymchuk Danylo, student**

*Overview of document optimization technologies at company*

*In today's global business environment, the importance of customer service, cost competitiveness, and quality are key factors in determining an organization's success, or undesirable failure. Organizations try to optimize their processes to maximize their profits and make the very process faster. Users usually work with documents in the process. Working with documents makes the process more slowly, since the documents are important to be scanned and attached to the form*

### Вступ

Наразі, на більшості підприємств використовують ERP-системи, які дають змогу оцифрувати багато важливих для компанії бізнес-процесів. Більшість бізнес процесів на підприємствах, в різний спосіб зв'язані з документами. Це зумовлено тим, що більшість операцій повинні мати якесь підтвердження документоване підтвердження, для того щоб мати юридичну силу та захищатися законом. Прикладом таких документів можуть бути: чеки, рахунки, акти, тощо.

Сучасні компанії, відходять від традиційної роботи з документами та з метою подолання технічних, фінансових та швидкісних бар'єрів, тяжіють до використання Систем електронного документообігу (СЕД). Переваги СЕД широко відомі: підвищена захищеність документів, контроль на кожному етапі, надійне резервне копіювання, високий рівень керування документами і низькі витрати на архівування, кращий контроль версій, кращі можливості управління та пошуку.

Об'єднання систем документообігу з системою управління бізнес-процесами дає змогу з'єднати зміст і процеси на всіх рівнях підприємства. Правильна інформація для потрібних людей у потрібний час критичне в оптимізації бізнес-процесів [1].

### **Постановка задачі**

Провести аналіз основних проблем, які виникають при роботі з документами на підприємстві. Розглянути технології та методи які дозволяють усунути описанні проблеми. Запропонувати метод оптимізації документообігу.

### **Основні проблеми, які виникають при паперовому документообігу**

Втрачаються документи. Якщо таке трапляється, то бізнесу загрожують втрати та неприємності.

1. Довгий пошук документів. Через велику кількість документів бухгалтер може годинами шукати потрібний, перебираючи гори паперів.

2. Дорога оренда приміщень. Для зберігання паперових документів потрібний «матеріальний» архів – приміщення, будинок, склад.

3. Відсутність централізованого сховища. Часто архівів в організацій кілька, бо всі документи не розміщуються в одному приміщенні [2]. Це незручно та затратно.

4. Витрати копії паперових документів. Іноді працювати з одним документом необхідно кільком співробітникам, тому доводиться робити екземпляри. А якщо працювати необхідно з оригіналом одного й того самого документа, виникають затримки в часі.

5. Проблеми із заміною версії документів. Якщо документ, відправлений до архіву, пізніше внесли зміни, то потрібно стару версію замінити на актуальну. Бухгалтеру знадобиться у разі знайти потрібний документ, оформити новий, списати старий. А може, таке, що в архіві залишиться стара версія документа, і це вже загрожує бізнесу санкціями.

6. Не можна працювати у реальному часі. Витрачається час на пошук та відправлення документів, доставка з відділу в відділ, що значно знижує швидкість обробки документів, не кажучи про те, що документ може знадобитись відправити в інше місто на підпис контрагенту [3].

7. Складний процес підготовки податкової звітності. Доводиться шукати в архіві необхідні документи, групувати їх, на що йде не одна година.

8. Низький рівень конфіденційності. Практично будь-який співробітник може отримати доступ до документів.

9. Небезпека даних у разі надзвичайної ситуації. Паперовий архів дуже важко захистити у разі пожежі або крадіжки.

10. Вартість паперу. В офісі працівники можуть витратити до 10 000 тисяч паперових аркушів щороку.

11. Утилізація документів. Багато документів які потрібно утилізувати можуть продовжувати займати місце в сховищі документів організації, та на їх утримання будуть витратитися кошти.

### **Способи оптимізації документообігу на підприємстві:**

1. Реорганізація архіву. Представляє собою масштабний проект із попереднім обстеженням системи документообігу в організації. Для цього наймається спеціаліст-аналітик. Після проведення аналізу розробляється план дій для реорганізації. Реорганізація - це впровадження нової системи документообігу, яка буде продовжувати використовувати паперові документи, але в більш оптимальний спосіб [4]. І тут свої недоліки: витрати на наймання фахівця, час на адаптацію співробітників до нових правил роботи (а до цього моменту вони допускать чимало помилок). Немає гарантій, що вирішивши одні проблеми, не виникнуть інші, є ризики завдати бізнесу шкоди.
2. Наймання додаткових співробітників. Таким методом можна оптимізувати лише один конкретний процес. Недолік в тому, що в цьому випадку доведеться розширити фонд оплати праці та підвищити витрати компанії [5]. А ще немає гарантій, що співробітник виявиться кваліфікованим і ви не дарма витратите час та ресурси.
3. Автоматизація. Найефективніший спосіб зробити роботу з документами комфортною та безпечною, а прийняття рішень дійсно швидшим [5]. Щоб вирішити проблеми паперового документообігу, організації впроваджують електронний документообіг. Це дає змогу вирішити всі основні проблеми паперового документообігу, забезпечуючи безпеку, надійність та швидкість обробки інформації. Також, такий підхід дає змогу економити кошти на оренді приміщень, папері. Проте це потребує додаткових затрат на навчання персоналу, та інтеграцію внутрішніх процесів компанії с системами електронного документообігу. Також, часто системи документообігу є не достатньо гнучкими для потреб підприємства.

### **Висновки**

У даній статті наведений опис основних способів рішення питання документообігу на підприємстві. Висвітлені основні проблеми паперового документообігу з метою їх подальшого аналізу і покращення. Наразі можна стверджувати, що підприємствам необхідно відмовлятися від паперового документообігу на користь електронних документів. Проте електронні

документи існують разом з системами документообігу, які також мають свої недоліки, часто це може бути складність в інтеграції процесів. Тому задача спрощення переходу підприємства від паперового документообігу до електронного залишається актуальною.

### **Література**

1. R. K. L. Ko, “A computer scientist’s introductory guide to business process management (BPM)”, in ACM Crossroads, vol. 15(4), Summer 2009, pp. 11–18.
2. W. M. P. van der Aalst, A. H. M. ter Hofstede, and M. Weske, “Business process management: A survey.” in Proc. Business Process Management: International Conference, BPM 2003, Eindhoven, the Netherlands, ser. LNCS. Springer, June 26-27, 2003, pp. 1–12.
3. Smith, Howard; Fingar, Peter: Business Process Management. The Third Wave. Meghan-Kiffer, US, 2003.
4. S. Cleveland. (2016, January, 14). BPM from a Business Point of View, business process management, Greatest Benefit from BPM..
5. P. Fenton. (2016, January, 20) 10 Benefits of Moving to Electronic Document Management System (EDMS).



УДК 519.6

Д.т.н, професор Яценко В.О., магістрант Калитка Є.О.

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

## АНАЛІЗ ТЕХНОЛОГІЇ «ІНТЕРНЕТУ РЕЧЕЙ» (INTERNET OF THINGS, IOT), ЇЇ РОЗВИТОК, СТРУКТУРА ТА ПОДАЛЬШІ ПЕРСПЕКТИВИ

### Abstract

Vitaliy O. Yatsenko, d.t.s. professor; Eugene O. Kalytka, student

*Analysis of Internet of Things (IoT) technology, its development, structure and future prospects*

*This paper discusses the functional features of IoT systems, namely, the logical levels and types of interactions that are characteristic of these networks. In addition, an analysis of the future prospects of such systems and the difficulties of their spread.*

### Вступ

Термін «Інтернет речей» (Internet of Things, IoT) був запропонований у 1999 році Кевіном Ештоном, одним з трьох засновників Центральної автоматичної ідентифікації Масачусетського університету (Auto-ID Center). Існує кілька визначень цього терміну, і кожне з них недостатньо точне. Ми будемо використовувати визначення, запропонований компанією Gartner:

«Інтернет речей - це мережа фізичних об'єктів, які мають вбудовані технології, що дозволяють здійснювати взаємодію із зовнішніми середовищами, передавати інформацію про свій стан та приймати дані із зовнішнього середовища».

Ключовими ж концепціями які реалізуються в системах IoT є взаємодії M2M та M2P.

### Постановка завдання

Проаналізувати сучасні IoT системи та виділити їх основні технічні компоненти, дослідити подальші перспективи концепції Інтернету речей

## **Термінологія**

**IoT** (Internet of Things) – це мережа фізичних об'єктів, які мають вбудовані технології, що дозволяють здійснювати взаємодію із зовнішніми середовищами, передавати інформацію про свій стан та приймати дані із зовнішнього середовища

**M2M** (Mashine to Mashine) – це взаємодія машини з машиною цей взаємозв'язок використовується для обміну інформацією між девайсами, аналізу даних і інколи для систем автоматичного прийняття рішень

**M2P** (Mashine to People) – це взаємодія машини з людиною цей взаємозв'язок як правило використовується для виводу інформації, що була отримана в результаті M2M взаємодії у доступному для користувача виді

## **Технології Інтернету речей**

В основі Інтернету речей лежать наступні технології:

### **Засоби ідентифікації**

Кожен об'єкт фізичного світу, що приймає участь в Інтернеті речей, навіть не підключений до мережі, все рівно має мати унікальний ідентифікатор. Для автоматичної ідентифікації пристроїв можуть використовуватися різні вже наявні системи: радіочастотна, для використання якої до кожного об'єкту прикріплюється радіочастотний маркер, оптична (штрих-коди, матриці даних, QR-коди), інфрачервоні мітки тощо..

### **Засоби вимірювання**

Завдання засобів вимірювання - забезпечити перетворення інформації про зовнішній світ в дані зручні для передачі засобам обробки. Це можуть бути як окремі датчики температури, освітленості і т.п., так і складні вимірювальні комплекси.

### **Засоби передачі даних**

Для передачі даних може бути використана будь-яка з існуючих технологій. У разі застосування безпроводних систем особливу увагу приділяють збільшенню надійності передачі даних. При використанні провідних мереж активно використовується технологія передачі даних по лініях електропередач, оскільки багато пристроїв (таких як торгові автомати, банкомати тощо) підключаються до електромережі.

### **Засоби обробки даних**

П'ятдесят і більше мільярдів пристроїв, які підключені в 2020 році до Інтернету, генерують 44 мільярди терабайт даних. Це приблизно в сім разів збільшує кількість цифрової інформації у всьому світі станом на 2010-й рік. Тому, головною частиною Інтернету речей являються хмарні сервіси, що забезпечують високу пропускну здатність і здатні швидко реагувати на різноманітні.

### **Виконуючі пристрої**

Це пристрої, здатні перетворювати цифрові електричні сигнали, що поступають від інформаційних мереж, в дії. Виконуючі пристрої зазвичай конструктивно поєднуються разом з датчиками.

### **Архітектура Інтернету речей**

- Physical Layer — Фізичний рівень
- Edge Layer - рівень периферійного обчислення
- Local Network Layer – комунікаційний рівень
- Gateway Layer - рівень шлюзу
- Wide Network Layer - рівень зовнішнього зв'язку
- Security Layer - рівень безпеки
- Etl layer - рівень збору, обробки та зберігання даних
- Big Data and Analytic Layer - рівень аналітики
- Notification layer - рівень повідомлення
- Presentation Layer - рівень представлення
- Configuration Layer - рівень конфігурацій

### **Перспективи Інтернету речей**

Уже в наш час Інтернет речей застосовується у найрізноманітніших галузях. Насамперед це промисловість, транспорт (220 млн. підключених автомобілів), розумні будинки, комунальні служби (мільярди датчиків, суттєве зменшення втрат енергії), охорона здоров'я (646 млн. пристроїв, що збирають дані про здоров'я людей), аграрний сектор (75 млн. датчиків для моніторингу стану землі). Крім того, Інтернет речей застосовується у торговій, логістичній, готельній сферах, у банківських системах, у будівництві та у збройних силах (126 тис. Військових дронів та роботів)

## Висновки

У віддаленій перспективі «розумними» стануть не тільки будинки, але і міста, і навіть (деякі) держави. Але на даному етапі розвитку технологій та суспільства Інтернет речей активно впроваджується в масштабах компаній, що займаються виробництвом товарів, енергії, транспортними перевезеннями, словом там, де за рахунок нових технологій очікується підвищення продуктивності та конкурентоспроможності. Складність масштабування даних систем зумовлена тим, що необхідно інтегрувати в єдине цілі багато систем від різних постачальників, і налаштувати їх злагоджену роботу є нетривіальною задачею яка займе багато часу та ресурсів, як технологічних так і людських.

## Література

1. Internet of things (Електронний ресурс): <https://www.redbooks.ibm.com/redpapers/pdfs/redp4975-ru.pdf>
2. Технології безпроводної передачі даних (Електронний ресурс): <https://wireless-e.ru/standarty/tehnologii-besprovodnoj-peredachi-dannyh/>
3. Thingsboard documentation (електронний ресурс): <https://thingsboard.io/docs/>
4. The Internet of Things (IoT): What is the potential of the internet of things (Електронний ресурс): [https://essay.utwente.nl/70018/1/Gong\\_BA\\_BMS.pdf](https://essay.utwente.nl/70018/1/Gong_BA_BMS.pdf)
5. What is an IoT platform? (Електронний ресурс): <https://www.kaaiot.com/blog/what-is-iot-platform>

УДК 004.021

Д.т.н, професор Яценко В.О., магістрант Троян Б.В.

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

## АНАЛІЗ ФЕНОМЕНІВ КОСМІЧНОЇ ПОГОДИ ТА ПЕРСПЕКТИВИ ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ ЇХ ДОСЛІДЖЕННЯ ТА ПРОГНОЗУВАННЯ

### Abstract

**Yatsenko Vitaliy, prof., PhD; Troian Borys, student**  
*Analysis of prospects of neural networks application for research  
and prediction of space weather*

*In this abstract we analyze different methods for research of space weather prediction. Considering increasing relevance of the topic both for scientific and practical applications, we are going to improve methods using latest achievements in IT and artificial neural networks in particular. We are figuring out, which methods will be most suitable to improve neural networks and how it can be done.*

### Вступ

Передбачення космічної погоди є однією з найскладніших проблем при її дослідженні. Зазвичай засоби передбачення тих чи інших явищ спираються на їх моделювання, але у випадку космічної погоди традиційне моделювання дає досить ненадійні результати. Наразі надійним є прогноз на 1 годину, а вже при прогнозуванні навіть на 10 годин точність передбачення зазвичай не вище 50%.

При моделюванні космічної погоди основною проблемою є порівняно невелика кількість доступних вхідних факторів. В свою чергу однією з визначних можливостей нейронних мереж є можливість рішення задач при умовах, коли закономірності в системі невідомі або їх важко математично описати. Так як обмеженість вхідних факторів веде до втрати відповідних взаємозв'язків в моделі, то використання штучних нейронних мереж може дозволити компенсувати відсутність деяких взаємозв'язків.

### Постановка задачі

Задача полягає в аналізі існуючих методик та прийомів передбачення космічної погоди з метою вибору тих, для яких найбільш актуальна модифікація з використанням нейронних мереж та формалізація підходу

для їх використання в тандемі з нейронними мережами, та подальше використання отриманих результатів для розробки засобів прогнозування Dst індексу за допомогою білінійних систем та нейронних мереж.

### **Термінологія**

*Космічна погода* - сукупність явищ, що відбуваються у верхніх шарах земної атмосфери, в іоносфері і навколоземному космічному просторі, та вивчається як галузь космічної фізики.

*Наукове моделювання* - це метод дослідження об'єктів пізнання, що ґрунтується на заміні конкретного об'єкта досліджень (оригіналу) іншим, подібною до нього спрощеною моделлю.

*Штучні нейронні мережі (ШНМ)* - це обчислювальні системи, натхнені біологічними нейронними мережами, що складають мозок тварин. Вони складаються зі штучних нейронів, з'єднаних зв'язками, по яким можуть передаватись сигнали. ШНМ можуть бути навчені на прикладах для розв'язання окремої задачі без жодних попередніх даних про неї.

### **Аналіз сучасних моделей та алгоритмів**

Моделі космічної погоди відрізняються від метеорологічних моделей тим, що обсяг вхідних даних значно менший. За поточного рівня розвитку технологій деякі фактори, зв'язані з сонячною активністю, неможливо врахувати достатньо точно. Внаслідок цього існуючі моделі космічної погоди є досить неточними, так як в них не враховано відсутні вхідні фактори, котрі можуть мати досить значний вплив на результат. Відомі випадки зіткнень штучних об'єктів на орбіті, зв'язаних з неправильним передбаченням їх положення в просторі, на котре впливають в тому числі і явища космічної погоди.

З розвитком науки завдяки появі нових технологій було формалізовано і почато дослідження багатьох показників та феноменів космічної погоди.

Кількість сонячних плям є одних з базових показників космічної погоди. Сонячні плями - це темні ділянки, у фотосфері Сонця, утворені завдяки інтенсивному магнітного потоку, що проривається на поверхню зсередини Сонця. Ділянки вздовж цього магнітного потоку у верхній фотосфері та хромосфері нагріваються і зазвичай стають видимими як факули та плями, котрі називають активними областями. Це створює холодніші, менш щільні та темні ділянки в основі цих магнітних полів, ніж у навколишній фотосфері, що називають сонячними плямами. Досить давно відомо, що загальна кількість сонячних плям змінюється приблизно з 11-річним циклом, відомим як сонячний цикл[1]. Наразі моніторинг

сонячних плям звітується кожного дня, але короткочасні зміни сонячних плям є досить непередбачуваними.

Потік сонячного радіовипромінювання 10,7 см (2800 МГц) (індекс F10.7) є чудовим показником сонячної активності. F10.7 виникає високо в хромосфері та низько в короні сонячної атмосфери. F10.7 добре корелює з числом сонячних плям, EUV та деякими іншими параметрами. F10.7 постійно вимірюється в Канаді з 1947 року. На відміну від багатьох інших параметрів, F10.7 можна легко і надійно виміряти щодня з поверхні Землі за будь-якої погоди. Вимірюється в "одиницях сонячного потоку" (solar flux unit s.f.u.), F10.7 може змінюватись від 50 s.f.u. до >300 s.f.u., протягом сонячного циклу[1]. Завдяки довгій історії записів, а також завдяки простоті вимірювання, що надає значний об'єм вимірювань, індекс дозволяє формувати кліматологію сонячної активності протягом шести сонячних циклів.

Сонячний екстремальний ультрафіолет (EUV) - це сонячне випромінювання, яке охоплює довжини хвиль 10 - 120 нм електромагнітного спектру. Це високоенергетичне випромінювання, воно поглинається у верхніх шарах атмосфери, цим нагріваючи та іонізуючи її, створюючи іоносферу. Термосфера Землі на висоті від 80 до 600 км нагрівається переважно завдяки EUV. Сонячне EUV випромінювання є одним з трьох основних рушіїв іоносферної мінливості. Фотони EUV досягають Землі і повністю поглинаються у верхніх шарах атмосфери вище 80 км. EUV може змінюватись на порядок за різні періоди, від хвилин до годин (сонячні спалахи), днів до місяців (обертання Сонця) та років до десятиліть (сонячний цикл)[1]. Мінливість EUV спричиняє мінливість термосфери та іоносфери. Оскільки EUV поглинається верхніми шарами атмосфери, її неможливо виміряти з землі, тому вимірювання необхідно проводити з ракет і супутників або покладатись на опосередковані показники, такі як кількість сонячних плям або F10.7.

Магнітосфера є дуже важливим фактором в дослідженні космічної погоди. Магнітосфера – це зона космосу довкола Землі, де переважає магнітне поле Землі, а не міжпланетного простору. Результуючий стан магнітосфери формується в процесі впливу сонячного вітру на магнітне поле Землі. Сонячний вітер являє собою потік плазми, що складається з переважно з протонів та електронів і несе в собі власне магнітне поле[2]. Саме варіації сонячного вітру значною мірою зумовлюють мінливість магнітосфери. Для характеристики стану магнітосфери було розроблено значну кількість індексів, серед них Dst індекс, група індексів Kp, Ap, група індексів AE, AU, AL, AO, PC-індекс, та інші.

Dst (*англ. Disturbance storm-time*) індекс геомагнітної активності в низьких широтах був введений в 1964 р як міра зміни поля через кільцеві

струми, що виникають в магнітосфері під час магнітних бур. На земній поверхні вплив кільцевих струмів позначається в зменшенні горизонтальної Н складової магнітного поля з максимальним зменшенням в низьких широтах.

Обчислюється Dst індекс як середня за інтервал часу величина збудження горизонтальної складової напруженості магнітного поля Землі, відносно спокійного рівня. Визначається за даними чотирьох обсерваторій на низькій (але не екваторіальній) широті, рівномірно розподілених по довготі. В магнітоспокійні дні величина Dst лежить в межах  $\pm 20$  нТл. Dst індекс обчислюється і публікується в Світовому центрі даних Кіото по геомагнетизму, кожну годину[3] (рис.1).

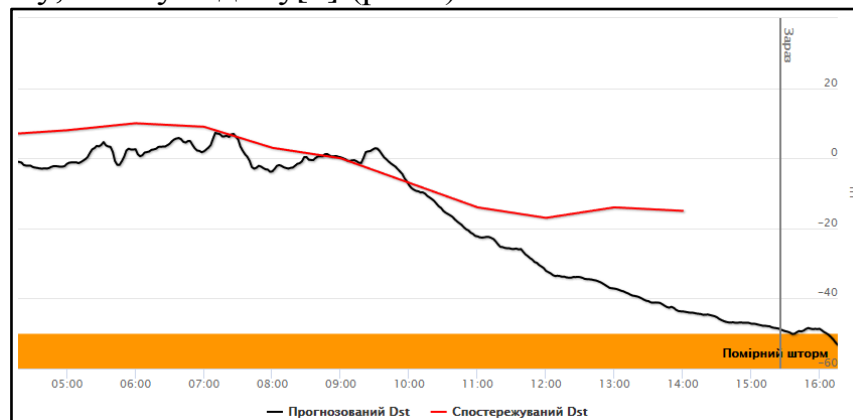


Рис. 1. Розбіжність прогнозу і реальних показників Dst індексу [4]

### **Актуальність показників з огляду на засоби ШНМ**

Для дослідження ефективності застосування засобів на базі ШНМ варто обрати показник, керуючись специфічними вимогами. Оптимальнішими будуть показники з більшою частотою звітування, що дозволить при прогнозуванні того самого часового проміжку точніше тренувати мережу. Показники з великою амплітудою нерегулярних в часі змін є небажаними, так як для навчання мережі поведінці таких показників треба досліджувати значно більший набір даних, що значно збільшить витрати на навчання, а при недостатньому розмірі вибірки викликати помилки в результатах (подібним показником є EUV). Феномени, поведінка котрих спостерігається на великому часовому проміжку також є недоречними для початкових досліджень через велику кількість даних що необхідно розглянути для навчання мережі, що ускладнить розробку, налаштування та навчання мережі. Індеси магнітосфери є хорошими кандидатами для подібних досліджень, так як мають частоту спостереження від однієї до кількох годин, демонструють характерну їм поведінку на порівняно невеликому проміжку часу, а власне магнітне поле Землі обмежує амплітуду коливань.



## **Застосування отриманих результатів**

Сформульовані у даній статі висновки можуть бути в подальшому використані для безпосередньої розробки відповідного методу передбачення окремих факторів космічної погоди з підвищеною точністю. Наразі інтенсивно розвивається комерційна діяльність зв'язана з космосом, прикладами цього є такі технології як “Starlink” та інші супутникові системи, та зародження космічного туризму[5].

## **Висновки**

Проведене дослідження факторів космічної погоди дозволило формалізувати характеристики, що сприяють кращій інтеграції засобів на базі штучних нейронних мереж для передбачення відповідних факторів. На основі цих факторів було відібрано кращі варіанти, серед яких найбільший потенціал в даному напрямку має Dst індекс. Існуючі дослідження демонструють, що подібним чином його можна прогнозувати принаймні на 5 годин вперед. Окрім того, нейронні мережі на основі білінійних моделей можуть значно покращити прогноз.

## **Література**

1. SPACE WEATHER PHENOMENA [Електронний ресурс] // SPACE WEATHER PREDICTION CENTER NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION – Режим доступу до ресурсу: <https://www.swpc.noaa.gov/phenomena>
2. Love J. The USGS Geomagnetism Program and Its Role in Space Weather Monitoring / J. Love, C. Finn. // Space Weather. – 2011. – Volume 9 I7.
3. Suigura M. On Dst index / M. Suigura, T. Kamei. // IAGA Bulletin №40,. – 1991. – С. 123.
4. Магнітометри [Електронний ресурс] // SpaceWeatherlive.com Real-time auroral and solar activity – Режим доступу до ресурсу: [https://www.spaceweatherlive.com/uk/avroralna-aktivnist/magnitometri.htmlDST\\_index](https://www.spaceweatherlive.com/uk/avroralna-aktivnist/magnitometri.htmlDST_index).
5. Commercial use of space [Електронний ресурс] // Wikipedia The free encyclopedia – Режим доступу до ресурсу: [https://en.wikipedia.org/wiki/Commercial\\_use\\_of\\_space](https://en.wikipedia.org/wiki/Commercial_use_of_space).

УДК 004.93

Професор, д.т.н. Яценко В.О., студент Ціпун В.Ю.

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

## КОНЦЕПЦІЯ ТУМАННИХ ОБЧИСЛЕНЬ ТА ПЕРЕВАГИ ЇЇ ВИКОРИСТАННЯ У ІНФОРМАЦІЙНИХ СИСТЕМАХ

### Abstract

Vitaly Yatsenko, prof.; Vladyslav Tsipun, student

*The concept of fog computing and the benefits of its use in information systems*

*This paper describes the usage of Fog computing concepts for improving performance of the information systems”.*

### Вступ

На даний момент хмарні обчислення широко використовуються у світі для вирішення різних завдань, таких як зберігання файлів, комунікація, споживання розважального контенту або наукові дослідження. Використання цього підходу надає величезні переваги, але також накладає деякі обмеження. Одне з найбільших обмежень - це затримка(латентність). Затримка запобігає використанню хмарних обчислень у тих областях, де критичною є реакція системи при виникненні певних подій.

### Термінологія

**ІоТ (Інтернет речей)** — концепція мережі, що складається із взаємозв'язаних фізичних пристроїв, які мають вбудовані датчики, а також програмне забезпечення, що дозволяє здійснювати передачу і обмін даними між фізичним світом і комп'ютерними системами в автоматичному режимі, за допомогою використання стандартних протоколів зв'язку. Окрім датчиків, мережа може мати виконавчі пристрої, вбудовані у фізичні об'єкти і пов'язані між собою через дротові чи бездротові мережі. Ці взаємопов'язані пристрої мають можливість зчитування та приведення в дію, функцію програмування та ідентифікації, а також дозволяють виключити необхідність участі людини, за рахунок використання інтелектуальних інтерфейсів.

**ДЦ (Дата-центр або Центр даних)** — приміщення в будівлі, будівля або група будівель, призначена для збирання, нагромадження й зберігання великих обсягів інформації та їх опрацювання

**Латентність** - затримка в електронній системі між запитом до системи та моментом, коли доступ завершується або повертаються запитувані дані.

### **Концепція туманних обчислень**

Починаючи з 2011 року виникла потреба розширити хмарні обчислення, щоб впоратися з великою кількістю IoT-пристроїв і величезними обсягами даних, що виникають при роботі чутливих до затримки додатків, що працюють у реальному часі. Одним із можливих рішень цієї проблеми стали туманні обчислення, які привернули увагу широкої аудиторії завдяки інтересу компанії Cisco, що розглядає їх як нову технологію, дозволяючу організувати додатковий рівень взаємодії між кінцевими пристроями та хмарними дата-центрами. 19 листопада 2015 року такі компанії як Cisco Systems, ARM Holdings, Dell, Intel, Microsoft та Принстонський університет заснували консорціум OpenFog для просування інтересів та розвитку туманних обчислень. 31 січня 2019 року консорціум OpenFog об'єднався з Industrial Internet Consortium.

Використання цього підходу дозволяє виконувати обробку частини даних на локальних пристроях, а не лише виключно у data-центрах. Туманні обчислення тісно пов'язані з IoT. В рамках даної концепції всі пристрої мережі діляться на три шари:

- шар кінцевих пристроїв(користувацькі)
- шар туману, що включає різного роду комунікаційне обладнання (комутатори, роутери, шлюзи)
- шар хмари, розміщений, як правило, в межах одного data-центру.

Припускається, що прилади туманного шару, а також прилади що оснащені обчислювальними потужностями та ресурсами для зберігання даних, стають продуктивнішими зі зменшенням відстані до «хмари».

Тоді при зміщенні обчислювальних задач, що займаються обробкою даних, що надходять до краю мережі, відбувається наступне:

- Знижується навантаження на пристрої хмарного шару
- Знижується латентність системи
- Суттєво знижується навантаження на мережу

Також з'являється можливість розвантаження кінцевих пристроїв шляхом переміщення обчислень в туманний шар, а також можливість розвантаження хмарних пристроїв у той самий спосіб.

### **Переваги туманних обчислень**

По суті розробка туманних обчислень надає організаціям більше можливостей для обробки даних. У деяких випадках необхідно обробити дані якнайшвидше - наприклад, у виробництві, коли підключені машини повинні негайно зреагувати на інцидент.

Туманні обчислення можуть створювати мережеві з'єднання з низькою затримкою між пристроями та кінцевими точками. Ця структура, у свою чергу, зменшує потрібну пропускну здатність – це краще, ніж якщо дані потрібно відправити назад у data-центр або хмару для обробки.

Туманні обчислення також використовують, коли пропускну спроможність недостатня для надсилання даних, тому вони повинні оброблятися поблизу місця, де були створені. Як додаткову перевагу, користувачі можуть розміщувати функції безпеки в туманну мережу, від сегментування мережевого трафіку до віртуальних брандмауерів.

### **Використання туманних обчислень**

Використання туманних обчислень все ще новинку, проте є безліч випадків, коли без них не обійтися.

Зв'язок між автомобілями: поява автономних автомобілів лише збільшить створення великої кількості даних. Наявність безпілотників вимагає локального аналізу певних даних у реальному часі, таких як навколишнє середовище, умови для водіння та напрямок руху. Деякі дані необхідно надсилати виробнику, щоб допомагати у покращенні технічного обслуговування транспортних засобів або відстежувати використання авто. Туманні обчислення дозволять забезпечити зв'язок усіх джерел даних – і автомобіль, і виробника.

Розумні міста та інтелектуальні мережі: Подібно до автономних автомобілів, комунальні системи все частіше використовують дані в режимі реального часу для більш ефективної роботи. Іноді ці дані знаходяться у віддалених областях, тому туманні обчислення таких даних є ідеальним рішенням.

Аналіз у режимі реального часу: Різні варіації використання вимагають аналізу у реальному часі – від виробничих систем, які мають блискавично реагувати на події, до фінансових установ, які використовують дані в режимі реального часу для інформування про комерційні рішення або моніторинг за шахрайством. Розгортання туманних обчислень дозволить спростити передачу даних між місцем їх створення та кінцевою точкою.

## **Література**

1. Bonomi F. Fog computing and its role in the internet of things // Proceedings of the first edition of the MCC workshop on Mobile cloud computing. , 2012. C. 13–16.
2. Alam M. G. R., Tun Y. K., Hong C. S. Multi-agent and reinforcement learning based code offloading in mobile fog // 2016 International Conference on Information Networking (ICOIN), 2016. IEEE, 2016. P. 285–290.
3. Fielding R. T., Taylor R. N. Principled design of the modern Web architecture // ACM Transactions on Internet Technology (TOIT). 2002. Vol. 2, No 2. P. 115–150.

## ЗМІСТ

1.	Брусенцов Ю.О., Сулема Є.С. АЛГОРИТМІЧНО-ПРОГРАМНИЙ МЕТОД ПРОЦЕДУРНОЇ ГЕНЕРАЦІЇ ВОКСЕЛЬНОГО ЛАНДШАФТУ ІЗ 3D-ТАЙЛІВ	4
2.	Заболотня Т.М., Корунська А.М. МЕТОД АВТОМАТИЗОВАНОЇ ВЕРИФІКАЦІЇ ПРОГРАМНОГО КОДУ СМАРТ-КОНТРАКТІВ	9
3.	Заболотня Т.М., Лук'янець М.О. АНАЛІЗ МЕТОДІВ ПОПЕРЕДНЬОГО ОБРОБЛЕННЯ ТЕКСТОВИХ ДАНИХ ЩОДО ЇХ ВИКОРИСТАННЯ У ВИЗНАЧЕННІ ТОНАЛЬНОСТІ УКРАЇНОМОВНИХ ТЕКСТІВ	14
4.	Заболотня Т.М., Мединський М.С. ПІДСИСТЕМА ІНТЕЛЕКТУАЛЬНОГО РЕДАКТОРА ПРИРОДНОМОВНИХ ТЕКСТІВ	19
5.	Заболотня Т.М., Свиначук М.В. АЛГОРИТМІЧНО-ПРОГРАМНИЙ МЕТОД ПОШУКУ ЗАПОЗИЧЕНЬ У ПРОГРАМНОМУ КОДІ	24
6.	Згуровський Я.Ю., Сулема Є.С. ПРОГРАМНА БІБЛІОТЕКА ДЛЯ РЕАЛІЗАЦІЇ ПРОЦЕДУР ОБРОБЛЕННЯ ТЕМПОРАЛЬНИХ МУЛЬТИМОДАЛЬНИХ ДАНИХ	30
7.	Ільїн М.О., Заболотня Т.М. МЕТОД ВИДІЛЕННЯ ВІДПОВІДІ ІЗ КОРИСТУВАЦЬКОЇ БАЗИ ЗНАНЬ ПИТАЛЬНО-ВІДПОВІДНОЇ СИСТЕМИ ВІДДІЛУ ТЕХНІЧНОЇ ПІДТРИМКИ	36
8.	Казимиров Д.М., Заболотня Т.М. УЗАГАЛЬНЕНИЙ МЕТОД АВТОМАТИЗОВАНОГО АНОТУВАННЯ ТЕКСТОВИХ ДАНИХ З ВИКОРИСТАННЯМ НЕЙРОННОЇ МЕРЕЖІ VERT	42
9.	Казимирова О.О., Заболотня Т.М. ГІБРИДНИЙ ПІДХІД ДО АВТОМАТИЗОВАНОГО РЕФЕРУВАННЯ ТЕКСТОВИХ ДАНИХ	47
10.	Легеза В.П., Атаманюк О.В., Нецадим О.М. АМПЛІТУДНО-ЧАСТОТНА ХАРАКТЕРИСТИКА ВІБРОЗАХИСНОЇ СИСТЕМИ ТА МЕТОД ВИЗНАЧЕННЯ ОПТИМАЛЬНИХ ПАРАМЕТРІВ ЇЇ КУЛЬОВОГО ГАСНИКА	53
11.	Oleshchenko L.M., Burchak Pavlo ANALYSIS AND OPTIMIZATION OF METHODS FOR STORING AND PROCESSING DATA OF WEB APPLICATIONS	59

12.	Oleshchenko L.M., Sadrytskyi Serhii MATHEMATICAL METHODS FOR CLUSTERING SOFTWARE USERS BY GIVEN ATTRIBUTES	64
13.	Oleshchenko L.M., Zhang Yingzhi SOFTWARE ANALYSIS METHOD OF STATISTICAL DATA ABOUT THE TERRITORIAL LIVING STANDARD OF PEOPLE	69
14.	Mykola Onai, Ahmad Husein METHODS OF PERFORMING OPERATIONS IN ELLIPTIC CURVE OVER $GF(P)$	75
15.	Онай М.В., Квітка О.В. МОДИФІКОВАНИЙ МЕТОД ГЕНЕРАЦІЇ КЛЮЧІВ ДЛЯ АЛГОРИТМУ АСИМЕТРИЧНОГО ШИФРУВАННЯ RSA	80
16.	Mykola Onai, Haoling Liu IMPROVEMENT OF KUBERNETES RESOURCE SCHEDULING ALGORITHM	85
17.	Сулема Є.С., Дичка А. І. АНАЛІЗ КОРЕКТУВАЛЬНИХ ВЛАСТИВОСТЕЙ МНОГОЗНАЧНИХ КОДІВ ХЕМІНГА	89
18.	Телефус І.А., Сулема Є.С. ОПТИЧНЕ РОЗПІЗНАВАННЯ РУКОПИСНИХ ТЕКСТІВ УКРАЇНСЬКОЮ МОВОЮ З ВИКОРИСТАННЯМ НЕЙРОННИХ МЕРЕЖ	98
19.	Tereikovskiy Ihor, Tianbang Wang SELECTION OF PHONEMES IN THE VOICE SIGNAL USING NEURAL NETWORKS	103
20.	Tereikovskiy Ihor, Isik Oguzhan SPAM RECOGNITION SYSTEM	108
21.	Хіцко Я.В., Мірошник В.І. АЛГОРИТМІЧНО-ПРОГРАМНИЙ МЕТОД ГЕНЕРАЦІЇ ХМАР ТЕГІВ БІРЖОВИХ НОВИН	114
22.	Хіцко Я.В., Нечипоренко В.О.. МЕТОД ДЛЯ АНАЛІЗУ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ ВІДГУКІВ КОРИСТУВАЧІВ	120
23.	Струцинський О.А., Щербина В.О. АЛГОРИТМ ЗБЕРІГАННЯ ТА АНАЛІЗУ КОРИСТУВАЦЬКИХ МЕТРИК МОНІТОРИНГОВИМИ СИСТЕМАМИ	125
24.	Бай Ю.П., Ориняк І.В., Городнюк С.О. ЗАСТОСУВАННЯ КОМБІНАЦІЙ ТОЧНИХ АНАЛІТИЧНИХ РІШЕНЬ ТА МЕТОДУ КОЛОКАЦІЙ ДЛЯ ВИЗНАЧЕННЯ ВЛАСНИХ ЧАСТОТ ПЛАСТИН	129

25.	Вовк Л.Б., Гридько О.С. СИСТЕМА АНАЛІЗУ ТЕНДЕНЦІЙ КРИПТОВАЛЮТНОГО РИНКУ ЗА ДОПОМОГОЮ ЧАСОВИХ РЯДІВ	135
26.	Вовк Л.Б., Мілевська О.О. МАТЕМАТИЧНІ МОДЕЛІ В ЕПІДЕМІОЛОГІЇ ДЛЯ ПРОГНОЗУВАННЯ COVID-19	140
27.	Ковальчук-Химюк Л.О., Юдін Г.Є. АНАЛІЗ МЕТОДІВ ВИЗНАЧЕННЯ ЕМОЦІЙНОГО ЗАБАРВЛЕННЯ ПІСЕНЬ	145
28.	Лось В.М., Без'язична К.Ю. МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ПРОГОЗУВАННЯ ЗАХВОРЮВАНOSTІ НА КІР З ВИКОРИСТАННЯМ МЕТОДУ ВАЛЬДА	151
29.	Маслянко П.П., Луцик М.С. МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ CRM СИСТЕМИ ДЛЯ ПІДПРИЄМСТВ ТЕХНОЛОГІЧНО- БУДІВЕЛЬНОГО БІЗНЕСУ	156
30.	Маслянко П.П., Маляренко І.С., Ковальчук-Химюк Л.О. МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ІДЕНТИФІКАЦІЇ ОЦІНОЧНИХ СУДЖЕНЬ У ТЕКСТАХ	166
31.	Норкін В.І., Козирев А.Ю. ДОСЛІДЖЕННЯ ЗБІЖНОСТІ СТОХАСТИЧНИХ ГРАДІЄНТНИХ ТА КІНЦЕВЕ-РІЗНИЦЕВИХ МЕТОДІВ ОПТИМІЗАЦІЇ	175
32.	Олефір О.С., Андрійчук В.В. ПРОЕКТУВАННЯ МОДЕЛІ УНІВЕРСАЛЬНОГО ЗАГАЛЬНОПЛАНЕТАРНОГО ЧАСУ	184
33.	Ориняк І.В., Кольцов Д.Р. ЗАСТОСУВАННЯ ТЕОРІЇ БАЛОК ДЛЯ ПОБУДОВИ ДВІЧІ ДИФЕРЕНЦІЙОВАНИХ ПЛОСКИХ КРИВИХ ПО НЕТОЧНИМ ДИСКРЕТНИМ ДАНИМ	188
34.	Ориняк І.В., Піць В.В. ПОРІВНЯННЯ МЕТОДІВ ГАЛЬОРКІНА, НАВ'Є ТА КІНЦЕВИХ РІЗНИЦЬ НА ПРИКЛАДІ СТАТИЧНОГО ДЕФОРМУВАННЯ ПРЯМОКУТНОЇ ПЛАСТИНИ	197
35.	Пашко А.О., Канєвський В.В. АЛГОРИТМ ГЕНЕРАЦІЇ ДВІЙКОВИХ ВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ В ЗАДАЧАХ ЗАХИСТУ ІНФОРМАЦІЇ	204



36.	Сирота С.В., Бандурін В.Ю. ІНТЕРАКТИВНИЙ СЕРВІС ПОРІВНЯЛЬНОГО АНАЛІЗУ УНІВЕРСИТЕТІВ УКРАЇНИ НА ОСНОВІ ДАНИХ ВСТУПНОЇ КАМΠΑНІЇ	212
37.	Сирота С.В., Фельдман М.Г. ОГЛЯД МЕТОДІВ ОБРОБКИ ТЕКСТІВ ПРИРОДНОЇ МОВИ ДЛЯ АВТОМАТИЗОВАНОГО ОЦІНЮВАННЯ СТИЛІСТИЧНОЇ ЯКОСТІ ЕСЕ	217
38.	Соловійов С.О., Копичко С.М., Рябокониць З.А. СТОХАСТИЧНЕ МОДЕЛЮВАННЯ СТРАТЕГІЙ КОНТРОЛЮ ПАНДЕМІЇ COVID-19	222
39.	Третиник В. В., Корінь Й. В. МОДЕЛІ СТАТИСТИЧНОГО АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ПОВЕДІНКИ РИНКУ НЕРУХОМОСТІ	227
40.	Чертов О. Р., Братун А. Ю. СИСТЕМА ВИЯВЛЕННЯ ТА РОЗПІЗНАВАННЯ НОМЕРНИХ ЗНАКІВ ВАГОНІВ ПОТЯГУ НА ВІДЕОПОТОЦІ ЗА ДОПОМОГОЮ НЕЙРОННИХ МЕРЕЖ	234
41.	Чертов О. Р., Мілев М. С. МЕТОД ВИЯВЛЕННЯ ТИПОВИХ ПРЕДСТАВНИКІВ КЛАСУ В РОЗМІЧЕНИХ ДАНИХ	241
42.	Боярінова Ю.Є., Гнатенко В.Д. СПОСІБ ШТРИХОВОГО КОДУВАННЯ ТА ДЕКОДУВАННЯ ІНФОРМАЦІЇ	247
43.	Боярінова Ю. Є., Самофалов А. В. СПОСІБ УЩІЛЬНЕННЯ ДАНИХ БЕЗ ВТРАТ	251
44.	Боярінова Ю.Є., Штогрин П.П. АВТОМАТИЧНЕ УПРАВЛІННЯ КВАДРОКОПТЕРОМ З ПОШУКОМ ОПТИМАЛЬНОГО МАРШРУТУ	256
45.	Боярінова Ю. Є. Шуть М. В. ВИКОРИСТАННЯ ФРЕЙМВОРКУ ReactJS НА ПРИКЛАДІ РОЗРОБКИ ІНФОРМАЦІЙНО-ДОВІДКОВОЇ СИСТЕМИ ЛІКАРЯ	261
46.	Війтенко А.М., Павловський В.І. КОМП'ЮТЕРНА ERP-СИСТЕМА ОСУЧАСНЕНИХ РОЗРАХУНКІВ ТЕПЛОПОСТАЧАННЯ М. ОДЕСИ	265
47.	Доляк Д.І., Дробязко І.П. МЕТОДИ ОБФУСКАЦІЇ ПРОГРАМНОГО КОДУ НА ОСНОВІ СКЛАДЕНИХ ФУНКЦІЙ ТА АФІННИХ ШИФРІВ	270

48.	Дробязко І.П., Дученко О.С. МОДУЛЬ АВТОМАТИЧНОГО ВИБОРУ АЛГОРИТМУ ПЕРЕВПОРЯДКУВАННЯ ЕЛЕМЕНТІВ РОЗРІДЖЕНОЇ МАТРИЦІ ДЛЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ РОЗВ'ЯЗУВАННЯ СЛАР	274
49.	Зайцев В.Г., Поліщук А.В. МОДИФІКАЦІЯ АЛГОРИТМУ RD2 ПЛАНУВАННЯ ЗАДАЧ У СИСТЕМАХ РЕАЛЬНОГО ЧАСУ	279
50.	Клятченко Я.М., Ясенко Л.С. ПОРІВНЯЛЬНИЙ АНАЛІЗ ШВИДКОСТІ ВИКОНАННЯ АЛГОРИТМІВ НЕЙРОННИХ МЕРЕЖ ЗАЛЕЖНО ВІД АПАРАТНИХ ЗАСОБІВ	284
51.	Марченко О.І., Вовчок О.В. СЕМАНТИЧНИЙ СПОСІБ ПОСДНАННЯ ВЕРСІЙ ПРОГРАМНОГО КОДУ ПРИ ОДНОЧАСНІЙ РОЗРОБЦІ ПРОГРАМ ДЛЯ ПІДМНОЖИНИ МОВИ PYTHON	290
52.	Марченко О.І., Кравчук В.В. СПОСІБ ВИЗНАЧЕННЯ НЕЗАЛЕЖНИХ КОНСТРУКЦІЙ REDEFINES	294
53.	Марченко О.І., Курдус А.О. RANDOM WALKS АЛГОРИТМИ	298
54.	Марченко О. І., Чорний Є.Г. СПОСІБ ОЦІНЮВАННЯ ОБЧИСЛЮВАЛЬНОЇ СКЛАДНОСТІ АЛГОРИТМІВ ЗА КРИТЕРІЄМ ЧАСУ ВИКОНАННЯ АСЕМБЛЕРНИХ ІНСТРУКЦІЙ	302
55.	Морозов К.В., Романкевич О.М., Білих О.С. ТЕСТОВИЙ ЕКСПЕРИМЕНТ ПО ВИЗНАЧЕННЮ СТАНУ ПРОЦЕСОРІВ БАГАТОПРОЦЕСОРНОЇ СИСТЕМИ	306
56.	Наливайчук М.В., Шостак М.С. АНАЛІЗ СТАТИСТИЧНИХ ДАНИХ ІНТЕРНЕТ РЕСУРСІВ ІЗ ВИКОРИСТАННЯМ АЛГОРИТМУ МАШИННОГО НАВЧАННЯ LINEAR REGRESSION	311
57.	Несук О.О., Потапова К.Р., Кучмій О.О. СПОСІБ ПРОГРАМНО ПРИСКОРЕНОГО ВІЗУАЛЬНОГО ПОДАННЯ ІНФОРМАЦІЇ	316
58.	Орлова М.М., Баглай І.Ю. ЕФЕКТИВНИЙ АЛГОРИТМ ДЛЯ ЗАДАЧІ УПАКОВКИ 3D- ПРЯМОКУТНИХ БЛОКІВ	321

59.	Орлова М. М., Коркішко А. О. АНАЛІЗ ТЕХНОЛОГІЇ БЛОКЧЕЙН, З ТОЧКИ ЗОРУ БЕЗПЕЧНОГО ПРОВЕДЕННЯ ЦИФРОВИХ ОПЕРАЦІЙ	326
60.	Орлова М. М., Сміюн І. В. АНАЛІЗ ВИДІВ ТЕХНОЛОГІЇ BLOKCHAIN ТА ЇХ ЗАСТОСУВАННЯ У СФЕРІ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ	330
61.	Павловський В.І., Бойко В.В. ОПИС ОСНОВНИХ СПОСОБІВ ДЕТЕКЦІЇ ПЛАГІАРИЗМУ	335
62.	Павловський В.І., Рекеда В.В. ВЕБ-СЕРВІС ПІДВИЩЕННЯ РОЗДІЛЬНОЇ ЗДАТНОСТІ ЗОБРАЖЕНЬ З ВИКОРИСТАННЯМ SR-АЛГОРИТМІВ	341
63.	Павловський В.І., Савосько О.М. ВИЯВЛЕННЯ ШКІДЛИВОГО ТРАФІКУ ЗА ВИКОРИСТАННЯМ ГЛИБИННОГО НАВЧАННЯ	346
64.	Петрашенко А. В. Коровій О.С. СПОСІБ РОЗПІЗНАВАННЯ ТОНАЛЬНОГО ЗАБАРВЛЕННЯ ТЕКСТІВ НА ОСНОВІ ПРОЦЕСУ "ДИСТИЛЯЦІЇ ЗНАНЬ"	351
65.	Петрашенко А.В., Рябоконь Т.О. ПРОГРАМНІ ЗАСОБИ ФОРМУВАННЯ ТА ОБРОБКИ БАЗИ ДАНИХ СЛОВОСПОЛУЧЕНЬ УКРАЇНСЬКОЇ МОВИ	355
66.	Петрашенко А.В., Бідяк М.А. ПРОЦЕС ВИЯВЛЕННЯ ТА ЗАПОБІГАННЯ ІНФОРМАЦІЙНИМ АТАКАМ ЗА ДОПОМОГОЮ МАШИННОГО НАВЧАННЯ	360
67.	Петрашенко А.П., Дідус А.В. АНАЛІЗ МЕТОДІВ СТВОРЕННЯ НАБОРУ ТЕКСТОВИХ ДАНИХ ІЗ ВІДКРИТИХ ДЖЕРЕЛ ДЛЯ ПОДАЛЬШОЇ ОБРОБКИ	364
68.	Потапова К.Р., Празднікова М. О. МОДИФІКАЦІЯ АЛГОРИТМУ КАННІ ДЛЯ ОБЧИСЛЕННЯ МЕЖ ОБ'ЄКТІВ ЗОБРАЖЕНЬ	368
69.	Стадниченко А.В., Дробязко І.П. ПОРІВНЯЛЬНИЙ АНАЛІЗ СПОСОБІВ З'ЄДНАННЯ ПРИСТРОЇВ У СИСТЕМІ РОЗУМНИЙ ДІМ	373
70.	Тарасенко-Клятченко О. В. , Олійник О. В. ВИКОРИСТАННЯ OLAP-ТЕХНОЛОГІЙ ДЛЯ БАГАТОВИМІРНОГО АНАЛІЗУ ДАНИХ	378
71.	Тарасенко-Клятченко О. В. , Бровдій Є. Ю. КЛІЄНТООРІЄНТОВАНІ СЕРВІСИ ДЛЯ РЕКЛАМИ НА ОСНОВІ OPENRTV SLICKHOUSE ТЕХНОЛОГІЇ	383

72.	Терейковський І.А., Кравчук О.В. АНАЛІЗ ВРАЗЛИВОСТЕЙ ПРОГРАМНОГО РІВНЯ ОПЕРАЦІЙНОЇ СИСТЕМИ ANDROID	388
73.	Терейковський І.А., Остапчук С.І. АНАЛІЗ ПРОГРАМНИХ КОМПЛЕКСІВ ДЛЯ ПЕРЕКЛАДУ ЗВУКОВИХ ДОРІЖОК ВІДЕОФАЙЛІВ	393
74.	Терейковський І.А., Парфіненко Ю.М. АЛГОРИТМ ОБРОБКИ ВІДЕОПОТОКУ	397
75.	Тесленко О.К., Перегудов М.С. АНАЛІЗ ПІДХОДІВ ТА МЕТОДІВ НТРОПОГЕННОГО ВПЛИВУ НА ПРОГРАМНІ КОМПЛЕКСИ	401
76.	Тесленко О.К., Ходоровський А.П. ШВИДКИЙ СПОСІБ ВИЗНАЧЕННЯ КЛІКИ НЕРОЗДІЛЕНОГО ЗАВАДОСТІЙКОГО КОДУ	405
77.	Штефанович Г.М., Коляда К.В. СПОСІБ ГЕНЕРАЦІЇ ТА ВИКЛИКУ SQL-ЗАПИТІВ НА ОСНОВІ REST API	410
78.	Яценко В.І., Без'язичний М.В. ІСНУЮЧІ КВАНТОВІ АЛГОРИТМИ ТА РОЗВ'ЯЗУВАННЯ ЗАДАЧ КОМБІНАТОРИКИ ЗА ДОПОМОГОЮ КВАНТОВИХ КОМП'ЮТЕРІВ	415
79.	Яценко В.О., Герасимчук Д.М. ОГЛЯД ТЕХНОЛОГІЙ ОПТИМІЗАЦІЇ ДОКУМЕНТООБІГУ НА ПІДПРИЄМСТВІ	420
80.	Яценко В.О., Калитка Є.О. АНАЛІЗ ТЕХНОЛОГІЇ «ІНТЕРНЕТУ РЕЧЕЙ» (INTERNET OF THINGS, IOT), ЇЇ РОЗВИТОК, СТРУКТУРА ТА ПОДАЛЬШІ ПЕРСПЕКТИВИ	424
81.	Яценко В.О., Троян Б.В. АНАЛІЗ ФЕНОМЕНІВ КОСМІЧНОЇ ПОГОДИ ТА ПЕРСПЕКТИВИ ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ ЇХ ДОСЛІДЖЕННЯ ТА ПРОГНОЗУВАННЯ	428
82.	Сирота С.В., Ковтун А.О. АВТОМАТИЗОВАНА СИСТЕМА ГЕНЕРУВАННЯ ТЕСТОВИХ ЗАПИТАНЬ НА ОСНОВІ МОДЕЛЕЙ ОБРОБКИ ПРИРОДНЬОЇ МОВИ	433
83.	Третиник В.В., Трішина А.О. МОДЕЛЬ ARIMA ДЛЯ ПРОГНОЗУВАННЯ ТРАФІКУ ВЕБ- СТОРІНКИ	438
84.	Tereikovskiy Ihor, Abdulwahid Ibrahim PREDICTING WEB SERVER LOAD	445

85. Потапова К.Р., Климчук І.О. 450  
ОСОБЛИВОСТІ ОРГАНІЗАЦІЇ ЗВУКОВОГО ІНТЕРФЕЙСУ  
ДЛЯ ЛЮДЕЙ З ДЕФЕКТАМИ МОВЛЕННЯ
86. Яценко В.О., Ціпун В.Ю. 455  
КОНЦЕПЦІЯ ТУМАННИХ ОБЧИСЛЕНЬ ТА ПЕРЕВАГИ ЇЇ  
ВИКОРИСТАННЯ У ІНФОРМАЦІЙНИХ СИСТЕМАХ

ДЛЯ НОТАТОК

ДЛЯ НОТАТОК

Наукове видання

**ПРИКЛАДНА МАТЕМАТИКА ТА КОМП'ЮТИНГ  
ПМК-2021**

Збірник тез доповідей

*Чотирнадцятої 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. (044) 234-15-86, 234-95-23 (факс).  
Свідоцтво ДК № 221 від 16.10.2000 р.

Підп. до друку 25.11.2021 . Формат 60x84<sup>1</sup>/<sub>16</sub>. Папір офс. Гарнітура Times.  
Спосіб друку – різнограф. Ум. друк. арк. 21,3. Обл.-вид. арк. 19. Зам. № .  
Наклад 110 пр.

---

«КПІ ім. Ігоря Сікорського» Видавництво «Політехніка»  
03056, київ, вул. Політехнічна, 14, корпус 15  
тел./факс (044) 204-81-78