



МАТЕРІАЛИ  
ПЕРШОЇ ВСЕУКРАЇНСЬКОЇ  
НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ  
МОЛОДИХ ВЧЕНИХ ТА СТУДЕНТІВ  
«ІНФОРМАЦІЙНІ СИСТЕМИ  
ТА ТЕХНОЛОГІЇ УПРАВЛІННЯ»



29-30 листопада  
Україна, Київ

## Оргкомітет конференції

**Голова організаційного комітету:** О.А. Павлов – завідувач кафедри АСОІУ НТУУ «КПІ ім. Ігоря Сікорського», д.т.н., професор.

### Члени організаційного комітету:

П.І.П.	Посада
О.А. Павлов	Завідувач кафедри АСОІУ, професор
І.П. Муха	Доцент кафедри АСОІУ
О.В. Гавриленко	Доцент кафедри АСОІУ
І.В. Баклан	Доцент кафедри АСОІУ
К.І. Ліщук	Доцент кафедри АСОІУ
Ю.О. Олійник	Старший викладач кафедри АСОІУ

### Члени програмного комітету:

П.І.П.	Посада
В.М. Томашевський	Професор кафедри АСОІУ
І.В. Стеценко	Професор кафедри АСОІУ
В.І. Литвиненко	Професор ХНТУ
Г.В. Рудакова	Професор ХНТУ
О.Г. Жданова	Доцент кафедри АСОІУ
Т.В. Ковалюк	Доцент кафедри АСОІУ
Е.В. Жаріков	Доцент кафедри АСОІУ
Ю.М. Селін	Доцент кафедри АСОІУ
О.С. Жураковська	Доцент кафедри АСОІУ
Т.О. Тєлишева	Доцент кафедри АСОІУ
В.Д. Попенко	Доцент кафедри АСОІУ
М.О. Сперкач	Старший викладач кафедри АСОІУ
М.І. Цюцюра	Старший викладач кафедри АСОІУ

**Секретар конференції:** С. П. Якуніна, інженер кафедри АСОІУ

І Всеукраїнська науково-практична конференція молодих вчених та студентів «Інформаційні системи та технології управління – ІСТУ-2018». Секція кафедри автоматизованих систем обробки інформації і управління. Матеріали конференції. – Київ. – 2018.

29-30 грудня 2018р. – 231 с.

У збірник включені тези доповідей, які були представлені на конференції «Інформаційні системи та технології управління – ІСТУ-2018» в секції кафедри автоматизованих систем обробки інформації і управління. В доповідях розглянуті наукові та методичні питання щодо сучасних аспектів інформатики та обчислювальної техніки.

Редакційна колегія:

Гавриленко О.В., к.ф.-м.н., кафедра АСОІУ НТУУ «КПІ ім. Ігоря Сікорського»,  
Муравйова І. М., старший лаборант кафедри АСОІУ

Дизайн титульної сторінки: Майєр З.О.

## Зміст

1.	<i>ЧЕРЕДНІЧЕНКО В. О., ЗАЛЕВСЬКА О. В.</i>	ТРИ – ВИМІРНІ КЛІТИННІ АВТОМАТИ В ГЕНЕРАЦІЇ ОБ’ЄКТІВ КОМП’ЮТЕРНОЇ ГРАФІКИ	6
2.	<i>БЕЛЯЄВА А.А. ТЄЛИШЕВА Т.О.</i>	ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АВТОМАТИЧНОГО ТЕСТУВАННЯ ВЕБ- ЗАСТОСУВАНЬ	10
3.	<i>ГОРОБЮК В.П.</i>	ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ КЛАСИФІКАЦІЇ КОРИСТУВАЧІВ НА ОСНОВІ ПОВЕДІНКОВОЇ МОДЕЛІ	12
4.	<i>КАСЯНЧИК Д.О.</i>	ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ КЛАСТЕРИЗАЦІЇ КОРИСТУВАЧІВ ДЛЯ ПІДГОТОВКИ ЦІЛЬОВИХ ПРОПОЗИЦІЙ	16
5.	<i>КОВАЛЕНКО А.О. ПОПЕНКО В.Д.</i>	ІНФОРМАЦІЙНА СИСТЕМА ФОРМУВАННЯ РЕКЛАМНОГО ПАКЕТУ З УРАХУВАННЯМ ІНДИВІДУАЛЬНИХ ХАРАКТЕРИСТИК КОРИСТУВАЧА	20
6.	<i>ПЄНСКОЙ В. В. ФІНОГЕНОВ О.Д.</i>	ОГЛЯД ЗМАГАНЬ ТА СИМУЛЯТОРІВ ФУТБОЛУ РОБОТІВ	23
7.	<i>БУЛГАР М.М., ЖАРИКОВ Е.В.</i>	СПОСІБ КЛАСТЕРИЗАЦІЇ КОРИСТУВАЧІВ СОЦІАЛЬНОЇ МЕРЕЖІ TWITTER	28
8.	<i>РОМАНЧУК В. М., ОЛІЙНИК Ю.О.</i>	ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ДЛЯ ПОШУКУ АНОМАЛІЙ У ВІДЕОПОТОКАХ	32
9.	<i>ГУНАВАРДАНА Р.С.Д. СЕЛІН Ю.М.</i>	СУЧАСНІ ПІДХОДИ ДО ВИРІШЕННЯ ЗАДАЧ РОЗМІЩЕННЯ НА ПРИКЛАДІ ОПТИМІЗАЦІЇ РОЗКЛАДУ РЕКЛАМНОГО КОНТЕНТУ	35
10.	<i>ДЕМИДЕНКО М. О.</i>	ПРОГРАМНО-АППАРАТНИЙ КОМПЛЕКС АНАЛИЗА УЯЗВИМОСТЕЙ УСТРОЙСТВ, НАХОДЯЩИХСЯ В ОДНОЇ ПОДСЕТИ С ДОСТУПОМ К СЕТИ	39
11.	<i>ДОВГАЛЬ Д.О. КОВАЛЮК Т.В.</i>	КОГНІТИВНИЙ ПІДХІД В МОДЕЛЮВАННІ І УПРАВЛІННІ	45
12.	<i>КАПОРІН Р. М. СПЕРКАЧ М. О.</i>	ПРОГРАМНО-ІНФОРМАЦІЙНИЙ КОМПЛЕКС ДЛЯ МОНІТОРИНГУ ТА КЕРУВАННЯ ГРОМАДСЬКИМ ТРАНСПОРТОМ	49
13.	<i>КОБИЛИНСЬКИЙ Д.А., ОЛІЙНИК Ю.О.</i>	СЕНТИМЕНТАЛЬНИЙ АНАЛІЗ КОРИСТУВАЦЬКИХ ТЕКСТОВИХ ДАНИХ У ВІДГУКАХ ПРО ТОВАРИ	55
14.	<i>КОГУЛЬКО О.С. ПОПЕНКО В.Д.</i>	НАДАННЯ РЕКОМЕНДАЦІЙ ЕЛЕМЕНТІВ НА ОСНОВІ ГІБРИДНОЇ ФІЛЬТРАЦІЇ	61
15.	<i>КОЛІЩАК Б.В., БАКЛАН І.В.</i>	ІДЕНТИФІКАЦІЯ ТА АУТЕНТИФІКАЦІЯ КОРИСТУВАЧА НА ОСНОВІ ДИНАМІЧНИХ БИОМЕТРИЧНИХ ХАРАКТЕРИСТИК З ВИКОРИСТАННЯМ АПАРАТУ ПРИХОВАНИХ МАРКОВСЬКИХ МОДЕЛЕЙ	66
16.	<i>МАМАЄНКО Є.О., ГАВРИЛЕНКО О.В.</i>	ДОСЛІДЖЕННЯ АЛГОРИТМІВ РОЗПІЗНАВАННЯ ЖЕСТІВ РУК ДЛЯ ВЗАЄМОДІЇ МІЖ ЛЮДЬМИ ТА МОБІЛЬНИМИ ПРИСТРОЯМИ	72
17.	<i>СКИДАН М. Т.</i>	ПОРІВНЯННЯ АЛГОРИТМІВ ДЛЯ ГЕНЕРАЦІЇ ЦИФРОВОГО ПІДПИСУ	76

18.	<i>ЧИЖ Ю.М., СТЕЦЮК В.З.</i>	ОПТИМІЗАЦІЯ ПРОЦЕСУ РЕЄСТРАЦІЇ ТА СПИСАННЯ ЛАБОРАТОРНОГО ВИТРАТНОГО МАТЕРІАЛУ І РЕАГЕНТІВ	79
19.	<i>КОРОБОВА Е.В., СПЕРКАЧ М.О.</i>	ІНФОРМАЦІЙНА СИСТЕМА ПІДТРИМКИ ДІЯЛЬНОСТІ РОБОТИ СТАНЦІЇ ШВИДКОЇ ДОПОМОГИ	83
20.	<i>КРАСНИКОВ С.О., ГУЛЯНИЦЬКИЙ Л.Ф.</i>	ПІДХІД ДО НАЛАШТУВАННЯ ПАРАМЕТРІВ АЛГОРИТМУ ІМІТАЦІЙНОГО ВІДПАЛУ ДЛЯ РОЗВ'ЯЗУВАННЯ ЗАДАЧ КОМБІНАТОРНОЇ ОПТИМІЗАЦІЇ	89
21.	<i>ЛАНЬКО В.В.</i>	ВИРІШЕННЯ ЗАДАЧІ АВТЕНТИФІКАЦІЇ КОРИСТУВАЧА ЗА РУХАМИ КУРСОРА МИШІ З ВИКОРИСТАННЯМ ЛІНГВІСТИЧНОГО ПІДХОДУ	95
22.	<i>МЕЗЕРЯ В. В.</i>	ІНФОРМАЦІЙНА ПІДСИСТЕМА ПОШУКУ ШЛЯХУ ДЛЯ МОДЕЛЮВАННЯ ПОВЕДІНКИ ВЕЛИКОЇ КІЛЬКОСТІ ЛЮДЕЙ В ОБМЕЖЕНОМУ ПРОСТОРІ	100
23.	<i>МЕХЕДОВ Є. В.</i>	ПРИНЦИПИ ПОБУДОВИ МАП РЕЛЬЄФУ ДЛЯ НАЗЕМНОЇ НАВІГАЦІЇ	103
24.	<i>НЕВМЕРЖИЦЬКИЙ Д. О., ОЛІЙНИК Ю.О.</i>	СИСТЕМА ОБЛІКУ РЕГУЛЬОВАНИХ ЛІКАРСЬКИХ ЗАСОБІВ ІЗ ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ ЛАНЦЮЖКІВ БЛОКІВ ТРАНЗАКЦІЙ	108
25.	<i>ОГОРОДНИКОВ Д., СТЕЦЕНКО І.В.</i>	БЕЗПЕКА ІНТЕРНЕТ РЕЧЕЙ	113
26.	<i>КОВАЛЕНКО А.С.</i>	ОГЛЯД БЕНЧМАРКІВ, ЇХ ЦІЛЕЙ ТА СКЛАД, АКТИВНІСТЬ З ВИКОРИСТАННЯ В ТММО І НАУЦІ В ЦІЛОМУ	116
27.	<i>ТКАЧЕНКО М.С., ЛІЩУК К.І.</i>	ДОСЛІДЖЕННЯ АРХІТЕКТУРИ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ РОЗПІЗНАВАННЯ 68 КЛЮЧОВИХ ТОЧОК ОБЛИЧЧЯ	122
28.	<i>КАХЕРСЬКИЙ О.І., ГАВРИЛЕНКО О.В.</i>	ДОСЛІДЖЕННЯ АЛГОРИТМУ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ АВТОМАТИЧНОЇ КЛАСИФІКАЦІЇ ОБЛИЧЬ	128
29.	<i>ПАВЛЕНКОВА Е. О., КОГАН А. В.</i>	СПОСОБ КОНСТРУИРОВАНИЯ ТРАФИКА В SDN	132
30.	<i>ПЕТРОВСЬКИЙ А.Д., ЦЮЦЮРА М.І.</i>	ДОСЛІДЖЕННЯ НАЙЕФЕКТИВНІШОГО АЛГОРИТМУ ДЛЯ СТИСНЕННЯ ЕЛЕКТРОННИХ ЗОБРАЖЕНЬ ДОКУМЕНТІВ	136
31.	<i>СКИДАН Д. О., ЖДАНОВА О. Г.</i>	КОМПЛЕКСНИЙ АСИМЕТРИЧНИЙ АЛГОРИТМ ШИФРУВАННЯ З ДИНАМІЧНИМ КЛЮЧЕМ	139
32.	<i>ДМИТРИШИН С. А.</i>	ПІДХІД ДО УПРАВЛІННЯ ЛОГАМИ КОМП'ЮТЕРНОЇ ІНФРАСТРУКТУРИ	144
33.	<i>СУХАНЮК М.В., СТЕЦЕНКО І.В.</i>	РОЗПІЗНАВАННЯ АВТОМОБІЛЬНИХ НОМЕРІВ У РЕЖИМІ РЕАЛЬНОГО ЧАСУ НА ОСНОВІ YOLO-ДЕТЕКТУВАННЯ	148
34.	<i>ТАЛЬКО Ю.С., СТЕЦЕНКО І.В.</i>	МЕТОДИ СТИСНЕННЯ МОДЕЛЕЙ В ГЛИБИННОМУ НАВЧАННІ НА ОСНОВІ МЕТОДУ УЧИТЕЛЯ-УЧНЯ	152

35.	<i>ТРОЦЮК А.Р., ПОПЕНКО В.Д.</i>	АНАЛІЗ СТРУКТУРИ СІВОЗМІН ЗА РІЗНИМИ ФОРМАМИ ЗЕМЕЛЬНОЇ ВЛАСНОСТІ В УКРАЇНІ	157
36.	<i>ХАНЬКО Г.В., ГАВРИЛЕНКО О. В., ОЛІЙНИК Ю. О.</i>	АНАЛІЗ ПІДХОДІВ ДО ВИЯВЛЕННЯ ЕЛЕМЕНТІВ АГІТАЦІЇ В ТЕКСТОВИХ ДАНИХ	163
37.	<i>ШИШКІН В.І., СТЕЦЕНКО І.В.</i>	АНАЛІЗ БРОКЕРУ ПОВІДОМЛЕНЬ ДЛЯ ВИРІШЕННЯ ЗАДАЧ РОЗУМНОГО ВІДЕОРЕЄСТРАТОРА	170
38.	<i>ЩЕРБАТЮК П.І.</i>	ПІДСИСТЕМА МОДЕЛЮВАННЯ ПОВЕДІНКИ ВЕЛИКОЇ КІЛЬКОСТІ ЛЮДЕЙ В ОБМЕЖЕНОМУ ПРОСТОРІ	174
39.	<i>ЯРОЦУК О.С., ОЛІЙНИК Ю.О.</i>	МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ВІЗУАЛІЗАЦІЇ ТА ПІДГОТОВКИ СТРУКТУР ДАНИХ НА КЛІЄНТСЬКОМУ ІНТЕРФЕЙСІ З ВИКОРИСТАННЯ ТЕХНОЛОГІЇ JAVASCRIPT	178
40.	<i>ЯЦУК С. М., СТЕЦЕНКО І. В.</i>	АНАЛІЗ ДОРОЖНЬОГО РУХУ З ВИКОРИСТАННЯМ UNITY 3D	181
41.	<i>ЛИТВАК Р. Б. ЖУРАКОВСЬКА О.С.</i>	ВИБІР АЛГОРИТМІЧНОГО ЗАБЕЗПЕЧЕННЯ ПІДТРИМКИ ПРОЦЕСУ СЕГМЕНТАЦІЇ КОРИСТУВАЧІВ СИСТЕМИ ТА ПОЗИЦІЮВАННЯ ТОВАРІВ НА ПРИКЛАДІ КНИЖКОВОГО РИНКУ	186
42.	<i>ШОЛОМ І.Д. МУХА І.П.</i>	ВЕБ-ЗАСТОСУВАННЯ ДЛЯ ОНЛАЙН-ОБРОБКИ ФОТОГРАФІЙ	192
43.	<i>СЯСЬКИЙ Д.П. ЖДАНОВА О.Г.</i>	ІНФОРМАЦІЙНА ПІДСИСТЕМА МУЛЬТИСПЕКТРАЛЬНОГО АНАЛІЗУ АЕРОФОТОЗНІМКІВ У СКЛАДІ СИСТЕМИ МОНІТОРИНГУ СТАНУ ПОСІВІВ	196
44.	<i>КОТЛЯР І. С. ГРИЩА О.В.</i>	АНАЛІЗ ВИКОРИСТАННЯ СУЧАСНИХ АЛГОРИТМІВ МАСШТАБУВАННЯ	200
45.	<i>ПАСТРЕЛО Н., КОГАН А.В.</i>	СПОСІБ ОРГАНІЗАЦІЇ БАГАТОШЛЯХОВОЇ МАРШРУТИЗАЦІЇ В SDN	205
46.	<i>ЛАЦІНА Р.О.</i>	ЕТЕКТУВАННЯ РУХІВ ЛЮДИНИ НА ВІДЕО ПОСЛІДОВНОСТІ В РЕАЛЬНОМУ ЧАСІ	210
47.	<i>ЧЕРНУШЕВИЧ Д.О., ОЛІЙНИК Ю.О.</i>	ПІДХІД ДО АНАЛІЗУ ПОВЕДІНКИ КОРИСТУВАЧІВ ВЕБ-САЙТІВ	216
48.	<i>ДРЕГАЛО Л. В. ТЄЛИШЕВА Т. О.</i>	ОГЛЯД МОДЕЛЕЙ АНАЛІЗУ ТРАНСПОРТНОЇ ДОСТУПНОСТІ РАЙОНІВ МІСТА	219
49.	<i>АНДРУСЯК К.А. ЖДАНОВА О.Г.</i>	ПІДСИСТЕМА АВТОПІЛОТУВАННЯ ТА МОДЕЛЮВАННЯ РОБОТИ БЕЗПІЛОТНОГО АВІАЦІЙНОГО КОМПЛЕКСУ У СКЛАДІ СИСТЕМИ МОНІТОРИНГУ СТАНУ ПОСІВІВ	222
50.	<i>НАФІЕВ А.Е. ГРИЩА О.В.</i>	БЕЗПЕКА ВІРТУАЛІЗАЦІЇ: МОНІТОРИНГ ТА ІНТРОСПЕКЦІЯ ВІРТУАЛЬНОЇ МАШИНИ	225

УДК 004.021

ЧЕРЕДНІЧЕНКО В. О.,  
ЗАЛЕВСЬКА О. В.

## ТРИВИМІРНІ КЛІТИННІ АВТОМАТИ В ГЕНЕРАЦІЇ ОБ'ЄКТІВ КОМП'ЮТЕРНОЇ ГРАФІКИ

У цій роботі вивчено деякі нові застосування клітинних автоматів у галузі цифрової графіки, особливо при створенні тривимірних полігональних сіток.

Наголос буде зроблено даним про утворення об'єктів в природі, але також будуть приклади штучних і абстрактних форм. Буде обговорено способи ефективного використання згенерованих продуктів клітинних автоматів в контексті графічних додатків в режимі реального часу. Буде згенеровано печерну систему за допомогою клітинного автомату, що буде симулювати природню генерацію та розповсюдження печери.

**КЛЮЧОВІ СЛОВА:** клітинний автомат, комп'ютерна графіка, моделювання об'єктів, процедурна генерація, процедурне текстурування, генетичний алгоритм

In this thesis we study some novel applications of cellular automata in the field of digital graphics, especially in generating three-dimensional polygon meshes. Emphasis will be given on formations found in nature, but examples of artificial and abstract forms are also included. We will discuss ways to effectively use the generative product of automata in a real-time graphics application context and compare it with other popular methods for mesh generation. We will generate the complex cave system with the three-dimensional sphere cellular automata, that will look like the nature-created cave system.

**KEYWORDS:** cellular automaton, computer graphics, object modelling, procedural generation, procedural texturing, genetic algorithm

### 1. Вступ

#### 1.1. Клітинний автомат

Клітинний автомат або КА складається з сітки регулярних комірок. Кожній клітині присвоюється одне з кінцевого числа станів. У кожній ітерації клітини змінюють стан синхронно за допомогою певної функції, що бере як вхідний параметр стан самої клітини, а також їхніх сусідів. Зазвичай стан клітини представлено як ціле чи логічне значення [3]. Сітка може бути будь-якою, але ми в основному занурюємося у тривимірні автомати, так як вони мають сучасні застосування в комп'ютерній графіці. Околиця клітини в одномірній сітці зображена на рисунку 1.1.1. Клітини в найближчому оточенні показані зеленим кольором, ті, що в жовтому та червоному кольорах, являють собою розширене сусідство в радіусі 2 і 3 від клітини.



У кожній ітерації автомату правило синхронно оцінює кожен комірку та змінює сітку відповідно. Поведінка простого правила показана на рисунку 1.1.2. А кінцева одномірна сітка ініціалізується однією сірою клітиною (станом 1) посередині. Для кожної ітерації, клітина стає сірою, якщо в його сусідстві існує одна або дві сірі клітини. В іншому випадку вона перетворюється або залишається білою (стан 0). Зауважте, що клітина входить до радіусу оцінки.

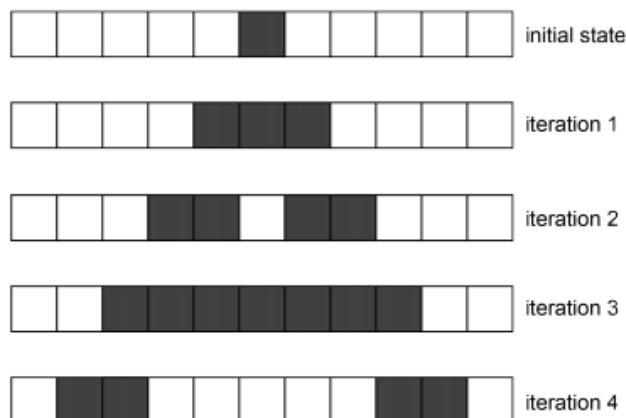


Рис. 1.1.1 Сусіди сірої клітини в одно-вимірній сітці

Рис. 1.1.2 Чотири ітерації правила елементарного клітинного автомата на одновимірній сітці

Розширення цієї концепції до двох чи трьох вимірів дає більше можливостей вибору околиці. Найпоширенішими є райони фон Неймана та Мура [1] і їх розширення, показані на рисунках 1.1.3 і 1.1.4.

Для цілей цієї статті ми розглядаємо найближчий район Мура, що складається з 26 клітин (включаючи діагоналі) та його розширення. Кількість клітин у розширеному радіусі сусідства Мура  $r$  в  $d$ -розмірах ( $2 \times$  радіус)

$d - 1$ . Насправді радіус являє собою відстань Чебишева або шахову дошку, тобто всі сусідні клітини на позиціях пр клітини в положенні  $sr$ .

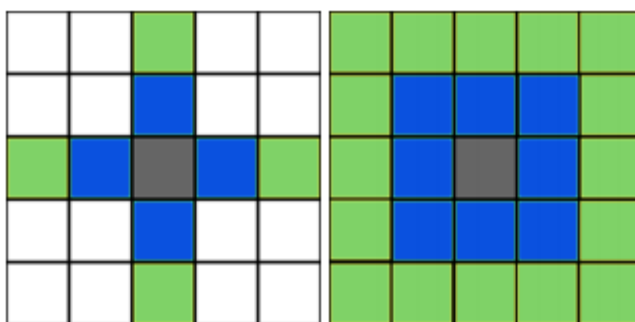


Рис.1.1.3 Область сусідства для двовимірної сітки. Зліва: Фон-Неймана, справа – Мура

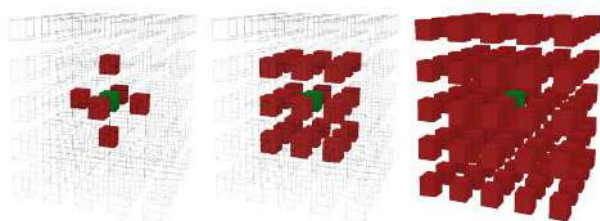


Рис.1.1.4 Область сусідства для тривимірної сітки. Зліва-направо: Фон-Неймана, Мура, Мура (розширена)

## 2. Генерація об'єктів за допомогою клітинних автоматів

### 2.1. Печерні структури

Клітинні автомати (КА) широко використовуються для генерації печерних рівнів [9]. Реалізації таких генеріцій в останніх іграх приділяють основну увагу створенню макетів двовимірного рівня. Для цієї статті ми перетворимо двомірне

правило печери до трьох вимірів і дослідимо способи його вдосконалення та параметризації.

Для цього ми використовуємо автомат із двома станами твердого (1) та повітряного (0). Відповідно до цього правило автомата, кожна клітина змінює стан на основі співвідношення повітря та твердих клітин навколо неї. Параметр зв'язності визначає, наскільки важливим співвідношення станів твердих та не твердих клітин навколо має бути, щоб це змінило стан клітини. Значення за замовчуванням - 0.5. Відсоток твердих клітин в примітивній структурі називається щільністю.

Правило: печерна структура

Вихідні дані: клітина

Кількість клітин  $\leftarrow$  кількість сусідніх клітин з таким же станом

Співвідношення  $\leftarrow$  кількість клітин /  $(2 * \text{радіус} + 1) ^ 3$

Якщо співвідношення  $<$  параметр приєднаності, то

Клітина  $\leftarrow$  стан більшості

Структури, що відображаються на наступних рисунках, не є порожніми зсередини. Їм потрібно бути інвертованими, щоб діяти, як передбачено. У випадку, якщо ми хочемо отримати поверхню з цього шару, то остаточний результат буде таким же.

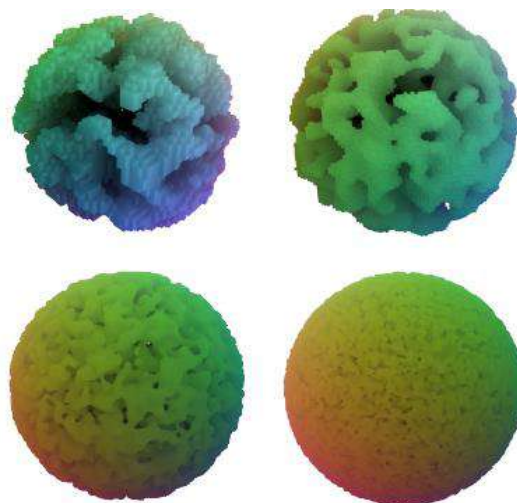


Рис. 2.1 Правило печери застосовано протягом 7 ітерацій для сфери з щільністю

0.5. Радіус пошуку виставлений на 1 одиницю. Зліва-направо: 32, 64, 128, 256 розширення сітки. Параметр приєднаності: 0.5

Ми спостерігаємо, що збільшення роздільної здатності моделювання призводить до повторення патерну печери, що є бажаною поведінкою у випадку, коли ми хочемо розширити наш світ зберігаючи таку ж роздільну здатність для вокселів для печерних структур. Є, звичайно, шляхи масштабувати структури або підвищити рівномірність печери. Ми будемо розглядати деякі з них.

## 2.2. Кількість ітерацій

Правило структури печери вимагає невеликої кількості ітерацій для отримання бажаного результату, в більшості випадків достатньо 4. Більше ітерацій приводять до більш гладкої маси з менш з'єднаними печерами. Я помітив, що при зростанні кількості ітерацій відбуваються менший вплив на обсяг. Ця поведінка спостерігається на рисунку 2.2.1.

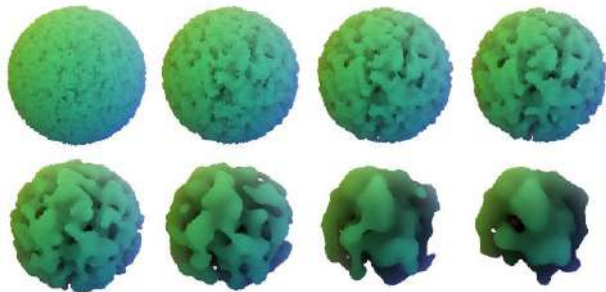


Рис. 2.2.1 Селективні ітерації з правилом печери, застосовані для сфери з роздільною здатністю  $64^3$ , щільністю 0,5 ступенем приєднаності 0,5 та радіусом 1. Зліва – направо: 1,2,3,4,5,13,20,37 ітерацій.

## 2.3. Генерація структури печери

Нашою метою було створити невеликий і зручний рівень, з добре структурованими печерними зв'язками, які гравець може легко пройти і дослідити. Ми використали куб з 2563 клітинами і 0,49 початковою щільністю, і застосували правило печери на 40 ітерацій, використовуючи радіус 3 і зв'язність 0,5 Отриманий об'єм показаний на рис. 2.3.1.

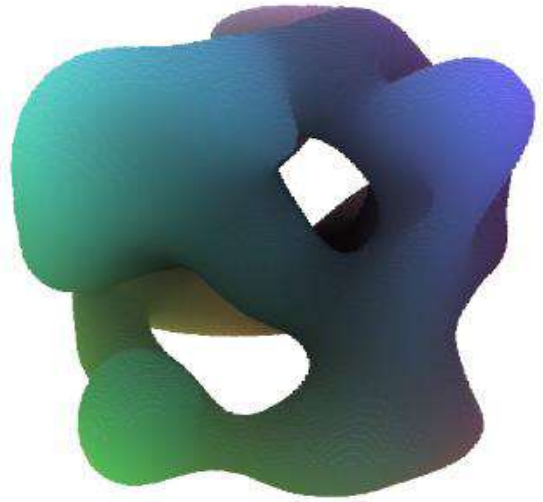


Рис.2.3.1 Згенерована печерна структура

Текстурування створює виклик для КА і всіх процедурних згенерованих сіток. Зазвичай текстура відображається на моделі через UV-координати, що зберігаються у кожній вершині багатокутної сітки [7]. Це, як правило, ручний процес, який допомагає розпаковувати алгоритми і різні інструменти. Для наглядності, зобразимо інтер'єр печерної структури, зображену на рисунку 2.3.2 використовуючи іншу техніку. Дифузна текстура проєктована за допомогою трипланарного відображення та модулюється по схилу і нормальній орієнтації У трипланарному відображенні текстури відбираються з використанням світу вершин в координатах простору, а потім змішані на основі нормальних вершин. Кілька текстур може об'єднати на основі проєкційної осі. Щоб забезпечити подальшу варіацію, відносний нахил багатокутної поверхні та інформації про висоту використовується для модуляції кольору текстури.



Рис. 2.3.2 Середина печерної структури після отримання поверхні та застосування мапінгу текстур



### 3. Висновки

Метою даної роботи є комплексне введення в 3D методи, що допомагають створити комп'ютерну графіку та рівні гри в режимі реального часу за допомогою клітинних автоматів. Наведені в розділі 2 застосування, є новими підходами до процедурної генерації, що не дивно, враховуючи властиві обмеження та непередбачуваність КА. Зосереджуючи дослідження на простих правилах, ми виявили досить багато корисних конфігурацій з можливістю створення вірогідних гірських утворень, печерних систем, сталагмітів та інших складних природних або фізичних структур, при цьому вони легко параметризуються і об'єднуються. Ми також обговорили методи ефективного використання генеративних продуктів автоматів у графічному движку.

### СПИСОК ЛІТЕРАТУРИ

- [1] Andrew Adamatzky and Genaro J Martinez. ' Designing Beauty: The Art of Cellular Automata. Vol. 20. Springer, 2016.
- [2] Alonso-Sanz and Martin. "Elementary cellular automata with memory". Complex Systems 14.2 (2003), pp. 99–126.
- [3] Bays. "A note about the discovery of many new rules for the game of threedimensional life". Complex Systems 16.4 (2006), p. 381.
- [4] Conway. "The game of life". Scientific American 223.4 (1970), p. 4.
- [5] Culik and Dube. "Fractal and recurrent behavior of cellular automata". Complex Systems 3.3 (1989), pp. 253–267.
- [6] RL Dobrushin, VI Kriukov and AL Toom. Stochastic cellular systems: ergodicity, memory, morphogenesis. Manchester University Press, 1990.
- [7] Fisch. "Cyclic cellular automata and related processes". Physica D: Nonlinear Phenomena 45.1-3 (1990), pp. 19–25.
- [8] Gobron and Chiba. "3D surface cellular automata and their applications". The Journal of Visualization and Computer Animation 10.3 (1999), pp. 143–158.
- [9] Johnson, Yannakakis and Togelius. "Cellular automata for real-time generation of infinite cave levels". Proceedings of the 2010 Workshop on Procedural Content Generation in Games. ACM. 2010, p. 10.
- [10] Ju, Losasso, Schaefer and Warren. "Dual contouring of hermite data". ACM Transactions on Graphics (TOG). Vol. 21. 3. ACM. 2002, pp. 339–346.

## ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АВТОМАТИЧНОГО ТЕСТУВАННЯ ВЕБ-ЗАСТОСУВАНЬ

Робота присвячується питанням вдосконалення технологій автоматичного тестування веб-застосувань, розроблених на базі скрипкових мов. Тестування програмного забезпечення - перевірка відповідності між реальною і очікуваною поведінкою програми, що здійснюється на кінцевому наборі тестів, обраному певним чином.

The work is devoted to improving the technologies of automatic testing of web applications developed on the basis of violin languages. Testing software - checking the correspondence between the real and expected behavior of the program, carried out on the final set of tests, selected in a certain way.

### 1. Вступ

У сучасному світі програмні системи оточують нас всюди. Вони вже давно увійшли в наше життя інтуїтивно зрозумілими і простими для користувача. Зараз для нас використання - звичайна справа.

З плином часу, сам процес розробки змінювався. з'являлися нові інструменти та методології, що дозволяють структурувати, прискорювати сам процес розробки, робити його більш передбачуваним щодо термінів і результату. А також дозволяти швидко підлаштовуватися під мінливі вимоги замовника прямо в процесі розробки, до того, як продукт буде випущений. При розробці продукту, необхідно, щоб в результаті, після випуску, він працював правильно, без помилок, а також відповідав вимогам замовника. Але також, бажано, щоб вже в процесі розробки, розробники могли мати частково функціонуючий продукт, з обмеженим, але надійно працюють і відповідним вимогам функціоналом, який згодом буде розширюватися. Це завдання допомагає вирішити такі процедури, як тестування.

Отже удосконалення методів тестування, перехід до автоматизації цього процесу та спрощення процесу розробки автоматичних тестів є актуальним.

### 2. Автоматичне тестування

Тестування програмного забезпечення - перевірка відповідності між реальним і

очікуваним поведінкою програми, що здійснюється на кінцевому наборі тестів, обраному певним чином.

**Автоматичне тестування** це частина процесу тестування на етапі контролю якості в процесі розробки програмного забезпечення. Воно використовує програмні засоби для виконання тестів і перевірки результатів виконання, що допомагає скоротити час тестування і спростити його процес.

Головна мета автоматизації - прискорити процес тестування без втрати якості, а також позбавити ручного тестувальника від рутини проходження регресійних сценаріїв, дозволивши сфокусуватися на перевірці бізнес-логіки додатка з точки зору кінцевого користувача.

Існує кілька рівнів автоматичного тестування:

- Unit tests;
- Integration tests;
- GUI tests;
- Manual tests.

Unit і Integration tests розробляються розробникам, Manual тестування здійснюють тестувальники. Ми зупинимося більш детально на GUI-автоматизації, що набирає популярність величезними темпами и може в рази зменшити об'єм робіт з Manual тестування.

GUI-автоматизація - найбільш поширений вид автоматизації тестування шляхом тестування програми через графічний інтерфейс користувача (GUI).

Головна його перевага в тому, що додаток тестують точно так, як його буде використовувати кінцевий користувач. Також цей підхід дозволяє тестувати без доступу до вихідного коду програми.

Основні плюси GUI автоматичного тестування:

- Виключений людський фактор. Автоматизація гарантує, що тест-скрипт завжди буде виконаний однаково, виключаючи помилки з необережності.
- Швидкість виконання - час виконання автоматизованого тест-скрипта в разі менше мануального проходження.
- Менші витрати на підтримку - один раз написані скрипти вимагають набагато менше часу на підтримку і аналіз результатів.
- Звіти - в результаті прогону генерується звіт з подальшим розсиланням всім зацікавленим особам.
- Виконання тестів у зручний час - Автотест можуть бути запущені в будь-який зручний час або по певній події. Часто це нічні прогони і тестування в неробочий час, що дозволяє раціональніше використовувати тест-ресурси.

Основні мінуси UI тестування:

- На написання навіть невеликого набору автоматичних тестів бути витрачено багато часу
- Процес розробки автоматичних тестів може бути запущений тільки після закінчення розробки, оскільки

якщо писати автоматичні кейси на проміжній версії функціоналу, що буде дороблюватися, велика вірогідність того, що автоматичні тести необхідно буде перероблювати

- При зміні функціоналу застосування майже завжди необхідно буде проводити рефакторинг автотестів.

Запропонована технологія – це технологія за допомогою якої можна в спростити процес створення та рефакторингу автоматичних тестів та сценаріїв. Для створення автоматичних тестів за допомогою цієї технології не так важливо мати чіткі вимоги для програмного продукту, оскільки автоматичні тести генеруватимуться з готового веб застосування.

За основу, тобто вхідні данні береться код веб сторінки веб застосування. Сама технологія дозволить на базі DOM (Document Object Model) – об'єктної моделі документу, згенерувати можливі дії із об'єктами документу і зафіксувати зміни у DOM при їх виконанні. Після чого усі дії і зміни складатимуть повну матрицю покриття станами веб застосунку до певної глибини сканування сторінок. Глибина аналізу сканування станів може регулюватися користувачем технології. З готової матриці покриття станами веб застосунку, до якої підв'язані всі згенеровані перевірки, користувач може сформувати тестовий сценарій для подальшої фінального налаштування та запуску.

### 3. Висновок

Отже запропонована для виробників і постачальників ПО, завдяки простій підтримки системи, оновленню і контролю доступу користувачів. Система розташована в одному місці і видається як сервіс через інтернет, а не встановлюється кожному користувачеві на комп'ютер. Розробникам потрібно підтримувати працездатність тільки однієї системи, а не всіх копій, встановлених у користувачів. але така модель висуває також і підвищені вимоги до надійності і цієї системи.

### Література

1. Канер Сэм. Тестирование программного обеспечения - Издательство «ДиаСофт», 2001. — 544 с.
2. James Whittaker. How to Break Web Software - Addison-Wesley Professional, 2 лют. 2006 р. - 240 стор.

## ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ КЛАСИФІКАЦІЇ КОРИСТУВАЧІВ НА ОСНОВІ ПОВЕДІНКОВОЇ МОДЕЛІ

В даній статті розглянуто можливі способи класифікації користувачів на основі їхньої поведінкової моделі. Сформульовано ціль та задачу, яка є необхідною для вирішення з точки зору надання рекомендацій, і також різноманітних маркетингових стратегій (надання реклами, тощо). Виділено основні фактори для аналізу, за якими здійснюється класифікація користувачів.

In this article, one can see the methods of users classification based on their behavioral model. Formulated the problem and its objectives, that we need to resolve from a point of view, of giving relevant recommendations, and different marketing strategies (for example advertisements). Were chosen main factors for analysis, based on which classification will be done

**Ключові слова:** задача класифікації, асоціативні правила, дерева рішень, маркетингова стратегія, поведінкова модель.

### 1. Вступ

В наш час величезного попиту отримали різноманітні рекомендаційні системи, які дають змогу користувачу значно економити свій час, при виборі якоїсь певної інформації. Так, як вони намагаються передбачити, яка інформація буде цікавою для користувача. Вони порівнюють однотипних користувачів та однотипні дані від різних людей, і визначають на основі цього рекомендація для конкретного користувача. Звичайно, що чим кращий сам рекомендаційний сервіс, тим більше користувачів його використовують. А якість рекомендаційної системи визначається тим, наскільки релевантними та актуальними є рекомендації для користувача в даний момент часу. Відповідно, якщо система справді надає актуальні та хороші рекомендації, то в неї появляться аудиторія, а з появою великої кількості зацікавлених людей, настає момент, який дає змогу отримання прибутку з системи.

В даному випадку рекомендаційні системи це саме той простір, де можна надати окрім релевантних рекомендацій також і релевантну рекламу, яка буде мати високу конверсію, за рахунок того, що реклама буде таргетованою, тобто розсилається лише зацікавленим людям. І тому саме такі системи мають значний попит в рекламодавців.

Звичайно, що розсилка реклами здійснюється не конкретно для кожної людини окремо, а для групи людей, які мають спільні інтереси, хобі, та мають в собі якісь спільні ознаки. І тут виникає декілька питань, на які необхідно отримати відповіді. Яким чином

згрупувати людей? Як найти в них щось подібне? Як і де отримати необхідні дані, для визначення класу користувача?

### 2. Збір даних

При класифікації, якої небудь інформації одним з найважливіших питань є питання збору або ж отримання даних (в нашому випадку даних про користувачів). Є різні механізми, використовуючи які можна досягти поставленої цілі, проте виділяють 2 основні:

- явне отримання даних;
- неявне отримання даних.

Явне - дає змогу отримати чітку інформацію про користувача, яка може його однозначно ідентифікувати. Це може бути різного роду інформація:

- заповнена анкети;
- профіль соціальних мереж;
- проведене опитування;
- тощо.

Неявне — дає змогу отримати інформацію, яка не є чіткою, проте з якої можна отримати певні знання. Наприклад:

- геолокація користувача;
- часте відвідування якихось місць;
- у випадку з електронними даними - це інформація про перегляд різних типів сторінок, скільки часу користувач знаходиться на сторінці, його активність, кількість кліків тощо;
- тощо.

Тобто неявне отримання даних дає змогу отримати інформацію, яка є динамічною, і яка

може охарактеризувати в більшій мірі саме поведінкову модель користувача.

Тому, для того, щоб отримати якомога більш якісні та валідні дані необхідно використовувати 2 підходи одночасно, так як це дасть змогу покрити більший широкий спектр інформації. Враховуючи вище сказано при аналізі та побудові моделі користувача був обраний варіант використання інформації про користувача з соціальних мереж, так, як там міститься як і явна інформація (профіль користувача) так і неявна (його репости, лайки).

### 3. Модель користувача

Кожний метод класифікації включає в себе алгоритм навчання системи, для того ідентифікувати модель, що найбільш підходить для зв'язку між набором атрибутів та класом. Створена модель має працювати коректно, як уже на навчених даних так і на даних, для яких має відбуватись передбачення до якого класу вони відносяться.

Для того, щоб здійснити класифікацію користувачів необхідно перш за все сформувати набір змінних, які зможуть його описати, тобто створити модель користувача. При цьому можна згрупувати ці дані в певні набори:

- географічні дані — дані, що містять інформацію про країну, клімат, геолокацію користувача — які зможуть задати глобальну категоризацію користувача.
- загальні дані — дані, що містять інформацію про вік, стать, коротку біографію, шлюбний статус, категорію роботи — які зможуть задати середньостатистичну інформацію про людину.
- дані з соціальних мереж — дані, що містять інформацію, про поведінку користувача в соціальній мережі.

Беручи до уваги те, що на даний момент більшість людей активно використовують соціальні мережі в своєму повсякденному житті, то можна сказати, що їхня поведінка в соціальних мережах в певній мірі відповідає їхній поведінці в повсякденному житті. В результаті, були обрані такі данні з соціальних мереж, які можуть бути найбільш корисними для класифікації, основані на поведінці:

- кількість постів та лайків користувача в день;

- кількість відповідей користувача в день;
- кількість хештегів в постах;
- кількість посилань в постах;
- середній інтервал часу між новими постами;
- середня кількість слів в кожному пості.

Також є необхідність враховувати й контент, який постить чи лайкає користувач, тому для цього було обрано наступні дані:

- топ 3 теми, про які користувач пише пости;
- топ 3 теми, які користувач лайкає;
- середня кількість лайків на постах, які лайкнув користувач;
- середня кількість лайків на постах самого користувача.

Для того, щоб визначити щоб визначити теми, постів використовується класифікатор тексту, який на вході отримує цілий текст посту, а на виході повертає категорії до яких відноситься даний текст, аналогічно відбувається й з лайками постів. Всі усереднені значення (наприклад середня кількість постів користувача в день), рахуються за формулою середнього арифметичного:

$$\bar{x} = \frac{1}{n} \cdot \sum_{i=1}^n x_i$$

### 4. Класифікація користувачів

Класифікація даних — процес організації вхідних даних у категорії для їхнього подальшого ефективного використання відповідно до особистих або певних бізнес цілей.

Вона тісно пов'язана з кластеризацією даних, але на відмінну від кластеризації є передбачуваною, що дає змогу подальше обробляти її вихідні дані, для отримання наступних необхідних результатів.

Вся задача класифікації по факту зводиться до мапінгу атрибуту  $x$  до класу  $y$ . Тобто наводиться наступна постановка задачі: існує множина об'єктів  $X = \{X_1, X_2, \dots, X_n\}$ , які розділені на класи  $Y = \{Y_1, Y_2, \dots, Y_n\}$ . Задана вибірка елементів  $Z$ , для яких уже визначено, до якого класу вони належать. Для інших елементів, необхідно визначити, до якого класу вони відносяться. Відповідно має існувати залежність та бути побудований алгоритм,  $f: X \rightarrow Y$  такий, що здатний класифікувати будь який об'єкт  $x \in X$ .

На даний момент існує велика кількість алгоритмів, що дають змогу класифікувати дані, проте в даній роботі використовується дерево рішень, так, як воно основане на чітких правилах класифікації, дає змогу легко візуалізувати як саму модель, так і прогноз окремо взятого елемента.

Основна ідея дерева рішень полягає в тому, щоб розділити вхідні дані настільки, доки не будуть отримані однородні дані. Множина правил, що дають змогу зробити таке розбиття, дозволяють надалі надавати прогноз для нових даних. Графічно зображається в деревовидній структурі, де моменти прийняття рішень — узли. В цих узлах відбувається розгалуження на основі від зробленого вибору. Кінцеві узли — листя, що є результатом послідовності прийняття рішень. Вхідні дані, які необхідно класифікувати знаходяться в корені дерева, і в залежності від прийнятого рішення процес закінчується в одному з листків, що надасть результат.

Основна ідея методу — рекурсивний поділ. Процес поділу відбувається доти, доки не виконається критерій, що означає завершення поділу, зазвичай це можуть бути такі критерії:

- всі дані належать до одного класу;
- не осталося жодних признаков, за якими можна й надалі розбити дерево;
- дерево досягло заданий ліміт росту.

Вибір признаку по якому відбувається розбиття виконується таким чином, щоб отримати якомога чистішу групу. Для того, щоб визначити міру чистоти групи вводиться поняття ентропії:

$$E(S) = \sum_{i=1}^n -p_i \log_2(p_i),$$

де  $p_i$ - імовірність знаходження системи в стані  $i$ .

Даний признак знаходиться методом перебору. Для кожного признаку  $S$  знаходиться значення приросту інформації, який обчислюється як різниця ентропії до та після розбиття:

$$I(S) = E(S_1) - E(S_2),$$

де  $I(S)$ - сумарна ентропія груп отриманих при розбитті. Відповідно чим більше значення приросту інформації, тим більше цей признак нам підходить, адже він дає змогу отримати найбільш однорідні групи.

Алгоритм побудови дерева.

1. На поточному кроці обирається признак, при розподілі по якому приріст інформації буде найбільшим.
2. Якщо ентропія рівна нулю, або менша заданій величині, то закінчити розбиття дерева, в іншому випадку повторити пункт 1.

## 5. Модифікація алгоритму

Модифікація самої класифікації, яка здійснена в даній роботі заключається в тому, що використовується також і внутрішня класифікація тексту постів, що лайкає чи пише користувач, що дає змогу більш детально оцінити його інтереси саме на рівні його поведінки з часом. Також варто зазначити, що в нас вводиться динамічна зміна, яка відповідає за топ 3 категорії, що вказує, який процент цих категорій підпадає на всі пости чи лайки користувача. Тобто у користувача  $U_i$  може бути заданий наступний набір:

- ТОП1: Фільми (50%)
- ТОП2: Книги (22%)
- ТОП3: Музика (12%)

В результаті ми маємо класифікацію, яка залежить від цього ймовірнісного значення топових категорій. Тому наприклад, ми зможемо сказати, що з імовірністю 55% (у випадку, якщо цей критерій є найбільш вагомим), користувач захоплюється фільмами і в результаті надати йому рекомендації основані на даному твердженні. При цьому дана класифікація є динамічна так, як топ 3 категорій може змінюватись з часом, відповідно до цього ми можемо, формувати нові рекомендації, що будуть актуальними для користувача на поточному етапі його життя.

## 6. Висновки

На основі різноманітних даних про користувача, які отримані з соціальних мереж, було створено його поведінкову модель, що дає змогу надати йому таргетовані рекомендації та рекламу. Був проведений аналіз методу класифікації, та введені імовірнісну міру, що дає змогу динамічно надавати рекомендації користувачу з часом.

В подальшому, для покращення методу, можна вводити також нові дані, для збільшення якості моделі, наприклад:

- аналіз змістовного забарвлення тексту;
- визначення класу слів;

- складання шаблону поведінки користувача, на основі змін його поведінки.

### Список літератури

1. Types of classification algorithms in Machine Learning [Електронний ресурс] : Medium – 2017. – Режим доступу: <https://medium.com/@sifium/machine-learning-types-of-classification-9497bd4f2e14>
2. Brett Lantz. Machine Learning with R. Packt Publishing, Birmingham — Mumbai, 2013
3. F. Benevenuto, T. Rodriguesy, M. Cha, and V. Almidia Characterizing User Behavior in Online Social Networks – 2009. – С. 49–62.
4. Content-Based Recommendation Systems [Електронний ресурс] : Michael J. Pazzani<sup>1</sup> and Daniel Billsus<sup>2</sup>–Режим доступу: <https://pdfs.semanticscholar.org/3444/6adc7d701a2c3a89c2fc5f6d3479eef407b0.pdf>
5. Machine Learning Algorithms for Classification [Електронний ресурс] : Machine Learning Algorithms for Classification. Rob Schapire – Режим доступу: <https://www.cs.princeton.edu/~schapire/talks/picasso-minicourse.pdf>

## ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ КЛАСТЕРИЗАЦІЇ КОРИСТУВАЧІВ ДЛЯ ПІДГОТОВКИ ЦІЛЮВИХ ПРОПОЗИЦІЙ

В даній статті розглянуто спосіб підготовки рекомендацій користувачам на основі побудованих кластерів. Визначено задачу та цілі даної системи. Обрано методи, які будуть використовуватися для кластеризації користувачів та надання рекомендацій. Описано модифікацію даних методів, та алгоритм їх застосування. Даний опис наведено для даних про користувача, які можна отримати з соціальної мережі facebook.

In this article are reviewed ways of preparing recommendations which based on clusters of users. There are defined tasks and goals of the system. Chose methods which will be used to cluster users and give them recommendations. Described modifications of the methods and algorithm of using them. The description is showed for data of an user which we can get from social network facebook.

**Ключові слова:** кластеризація, нечітка кластеризація, рекомендація, міра схожості, колаборативна фільтрація.

### 1. Вступ

Кожен із нас щодня отримує досить велику кількість інформації. Не завжди отримана інформація є релевантною, що зумовлює необхідність пошуку джерел, які будуть цікавими та яким можна довіряти. Більшість людей мають роботу, також багато хто має якісь захоплення. В обох випадках перед особою може постати задача у вивченні чогось нового, в поглибленні знань в певній сфері. Також майже кожен щодня дивиться фільми або читає книги, або ж прослуховує музику. І відповідно щоб виконати ці дії перед тим потрібно знайти матеріал, який буде нам цікавим. Досить часто виникає ситуація, коли особа тоне в тій величезній кількості матеріалів, коли шукає, саме те що їй потрібно. Адже до прикладу ми можемо знайти до 10, а інколи і більше книг на одну тему, а перечитати всі в нас просто не вистачить часу.

В описаних випадках виникає необхідність спитати в когось поради, і звісно не завжди в нас є така можливість. В таких випадках на допомогу, приходять саме рекомендаційні системи.

Більшість рекомендаційних систем видають пропозиції на основі раніше переглянутих або на основі схожих матеріалів. Ще одною досить вагомою проблемою таких систем є спрямованість лише на якусь одну тему. Дані системи часто не справляються з роллю поради, коли особу цікавлять декілька тем. Також у випадку, якщо людина хотіла б дізнатися, що цікаво її однодумцям, то

рекомендація на основі раніше переглянутих матеріалів може виявитися не зовсім підходящою.

Тому доцільною є розробка технології для об'єднання однодумців.

Виникає лише проблема де взяти дані на основі, яких можна було б об'єднати цих користувачів. Ця проблема вирішується за допомогою активного соціального життя сучасних людей в мережі інтернет. Тому вирішено об'єднувати користувачів на основі діяльності їх в соціальних мережах. Завдяки застосуванню технологій з науки про дані система буде мати можливість кластеризувати користувачів зі схожими інтересами, що надасть можливість рекомендувати користувачам з врахуванням їх інтересів.

### 2. Підготовка даних

Для надання кластеризації та надання рекомендацій потрібні дані двох видів:

- які описують користувача;
- які відображають вподобання користувача.

Для отримання даних, що описують особу скористаємося відкритим API. Виділимо інформацію, з тієї яку можна отримати, що буде використовуватися:

- домашня адреса;
- поточна геолокація;
- вік;
- стаття;
- кількість друзів;



- лайки, які користувач поставив сторінкам: (книг, фільмів, музики, телепередач)

Для отримання даних, що описують вподобання користувача будуть використані оцінки користувачем певних об'єктів (книга, фільм, музика або телепередача) в даній рекомендаційній системі. Значення оцінки можуть бути від 0 до 5.

Для «холодного старту» використовуються лайки з фейсбуку. Тобто зіставляються назви сторінок, яким користувач поставив лайк та об'єктам у системі. У випадку збігу об'єкту виставляється автоматично оцінка «5», у випадку, коли об'єкт не збігається з жодним лайком значення оцінки дорівнює «0».

Побудувати вектор з усіх наявних об'єктів, що описує користувача, та застосувати його для кластеризації, навряд є можливим. Тому кластеризація буде здійснюватися для кожного типу об'єктів окремо. Також у вектор буде входити топ 100 об'єктів даного типу за кількістю оцінок користувачів в даній інформаційній системі, наприклад топ 100 книг, що зібрали найбільшу кількість оцінок користувачів. Даний вибір зумовлено тим, що користувача на основі ставлення до найбільш популярних книг можна віднести до певної групи.

### 3. Постановка задачі

Задача складається з двох підзадач:

- об'єднати користувачів у кластери;
- використати кластери для надання рекомендацій.

В загальному задача має наступний вигляд:

Дано множину об'єктів даних  $I$ , кожен з яких є вектором атрибутів. Потрібно знайти таку функцію  $F$ , яка розіб'є множину на кластери  $C$ , та вкаже на ймовірність належності об'єкта кластеру  $\mu(i)$ .

$$\{i_1, i_2, \dots, i_n\}$$

Де  $n$  - кількість об'єктів (користувачів).

$$C \quad \{c_1, c_2, \dots, c_k, \dots, c_m\}$$

Де  $m$  - кількість кластерів, яка в даному випадку не є зарання відомою.

$c_k$  - кластер, що містить в собі схожі об'єкти з множини  $I$ .

$$c_k \quad \{i_j, i_g \mid i_j \in I, i_g \in I \ \& \ d(i_j, i_g) < \sigma \}$$

Де  $\sigma$  - величина, яка визначає значення близькості для включення об'єкту до кластеру,  $d(i_j, i_g)$  - відстань між об'єктами.

$$\mu_k(i_j) \quad [0,1]$$

$$\sum_{k=0}^m \mu_k(i_j) = 1$$

Дані обмеження визначають, що об'єкт може належати різним кластерам з різними ступенями приналежності і сума приналежності об'єкта всім кластерам дорівнює одиниці.

Дані кластери тепер необхідно використати, як узагальнених користувачів. Це дає змогу використати колаборативну фільтрацію засновану на сусідстві. Де функція схожості, виглядає наступним чином:

$$simil(x, y) \quad \mu_y(x)$$

де  $x$  - поточний користувач,  $y$  - узагальнений користувач, тобто кластер.

### 4. Кластеризація користувачів

Кластерний аналіз походить з природничих наук, і остаточно став відомим в середині минулого століття. З розвитком комп'ютерної техніки, почався новий етап розвитку даного напрямку. Саме тому існує досить велика кількість алгоритмів кластеризації. Всі ці алгоритми об'єднуються в певні моделі. Основними є наступні моделі:

- Моделі зв'язності. Наприклад: ієрархічна кластеризація.
- Центроїдні моделі. Наприклад: метод К-середніх (K-means).
- Статистичні моделі, де кластери будуються ґрунтуючись на статистичних розподілах.
- Моделі засновані на щільності. Наприклад: DBSCAN, OPTICS.
- Групові моделі. Дані алгоритми просто описують групування об'єктів.
- Графові моделі. Наприклад: алгоритм поширення близькості
- Нейронні моделі. Найбільш відомою моделлю нейронної мережі з навчанням без учителя є нейронна мережа Кохонена.

Кожен з даних методів, має свої переваги і недоліки. Проте для даної роботи було обрано саме метод поширення близькості (Affinity Propagation).

Перевагою даного методу є те, що не потрібно зарання задавати центри класів, а

також кількість кластерів. Також нам не важлива кількість елементів, і відповідно їй не потрібно задавати як в деяких методах.

### 5. Колаборативна фільтрація

Методи колаборативної фільтрації застосовуються для надання рекомендацій.

Ці методи використовують наступні дані:

- оцінки інших користувачів певних об'єктів;
- оцінки поточного користувача.

Для врахування оцінок інших користувачів, потрібно врахувати схожість поточного користувача на них. Для цього мають бути всі необхідні дані, що є атрибутами вектора користувача. Дані які використовуються для характеристики користувача описувалися раніше.

Колаборативна фільтрація може бути заснована на :

- пам'яті;
- сусідстві;
- моделях.

Найкраще, для цілей вказаних в цій публікації. Підходить метод колаборативної фільтрації заснований на сусідстві. Адже отримані кластери будуть враховуватися, як узагальнені користувачі, а ймовірність належності користувача до кластеру є мірою схожості даного користувача з узагальненим. Але для цього потрібно внести деякі зміни в алгоритм.

### 6. Модифікація алгоритму

В першу чергу внесемо зміни в алгоритм надання рекомендацій, де замість стандартного обчислення схожості та пошуку схожих користувачів будемо використовувати нечітку кластеризацію.

Також, потрібно модифікувати сам метод кластеризації об'єктів, шляхом перетворення його на не чіткий. Тобто замість стандартної відстані між об'єктами буде враховуватися відсоток належності поточного об'єкта до певного іншого. Для цього потрібно замінити початкову матрицю відстаней, на матрицю схожості.

Мінусом, є трохи більша складність алгоритму, відповідно довше виконання, проте це компенсується на етапі реалізації інформаційної системи. Тобто ми зможемо виконувати кластеризацію в бекграунді

непомітно для користувача, і він не буде змушений чекати підготовки пропозицій.

Також перевагою є те, що враховуватися буде думка не лише «схожих» користувачів, а всіх користувачів системи, яка звісно ж домножується на коефіцієнт близькості.

Також вагомим мінусом, даної системи можна назвати сильну прив'язку до соціальної мереж. Проте сьогодні складно уявити людину, яка не користується останніми.

### 7. Розробка алгоритму розв'язання

Нижче наведено модифікацію алгоритму кластеризації:

Крок 1. ПЕРЕДАТИ в метод матрицю відстаней, що обраховані за один зі стандартних методів.

Крок 2. ПЕРЕТВОРИТИ матрицю відстаней, у матрицю схожості між об'єктами.

Крок 3. СТОРИТИ матрицю належності об'єктів кластерам

Крок 4. ПЕРЕЙТИ до ітерації.

Крок 5. ОБЧИСЛИТИ матрицю «відповідальності».

Крок 6 . ОБЧИСЛИТИ матрицю «доступності».

Крок 7. ОБРАТИ центри та обрати значення

Крок 8. ЗАПИСАТИ значення з матриці схожості в матрицю належності для центрів, що змінилися або додалися.

Крок 9. ЯКЩО остання ітерація перейти до (9), ІНАШКЕ до (5).

Крок 10. ЗБЕРЕГТИ центри та матрицю схожості.

На цьому робота алгоритму завершується.

Алгоритм колаборативної фільтрації не наводиться, адже він змінився не суттєво.

Зміна полягає лише в тому, що використовують кластери, як узагальнені користувачі замість всіх користувачів.

### 8. Висновки

На прикладі даних про користувача, які можуть бути отримані з соціальної мережі facebook описано засіб для підготовки цільових рекомендацій користувачам. Для розробки засобу використані методи нечіткої кластеризації та колаборативної фільтрації. Був приведений аналіз існуючих методів розв'язання задачі та описаний модифікований метод кластеризації розповсюдження близькості, який був використаний для надання рекомендацій. Визначено переваги та недоліки даного алгоритму.

### Список літератури

1. Brendan J. Frey and Delbert Dueck // Clustering by Passing Messages Between Data Points, Science Feb. – 2007.
2. Omar M. al-akash, Sharifah Sakinah Syed Ahmad and Mohd Sanusi Azmi// Fuzzy Distance Measure Based Affinity Propagation Clustering // International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 2 (2018) pp. 1501-1505.
3. А.А.Барсегян, М.С.Куприянов: Методы и модели анализа данных: OLAP и Data Mining // СПб.:БХВ-Петербург – 2004 – 336с. – ст.162.
4. Sneha Y. S. and Dr. G. Mahadevan// A Study on Clustering Techniques in Recommender Systems, International Conference on Computational Techniques and Artificial Intelligence – 2011.
5. Kyoung-jae Kim and HyunchulAhn// A recommender system using GA K-means clustering in an online shopping market, Expert Systems with Applications, Volume 34, Issue 2, February 2008, Pages 1200-1209.
6. Manh Cuong Pham, Yiwei Cao, Ralf Klamma, Matthias Jarke// A Clustering Approach for Collaborative Filtering Recommendation Using Social Network Analysis, Journal of Universal Computer Science, vol. 17, no. 4 (2011), pp. 583-604.

## ІНФОРМАЦІЙНА СИСТЕМА ФОРМУВАННЯ РЕКЛАМНОГО ПАКЕТУ З УРАХУВАННЯМ ІНДИВІДУАЛЬНИХ ХАРАКТЕРИСТИК КОРИСТУВАЧА

В статті описано спосіб підбору рекомендацій на основі колаборативної фільтрації. Коротко описано метод колаборативної фільтрації. Було описано постановку задачі та опис рішення на прикладі системи рекомендації інтернет-реклами.

This article describes way of providing recommendations, based on collaborative filtering method. Short explanation of collaborative filtering is provided. Main subject and explanation of resolving were described.

**Ключові слова:** рекомендації, колаборативна фільтрація, кластеризація, відбір, інтернет-реклама, відстань Левенштейна.

**Keywords:** recommendations, collaborative filtration, clusterisation, choosing, web-advertising, Levenstein range.

### 1. Вступ

В сучасному світі рекомендаційні системи стали дуже популярними для різноманітних сервісів, оскільки вони дозволяють завжди надавати користувачу певну інформацію, на пошуки якої в користувача може уходити певний час. Найяскравішим прикладом є потокові музичні сервіси, які звертають увагу на вподобання користувача та регулярно формують списки відтворення, які можуть бути цікавими окремому користувачу. Дана система формує такі списки для кожного користувача, що дозволяє системі бути різносторонньою, не поширюючи між усіма однакові підбірки музики.

Також рекомендаційні системи стали дуже популярними в сфері інтернет-реклами, оскільки велика кількість веб-сайтів змогла фінансово підтримувати свою діяльність за допомогою вбудованої реклами. Певний сайт може запропонувати зробити свою веб-сторінку платформою для розміщення реклами, тим самим отримувати гроші від рекламодавців, які витратити на власні потреби. В наш час оборот грошей у рекламному бізнесі перевищує декілька мільярдів, тому цей бізнес активно розвивається і інтегрується в різні медіа. А саме рекомендаційні системи допомагають зробити процес відображення реклами не нав'язливим, відображаючи користувачу тільки ту рекламу, яка є для нього актуальною в певний момент часу.

Отже, узагальнюючи, можна сказати, що головною задачею рекомендаційної системи є

аналіз певного суб'єкта з метою створення його найбільш точного опису, та формування під цей опис певного пакету об'єктів(реклами, музики, книг і т.д).

### 2. Постановка задачі

Головною задачею рекомендаційної системи є знайти такий об'єкт, який буде максимально підходити під опис суб'єкта(користувача).

Загалом цю задачу можна описати так:

$$r_{u,i} = \max_{i \in I} S(u, i), \text{ де } u \in U \quad (1)$$

Де  $S$  – функція, яка відображає релевантність певного користувача для певного елемента системи,  $U$  – множина усіх користувачів системи,  $u$  – певний користувач, який розглядається,  $I$  – множина всіх елементів системи(у даному випадку тем),  $i$  – певна тема, яка розглядається для певного користувача.

Тобто, якщо коротко, головною ціллю задачі являється знайти такий набір(у нашому випадку тем)  $i$ , які будуть максимально релевантними для користувача  $u$ .

Зазвичай рекомендаційні системи реалізують одну з трьох типових задач:

- Item-based фільтрація(тобто фільтрація, яка опирається на об'єкти);
- User-based фільтрація(тобто фільтрація, яка опирається на суб'єктів, тобто користувачів системи);
- Гібридна фільтрація.

Зазвичай рекомендаційні системи використовують гібридну фільтрацію, але ми розглянемо кожену.

### 3. Item-based фільтрація

Даний метод підбору рекомендацій опирається на підбір рекомендацій за допомогою об'єктів, тобто знаходження схожих об'єктів до тих, які уже позначені користувачем, як оцінені. Для цього потрібно слідкувати за користувачем, щоб мати повну картину того, що йому може бути цікаво.

Перш за все треба розглянути матрицю корисності (від англ. Utility matrix). У рекомендаційних системах зазвичай є дві одиниці, на які звертається увага – це користувачі та одиниці даних. В матриці зображуються одиниці та користувачі, де на стиках показано рейтинг зацікавленості користувача в тій чи іншій одиниці.

Задача зводиться до знаходження кута між двома  $n$ -вимірними векторами, одним з яких являється сам користувач, а іншим є один з об'єктів:

$$S(a, b) = \cos(\langle a, b \rangle) = \frac{r_a * r_b}{\|r_a\| * \|r_b\|} \\ = \sum_i \frac{r_{ai} * r_{bi}}{\sqrt{\sum_i r_{ai}^2} \sqrt{\sum_i r_{bi}^2}} \quad (1)$$

Де  $S(a, b)$  – ступінь схожості двох користувачів,  $S$  – функція яка обчислює схожість,  $a$  – аргумент користувача А,  $b$  – аргумент користувача В.

### 4. User-based фільтрація

Даний тип фільтрації також називають колаборативною фільтрацією. Перед тим як розглянути сам алгоритм, потрібно розглянути матрицю корисності, оскільки вона являється головним джерелом даних при використанні колаборативної фільтрації.

Таблиця 1: приклад матриці корисності

Користувач	Од1	Од2	Од3
1	×	×	R1,3
2	R2,1	R2,2	×
3	×	R3,2	×

Де строки відповідають кожному з трьох користувачів, Од1, Од2 та Од3 відповідають трьом одиницям у системі, які розглядаються.  $R(n, m)$  означає рейтинг одиниці  $m$  для користувача  $n$ , тобто його зацікавленість у даній одиниці, яка обчислюється самою системою.

Ми можемо передбачити оцінку користувача а для теми і якщо візьмемо зважену суму його рейтингів з усіх користувачів (позначимо  $s$ ), де вага це ступінь схожості між кожним користувачем та користувачем а:

$$r_{i,j} = \bar{r}_{i,j} + \frac{\sum_n w(i,k)(v_{k,j} - \bar{v}_k)}{\sum_n |w(i,k)|} \quad (2)$$

Де  $r_{i,j}$  означає оцінку  $j$ , яка не була оцінена користувачем і (у Табл.1 такий випадок позначено як X). Для передбачення ми вибираємо середню оцінку одиниць для певного користувача та агрегуємо це значення. Це зроблено для того, щоб можна було компенсувати різницю в оцінках між двома користувачами. Наприклад, якщо один користувач має середню оцінку 6 а інший 8, то ця різниця і буде значенням, яке алгоритм повинен компенсувати. В цій формулі (2)  $w(i, k)$  означає коефіцієнт кореляції Пірсона, який показує лінійну залежність користувача  $i$  від користувача  $k$ . Ця величина лежить в межах  $-1 < w(i, k) < 1$ . Значення, яке найближче до 1, являється позитивною кореляцією, а яке наближається до -1 – негативною. При множенні результату одного користувача на

### 5. Вдосконалення алгоритму

Потрібно модифікувати алгоритм для покращення рекомендацій.

Даний підхід до вирішення задачі має проблему так званого ‘холодного’ старту, яка виражається в недостатній кількості даних при першому використанні. Для вирішення цієї проблеми треба брати такі дані від користувача, які можна дістати з першого використання, а саме його опис та розширення взаємодії з системою, тобто з витягувати з кожної дії користувача максимальну кількість даних. Це буде реалізуватися за допомогою ключових слів.

Прикладом буде система підбору інтернет-реклами.

Робота системи буде поділена на 3 кроки:

#### Крок 1: Витягування даних

Дані беруться з будь-якої взаємодії користувача з системою, будь то клік по певному елементу (до якого прив'язані певні ключові слова) або час, проведений на певній сторінці. Всі ці ключові слова будуть вноситися у матрицю корисності.

#### Крок 2: Знаходження найближчих сусідів

В цьому кроці головним є створити з елементів вектори та сформувані найближчих сусідів. В нашому випадку цими ключовими словами будуть описи тем, на які можуть бути дані рекомендації. Ключові слова використовуються з кроку 1. Нам потрібно їх кластеризувати щоб не втрачати слова, які можуть писатися по-різному але означати одне

й те саме та виділити найпопулярніші слова щоб можна було звернути увагу на вподобання користувача.

Для кластеризації слів буде використана відстань Левенштайна. Вона допоможе об'єднати спільнокореневі слова в одне, щоб уникнути непотрібної роздрібності у матриці корисності. Для цього будуть створені групи ключових слів з найбільшим коефіцієнтом схожості

### **Крок 3: Аналіз та формування рекомендації**

Найближчі сусіди формуються на основі відібраних ключових слів та алгоритм відбирає декілька  $N$  найближчих сусідів за допомогою порогової кластеризації. Далі формується матриця з тем, які були відібрані та косинус подібності, за допомогою якого можна уже робити рекомендації.

В таблиці 2 наведено результат при використанні модифікованого алгоритму та при використанні колаборативної фільтрації(CF). Мірою будемо використовувати точність та повноту рекомендацій, наданих користувачу.

Таблиця 2: Результат експерименту

Метод	Точність	Повнота
CF	0.64334	0.4345
Даний метод	0.6913	0.4889

Як видно з результатів експерименту, використання кластеризації при обробці ключових слів(які використовуються замість оцінок, які стандартні для колаборативної фільтрації), помітно покращення рекомендацій через більш точну систему оцінювання та пришвидшення так званого 'холодного старту' через більш активну збірку інформації про користувача, хоча при першому відвідуванні, рекомендації все одно будуть максимально підібрані під користувача через використання його характеристик(географічне положення, вік, стать і т.д.).

## **Список літератури**

1. G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item- to-item collaborative filtering," *Internet Computing* 7:1, pp. 76–80, 2003.
2. G. Adomavicius and A. Tuzhilin, "Towards the next generation of rec- ommender systems: a survey of the state-of-the-art and possible exten- sions," *IEEE Trans. on Data and Knowledge Engineering* 17:6, pp. 734– 749, 2005.
3. Balabanovic M., Y. Shohan, 1997. Fab: Content-based, collaborative recommendation. *Commun. ACM.*,40: 66-72.
4. Resnik P., H.R. Varian, 1997. Recommendation systems. *Commun. ACM.*,40:56-58.
5. Mulvenna. M.D., S.S. Anand and A.G. Buchner, 2000. Personalization on the net using web mining. *Commun. ACM.*,43: 123-125
6. C.C. Aggarwal *Horting Hatches an Egg: A New Graph-Theoretic Approach to Collaborative Filtering* / K-L. Wu, C.C. Aggarwal, , P.S. Yu, J.L. Wolf, // *Proc. Fifth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining.* – 1999.

УДК 004.021

*ПЕНСКОЙ В. В.  
ФІНОГЕНОВ О.Д.*

## ОГЛЯД ЗМАГАНЬ ТА СИМУЛЯТОРІВ ФУТБОЛУ РОБОТІВ

Робота присвячена розвитку робототехніки, а саме змаганням з футболу роботів. Розглянуто відмінності у правилах основних міжнародних ініціатив RoboCup та FIRA. Дано короткий огляд симуляційних ліг.

**КЛЮЧОВІ СЛОВА** Футбол роботів, симуляція 2D, робототехніка, самоорганізуючі системи, багатоагентні системи, стратегічне прийняття рішень, штучний інтелект.

The work is devoted to the development of robotics, namely the competition of football robots. The differences in the rules of the main international initiatives of RoboCup and FIRA are considered. A brief overview of the simulation leagues is given.

**KEYWORDS** RoboCup, Simulation 2D, Robot Soccer, robotics, self-organizing system, multiagent systems, strategic decision-making, artificial intelligence.

### 1. Вступ

На сьогоднішній день робототехніка та штучний інтелект є передовими галузями наукового дослідження. Саме тому починаючи із 1997 року проводяться змагання футболу серед роботів. Такі турніри дозволяють вирішувати локальні задачі і обирати найкращі рішення, а також знаходити нові проблеми для дослідження. Змагання також об'єднують численні дослідницькі групи, що працюють над однією проблемою. Це сприяє обміну ідеями та досвідом. Одним з ефективних способів просування науки та інженерних досліджень є постановка складної довгострокової мети. Побудова робота, що грає в футбол, само по собі не викличе значних соціальних і економічних наслідків, але досягнення, безумовно, буде вважатися великим проривом у цій області. Тому кінцева мета ініціативи RoboCup сформульована так: "У середині XXI століття команда повністю автономних гуманоїдних роботів-футболістів повинна виграти у чемпіона світу з футболу (команди людей-футболістів) за правилами FIFA" [1].

### 2. Постановка проблеми

Стандартною проблемою, що рухала розвиток штучного інтелекту довгий час вважалися комп'ютерні шахи, що дозволили

розробити і оцінити алгоритми пошуку. Проте з розробкою комп'ютера Deep Blue, який обіграв гросмейстера Гаррі Каспарова, використовуючи офіційні правила, комп'ютерний шаховий виклик наблизився до мети. Однією з основних причин успіху комп'ютерних шахів в якості стандартної проблеми є те, що була чітко визначена оцінка прогресу. Прогрес дослідження можна оцінити як силу системи, про що свідчить її рейтинг. Однак, оскільки комп'ютерні шахи наблизилися до своєї мети, постала необхідність у новій стандартній проблемі для створення набору технологій для наступного покоління галузей [1]. Такою проблемою став футбол роботів, що дозволяє оцінити різні теорії, алгоритми та архітектури на новому рівні. Відмінності між проблемами комп'ютерних шахів і футболу роботів розглянуто у таблиці 1.

Комп'ютерні шахи є грою з повною інформацією, по-перше, оскільки гравці впливають на ігрову ситуацію дискретними ходами, порядок яких визначений правилами, і не залежить від таких параметрів як швидкість реакції гравців, по-друге, оскільки у будь-який момент гри усі гравці мають повну інформацію про стан гри, тобто про позиції і усі можливі ходи будь-якого з гравців. Футбол роботів відносять до ігор з неповною інформацією, оскільки кожний автономний агент має інформацію про ігрову

ситуацію лише у полі видимості своїх сенсорів.

**Табл. 1. Різниця між характеристиками проблем комп'ютерних шахів і футболу роботів.**

	Комп'ютерні шахи	Роботизований футбол
Середовище	Статичне	Динамічне
Зміна стану	Покрокове	У реальному часі
Доступність інформації	Повна	Часткова
Зчитування сенсорів	Символічне	Несимволічне
Контроль	Централізований	Розподілений

Роботизований футбол пропонує інтегроване завдання дослідження, яке охоплює широкі області штучного інтелекту та робототехніки. Наприклад: злиття даних сенсорів в режимі реального часу, реактивна поведінка, визначення стратегії, машинне навчання, планування в режимі реального часу, багатоагентні системи, визначення контексту, штучне бачення, стратегічне прийняття рішень, інтелектуальне управління роботами та багато іншого.

### 3. RoboCup

RoboCup - це міжнародний проект для просування штучного інтелекту, робототехніки та суміжних областей. Федерація RoboCup організовує з 1997 року міжнародні змагання з футболу роботів. Змагання RoboCup 2018 пройшли у Монреалі (Канада) і зібрали 35 країн-учасниць, близько 4000 учасників і 5000 роботів [1].

На даний час RoboCup складається з чотирьох основних змагань, кожне з яких має ряд ліг:

- 1) RoboCup Soccer;
  - a) Standard Platform League (SPL);
  - б) Small Size League (SSL);
  - в) Middle Size League (MSL);
  - г) Simulation League;
    - 2D Soccer Simulation;
    - 3D Soccer Simulation;
- 2) Humanoid League;
- 3) RoboCup Rescue;
- 4) RoboCupJunior.

Крім футбольних змагань, з 2001 року проводиться змагання для пошуку жертв стихійних лих та координації рятувальних

(RoboCupRescue) та конкурси для молодих дослідників (RoboCupJunior).

У симуляційній лізі команда взаємодія та навчання є найбільш розвиненою. У лізі колісних роботів основне завдання полягає у сприйнятті ситуації на полі (системи зору, сенсори відстані) та реалізація базових навичок у футболі (наближення, контроль, дриблінг та передача м'яча). У RoboCup MSL команди з п'яти повністю автономних роботів грають у футбол з футбольним м'ячем розміру FIFA. Команди можуть вільно розробляти власне обладнання, але усі сенсори повинні бути встановлені на роботі. Також існує обмеження на максимальний розмір і вагу. Основна увага в дослідженні приділяється розробці мехатроніки, управлінню і багатоагентній співпраці на рівні планування і сприйняття. У лізі гуманоїдів автономні роботи з людиноподібною структурою тіла і людиноподібними почуттями грають в футбол один проти одного. На відміну інших роботів завдання сприйняття і моделювання світу не спрощується за рахунок використання сенсорів, не характерних для людини. Додатково необхідно боротися із технічними викликами. Динамічний біг і удар м'яча при збереженні рівноваги, візуальне сприйняття м'яча, інших гравців і поля, самокалібрація і командна гра є одними з багатьох проблем дослідження, що виникають у лізі гуманоїдів. Збільшуючи продуктивність роботів, правила змагань стають більш вимогливими до зменшення відхилень від офіційних правил FIFA.

### 4. FIRA

FIRA складається з 7 ліг:

- 1) MiroSot (середній 5-vs-5, 11-vs-11);
- 2) RoboSot (великий 3-vs-3);
- 3) SimuroSot (симуляція 5-vs-5, 11-vs-11);
- 4) KheperaSot (автономний 1-vs-1);
- 5) AndroSot (контроль людиною 3-vs-3);
- 6) NaroSot (до 50 см);
- 7) HuroSot (з використанням робота-гуманоїда).

Конкурс RoboSot складається з ряду задач, а також традиційного футбольного змагання. Дослідження на платформах мобільних роботів отримують все більшу увагу, особливо в області побутової техніки. Проблеми та виклики, що стоять перед розробкою цих платформ різноманітні та



часто суттєво відрізняються від проблем, що постають як у менших, так і в гуманоїдних лігах [2].

AndroSot має на меті тренувати здібності атаки та захисту у роботах, а також складається з декількох завдань, таких як дриблінг, ухилення від перешкод, удар, виявлення траєкторій, роль голкіпера, рольові умови та контроль позиціонування.

SimuroSot є платформою імітації для оцінки своїх алгоритмів, які можуть бути використані для участі у лігах моделювання MiroSOT, HuroCup, RoboSot та FIRA Air.

### 5. Симуляційні ліги

Основними симуляційними лігами на сьогодні є RoboCup Soccer Simulation та FIRA SimuroSot. Змагання цих ліг відбуваються на симуляторах, що основані на клієнт-серверній взаємодії. Основною відмінністю є наявність тренера у симуляторі RoboCup та різна архітектура систем.

RoboCup Simulation league - одна з найстаріших ліг в RoboCupSoccer. Ліга симуляції зосереджена на штучному інтелекті і командній стратегії. Незалежно рухомі програмні гравці (агенти) грають у футбол на віртуальному полі, змодельованому на комп'ютері. Дана ліга включає 2 підкласи: 2D і 3D.

У 2D симуляційній лізі дві команди з одинадцяти автономних агентів грають у футбол на двовимірному віртуальному стадіоні, представленому центральним сервером під назвою SoccerServer. Цей сервер знає все про гру, тобто поточну позицію всіх гравців і м'яч, фізику і так далі. Гра також залежить від зв'язку між сервером і кожним агентом. З одного боку, кожен гравець отримує відносні дані своїх віртуальних сенсорів (візуальних, акустичних і фізичних) і, з іншого боку, може виконувати деякі основні команди (наприклад, поворот або удар), щоб впливати на його середовище.

Сервер є центральним процесором, який імітує всі рухи і параметри гравців і м'яча. Кожен клієнт грає роль гравця, в тому числі голкіпера, виконує свою індивідуальну операцію і стратегію команди. Сервер і клієнти взаємодіють через сокети UDP / IP. Клієнт використовує сокет UDP для відправки команд гравців на сервер і отримання інформації від сенсорів про навколишнє

середовище з сервера. Клієнт - це мозок гравця в іграх RoboCup 2D Simulation. У звичайній грі кожна команда має 11 гравців, і, отже, в лізі беруть участь 22 клієнта, що підключаються до сервера. Крім того, у кожній команді є один тренер, який організовує формування і стратегію команди.

Короткий огляд змін у правилах RoboCup Soccer Simulation league:

1998 р. - введено спеціального гравця голкіпера, який може виконувати додаткову дію ловити м'яч; введено правило положення поза грою, що завершило повне впровадження правил футболу FIFA. Введено витривалість, що накладає ресурсну модель на гравців. Для виконання дій потрібно витратити витривалість, модель реалізує короткострокові витрати з довгостроковими фазами відновлення, що заохочує розвиток стратегічного управління ресурсами.

1999 р. - запроваджено команду turn\_neck, що дозволяє гравцям змінювати напрям голови. Оскільки область видимості обмежена, така команда дозволяє гравцям збирати більше інформації про навколишнє середовище. Для заохочення адаптації у реальному часі було запроваджено тренера. Кожна команда може керувати одним агентом тренера. Тренер може спостерігати за цілим полем під час гри та надсилати поради.

2000 р. - збільшено силу удару м'яча, що прискорило темп гри і зробило змагання більш видовищними.

2001 р. - введено гетерогенних агентів та заміни. Тільки тренер може міняти гравців у неігровий період. Агент тренер повинен визначати розподіл гравців відповідно до стратегії своєї команди, оскільки фізичні здібності гетерогенного гравця створюються випадковим чином для кожної гри. Також було додано нове змагання - конкурс тренерів. У цьому змаганні агент тренера має можливість давати поради команді, яка намагається покращити свою продуктивність. Щоб забезпечити універсальність зв'язку, була введена стандартизована мова тренера. Правила цього змагання часто змінювалися, а у 2005 році - завданням тренера стало ідентифікувати стратегії опонента (opponent modeling) замість надання ефективних стратегій для власної команди.

2002 р. - для того, щоб представити нові виклики, пов'язані з спілкуванням, довжина

слухових повідомлень серед гравців була скорочена. Оскільки вдосконалення майстерності дриблінгу продовжували розвиватися, захисникам стало надто складно заблокувати гравця-дриблера. Нова команда tackle дозволила гравцям вибивати м'яч у діапазоні, ширшому за команду kick, яка наслідує ймовірність, що ґрунтується на положенні м'яча відносно гравця, що займається боротьбою, але ще кілька циклів викликає нерухомість.

2003 р. - додано режим автоматизованої серії пенальті. Якщо гра закінчилася у нічию і опція симуляції доступна, то відбувається автоматизована серія пенальті.

2008 р. - кількість гетерогенних гравців збільшено з 7 до 14. Крім того, було заборонено призначення стандартного типу гравця, за винятком голкіпера. Це означає, що команда повинна використовувати принаймні 10 гетерогенних гравців. Таким чином, проблема призначення ролі стала складнішою. Ефективний діапазон дії лову голкіпера зменшився. Цю зміну було введено в дію, щоб заохотити команди до більш ефективного позиціонування, а не тільки для воротаря, а також для оборонних гравців.

2009 р. - введено модель кількості витривалості. Ця модель обмежує загальну суму витривалості для кожного гравця під час гри. Гравці повинні керувати їхньою витривалістю більш ретельно, а тренер обирати час заміни гравця відповідно до ігрової ситуації і втоми гравців. Модель прискорення була розширена, щоб дозволити гравцям прискорювати своє тіло до 4 напрямків. Гравці можуть налаштувати свою позицію більш гнучко і швидко, використовуючи розширену модель. Але це вимагає більш складне планування всіх дій.

2010 р. - команду tackle було розширено додаванням навмисного порушення правил. Коли здійснюється навмисний фол, ймовірність успіху збільшується, проте гравець може бути покараний. Ймовірність виявлення фолу визначається для кожного гетерогенного типу гравця. Якщо автоматичний арбітр визначає навмисний та небезпечний фол, суддя штрафуватиме гравця, даючи жовту або червону картку. Також введено новий неоднорідний параметр, який впливає на рівень лову воротаря. Параметр

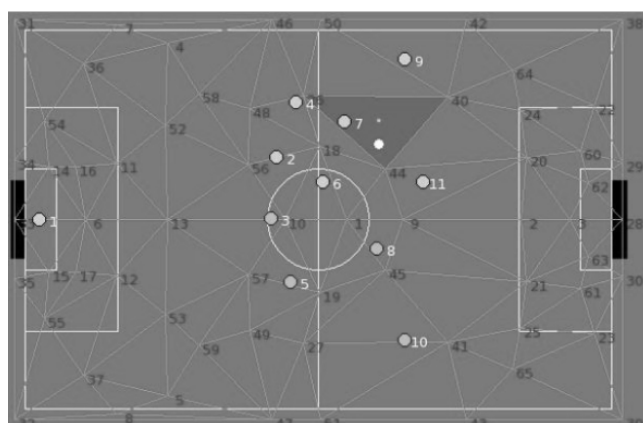
dash моделі змінився. Тепер гравці можуть прискорити своє тіло до 8 напрямків.

Симулятор FIRA, так само як і сервер RoboCup, є центральним процесором, який обробляє запити, а також відповіді від клієнтів, імітує невизначене середовище, а також відображає хід поточного матчу на моніторі комп'ютера.

До симулятора FIRA може підключитися лише 2 клієнти. Кожен клієнт представляє команду і визначає її стратегію. Симулятор FIRA призначений для розробки стратегії, яка потім може застосовуватися безпосередньо у лізі MiroSot. [3]

### 6. Аспекти моделювання футболу роботів

У 2018 році свою другу перемогу поспіль у чемпіонаті RoboCup у лізі симуляції 2D здобула команда HELIOS2018 студентів університету префектури Осака та університету Фукуоки (Японія). В останні роки команда зосередилась на покращенні планування послідовності дій за допомогою дерев пошуку. Було запропоновано механізм позиціонування на основі триангуляції Делоне та алгоритму лінійної інтерполяції [4]. Був створений фреймворк дерев пошуку, який генерує і виконує послідовності дій віртуальних футболістів. Дана структура шукає найкращу послідовність дій використовуючи генератор дій-кандидатів та функцію їх оцінки.



**Рис. 1. Демонстрація алгоритму позиціонування команди HELIOS2018**

Інша команда FCPortugal основну увагу приділила стратегічному позиціонуванню, основаному на ігрових ситуаціях [5]. Учасники даного дослідницького колективу створили бібліотеку та графічний інструмент для планування спільної поведінки між

агентами. Поняття, яке ввела FCPortugal, Setplay - це послідовності дій і стратегічні позиції гравців, які повинні бути виконані спільно з набором гравців для досягнення мети під час матчу, частіше після зупинки, але, можливо і в процесі гри. Setplay часто використовуються для кутових ударів і прямих вільних ударів.

Команда UvA Trilearn застосовувала підхід координаційних графів для кооперативних завдань (наприклад, передача м'яча, передбачення точок передачі м'яча). Також застосовується багатопланова ієрархія навичок (наприклад, передача, перехоплення). Даний метод поєднав прийняття рішення щодо передачі м'яча та отримання передачі [6].

### **Висновки**

У даній роботі розглянуто сучасний стан розвитку робототехніки у галузі футболу роботів. Описані основні міжнародні ініціативи та змагання, а також передумови їх виникнення. Подано огляд симуляційних ліг та правил змагань. Було досліджено системи симуляції футболу роботів. Розглянуто розвиток симуляційної ліги RoboCup та зміни правил відповідно до нових задач дослідження.

### **Список літератури**

1. RoboCup Federation. Official materials. [Електронний ресурс] // Режим доступу: <https://www.robocup.org>
2. Official website of the Federation of International Robot-football Associations [Електронний ресурс] // Режим доступу: <http://www.fira.net>
3. FIRA robot soccer [Електронний ресурс] // Режим доступу: <http://www.firaworldcup.org>
4. HELIOS2018: Team Description Paper / Tomoharu Nakashima , Hidehisa Akiyama , Yudai Suzuki , An Ohori , and Takuya Fukushima// Osaka Prefecture University, Osaka, Japan, 2018
5. Co-ordination in RoboCup's 2D Simulation League: Setplays as flexible, multi-robot plans./ Mota, L., Reis, L.P., Lau, N. //In: IEEE International Conference on Robotics, Automation and Mechatronics (RAM 2010), Singapore (2010)
6. RoboCup 2007: Robot Soccer World Cup XI // Ubbo Visser, Fernando Ribeiro, Takeshi Ohashi, Frank Dellaert / Springer, 2008, p.229

## СПОСІБ КЛАСТЕРИЗАЦІЇ КОРИСТУВАЧІВ СОЦІАЛЬНОЇ МЕРЕЖІ TWITTER

В статті розроблено спосіб класифікації користувачів соціальної мережі Twitter. Сформульовано задачу та її цілі з точки зору стратегій використання вихідних даних. Виділено основні фактори для проведення аналізу, а також методи, за допомогою яких можна здійснити кластеризацію користувачів на підгрупи. Описано алгоритм кластеризації та знаходження неявної тематики тексту.

This article discusses how to classify users of the social network Twitter. The task and its goals are formulated in terms of strategies for using the final ones data. The main factors for conducting the analysis are highlighted, as well as methods by which clustering of users into subgroups. An algorithm for clustering and finding implicit text topics is described.

**Ключові слова:** задача кластеризації, кластеризація, користувачі Twitter, соціальна мережа.

### 1. Вступ

Сьогодні Інтернет є основним каналом отримання або поширення інформації. Більшість людей використовує інтернет для того щоб проводити час у соціальних мережах. Кожна людина може відчувати себе справжнім репортером, і поділитись важливими новинами, або поширити свою думку, і знайти однодумців/послідовників. Так само можна бути і простим читачем, і дізнаватись нову інформацію від інших людей.

Серед всіх соціальних мереж, свою нішу займає Twitter. На початок 2018 року його аудиторія становила 336 мільйонів людей. Користувачі спілкуються короткими повідомленнями(до 280 символів) які називаються твітами. Це є дуже зручно, так як можна отримувати дуже великий об'єм інформації, витрачаючи на це не дуже велику кількість часу.

Якщо згадати вибори 2008 року у США, то можна побачити як соціальні мережі були використані у політичній боротьбі. Так, перемога Барака Обами, колишнього президента США, також завдячує соціальним мережам. Деякі з аналітиків, заявляли, що це була одна з вирішуючих частин його кампанії[1].

Дані твітів несуть дуже велику кількість інформації. Вони навіть передають настрої. Так у дослідженні було проведено спостереження, за змінами настрою у групі людей, проаналізувавши що саме писали ці люди у своїх блогах.

Але отримання інформації, чи нав'язування певній групі людей якоїсь ідеї, це не єдині способи у які можна використати ці дуже великі об'єми даних.

Так наприклад можна отримувати у реальному часі інформацію про якісь стихійні лиха, як наприклад повинь чи землетрус. Та за рахунок швидкого сповіщення кращу протидіяти їм. Адже саме за рахунок швидких твіттер-сповіщень, ми можемо про це дізнатись.

Тому доцільною є розробка технології для об'єднання однодумців на основі діяльності в соціальних мережах. Завдяки застосуванню методів кластеризації система буде мати можливість групувати користувачів зі схожими інтересами, що надасть можливість рекомендувати користувачам те, що входить в коло їх інтересів.

### 2. Предмет дослідження

Для того, щоб зібрати інформацію про користувачів Twitter-у, ми використовуємо API розробника Twitter.

Статистика показує такі основні дані про використання Twitter[2]:

- 326 млн активних користувачів кожного місяця;
- 500 млн твитов кожного дня (це приблизно 6 тис. постів в секунду);
- 83% світових лідерів завели свої власні акаунти;

- більше 67% користувачів схильні робити покупки тієї торгівельної марки, за якою вони спостерігають у Twitter.

Ці всі данні вказують на те що Twitter підходить для такого типу аналізу краще за інші соціальні мережі.

Також цікавою для аналізу є і статистика по способу використання самих твітів:

- звичайні твіти становлять лише 26%;
- відповіді на існуючі твіти становлять значно більше – 68%;
- звичайні ретвіти – 6%;
- 31% твітів містять у собі посилання на деякі ресурси;
- 55% твітів містять у собі медіа(картинки або анімації).

Важливо те що самі твіти є дуже короткими, і тому це примушує користувачів вживати основні слова які описують тематику. За допомогою цього ми можемо також аналізувати текстову складову.

### 3. Кластерний аналіз

Кластерний аналіз (англ. data clustering) – завдання розбиття об'єктів (ситуацій) на підмножини, що називаються кластерами, так, щоб кожен кластер складався зі схожих об'єктів, а об'єкти різних кластерів істотно відрізнялися[3].

Кластерний аналіз застосовується в різних сферах і галузях. Він працює навіть тоді, коли даних мало і не виконуються вимоги нормальності розподілу випадкових величин й інші вимоги класичних методів статистичного аналізу, корисний, коли потрібно класифікувати велику кількість інформації. Наприклад, у медицині кластеризація використовується для класифікації захворювань або їх симптомів, пацієнтів, препаратів тощо.

У маркетингу це можуть бути завдання сегментації конкурентів і споживачів.

У соціології розбивка респондентів на однорідні групи. У маркетингових дослідженнях кластерний аналіз застосовується як у теоретичних дослідженнях, так і маркетингологами, що практикують і вирішують проблеми угруповання різних об'єктів (аналіз поведінки споживача, групування споживачів в

однорідні класи для одержання максимально повної картини про поведінку клієнта з кожної групи й про фактори, що впливають на його поведінку). Кластерний аналіз добре зарекомендував себе у всіх сферах життєдіяльності людини.

Сучасний напрямок розвитку маркетингу є все більше спрямованим на отримання інформації про велику групу людей які цікавляться певними речима.

### 4. Латентно симантичний аналіз

ЛСА використовується для виявлення латентних (прихованих) асоціативно-семантичних зв'язків між термами (словами) шляхом скорочення факторного простору. Термами можуть виступати як слова, так і їх комбінації, документами — в ідеалі: набори тематично однорідних текстів, або просто об'ємний текст (кілька млн. слів), довільно розбитий на шматки, наприклад абзаци.

Основна ідея латентно-семантичного аналізу полягає в наступному: якщо у вихідному ймовірнісному просторі, що складається з векторів слів (вектором може бути речення, абзац, документ тощо), між двома будь-якими словами із двох різних векторів може не спостерігатися ніякої залежності, то після деякого алгебраїчного перетворення даного векторного простору ця залежність може з'явитися, причому величина цієї залежності буде визначати силу асоціативно-семантичного зв'язку між цими двома словами.[4]

Лексикон дітей в середньому щодня збільшується на 3-8 слів. При цьому, як кажуть лінгвісти, не завжди денотат має свій строго певний референт, або іншими словами — не кожне слово має співвідношення з реально існуючими речами або виконуваними діями (наприклад, абстрактні поняття, слова, що несуть неінформативну навантаження та ін.)

Виникає питання: як дитина визначає кожне нове значення слова і його співвідношення з іншими значеннями або чому утворюються нові «сенси» (денотати) і як вони співвідносяться між собою?

Роботу «смилових» механізмів концептуально можна порівняти з процесами категоризації або кластеризації.

ЛСА, його різновиди (PLSA, GLSA) і йому подібні (LDA — відоме латентне розміщення

Дирихле) дозволяє моделювати асоціативно-семантичні зв'язки між словами, що з одного боку дозволяє відмовитися від жорсткої прив'язки лексичної одиниці до певного з кластерів, а з – іншого представити цілісну систему зв'язків між словами.

Це означає, що слова в нашому мозку не класифіковані за поняттями (чи не лежать по полицках-кластерах), а утворюють складну систему зв'язків між собою, причому зв'язки ці можуть динамічно змінюватися залежно від безлічі причин: контексту, емоцій, знань про зовнішній світ і пр. А алгоритми, подібні ЛСА, дають нам можливість моделювати найпростіші елементи «розуміння». Але, сказати, що мозок працює за принципом ЛСА не можна і пояснити чому теж ні, бо в цьому немає необхідності: літаки теж літають, але крилами не махають. ЛСА — це лише один з методів, який дозволяє моделювати прості системи «мислення» для їх використання у практичних цілях (інтелектуальні системи).

### 5. Модифікація алгоритму

В роботі досліджено як можна виділити найкращий варіант оцінки для відстані між кластерами, маючи відкриту інформацію про користувачів Twitter.

Отже як оцінку відстані між елементами кластеру пропонується обрати певний індекс схожості, який побудований на таких критеріях:

- текстова схожість;
- схожість по використаним хештегам;
- схожість по ретвітам;
- схожість по людям яких користувач фоловить.

Сума всіх цих індексів має бути рівною одиниці, це є випадок коли користувач ідентичний.

Я використав кінцеву подібність користувача, а також класичні алгоритми кластеризації для кластеризації користувачів. Є два відомі алгоритми кластеризації: ієрархічні кластери[5] та k-means. Ієрархічний алгоритм занадто повільний для обробки великомасштабного набору даних, як, користувачів у Twitter, тоді як k-means не тільки ефективний, але і дуже швидкий в реалізації. Отже, я застосую k-means для кластеризації користувачів у Twitter. Невелика складність застосування k-means полягає у виборі параметра k, який визначає кількість кластерів і впливає на остаточну ефективність кластеризації.

Важливою частиною цієї роботи є автоматичне визначення тем, які цікавлять користувачів, на основі твітів, які вони опублікували, та обчислення тотожності тексту на основі цих тем. Щоб уникнути проблеми малого розміру одиночного твіта та недостатньої детальності даних, ми об'єднуємо твіти, опубліковані окремим користувачем, у великий документ. Таким чином, кожен документ суттєво відповідає користувачеві, і пошук тем, які цікавлять користувачів, просто означає пошук прихованих тем у цих документах. З цією метою застосовується модель Latency Dirichlet Allocation (LDA)[7], яка є методом автоматичного навчання, який не контролюється, для виявлення прихованих тем із великих колекцій документів.

### 6. Висновки

Отже, основна частина дослідження виконана. Наступними кроками буде знаходження  $k$  при якому кластеризація буде найбільш успішна, та провести серію експериментів для перевірки.

**Список літератури**

1. What is a Social Network Worth? Facebook and Vote Share in the 2008 Presidential Primaries [Електронний ресурс] // Режим доступу: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.471.7745&rep=rep1&type=pdf>
2. Twitter by the Numbers: Stats, Demographics [Електронний ресурс] // Режим доступу: <https://www.omnicoreagency.com/twitter-statistics/>
3. Leonard Kaufman, Peter J. Rousseeuw // Finding Groups in Data: An Introduction to Cluster Analysis, 2005
4. TK Landauer, PW Foltz, D Laham // Discourse processes, 1998
5. Метод ієрархічної кластеризації [Електронний ресурс] // Режим доступу: <http://mechanoid.kiev.ua/ml-lnwl.html>
6. Landauer T.K., Dumais S.T. A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge // Psychological Review. 1997. 104. – P.211-240.
7. Landauer T.K., Foltz P., Laham D. An Introduction to Latent Semantic Analysis. // Discourse Processes, 25, 1998 — P.259-284.

УДК 683.519

*РОМАНЧУК В. М.,  
ОЛІЙНИК Ю.О.***ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ДЛЯ ПОШУКУ АНОМАЛІЙ У ВІДЕОПОТОКАХ**

В даній статті розглянуті різноманітні методи пошуку аномалій у відеопотоках, та запропоноване покращення існуючої архітектури ЗНН, яка застосовується для пошуку аномалій у відеопотоках даних та порівняння запропонованої архітектури з архітектурою до покращення.

Крім того пропонується архітектурне рішення для побудови інформаційної технології, яка застосовуватиме покращену модель для обробки великої кількості відео шляхом використання технології Apache Spark для обробки надвеликих масивів інформації.

**КЛЮЧОВІ СЛОВА:** Спостереження, Знаходження аномалій, Аналіз Відео, Згорткові Нейронні Мережі, Глибоке Навчання

This article examines a variety of methods for finding anomalies in video streams, as well as proposing improvements to the existing CNN's architecture, which is used to search for anomalies in video data streams and to compare proposed architectures with architecture to improvement.

**KEYWORDS:** Closed Circuit TV, Surveillance, Anomaly Detection, Video Analytics, Convolutional Neural Network, Deep Learning

**1. Вступ**

Кількість даних в інтернеті зростає кожного дня. Зокрема одним з найпопулярніших видів контенту є медіа контент. Його кількість зростає щосекунди через широке використання смартфонів, які здатні робити знімки високої чіткості. За даними компанії Sandvine у 2015 році на YouTube щохвилини з'являлося 400 годин нового відео[1]. Велика кількість інформації потребує автоматизації обробки цієї інформації, що створює потребу в нових методах та підходах для її здійснення. Вчені та ентузіасти у сфері комп'ютерного бачення працюють над проблемами відеоаналізу вже не перший рік і змогли вирішити ряд проблем у цій сфері, зокрема розпізнавання незвичних подій[2], розпізнавання активності[3], знаходження аномалій[4]. Для кожної з перелічених задач були використані різні методи вирішення. Єдиний підхід для обробки відео був запропонований у [5] шляхом використання тривимірних згорткових нейронних мереж для визначення ключових характеристик вхідного відео (як часових так і просторових). Даний дескриптор пропонується використовувати шляхом переносу навчання (transfer learning) для інших задач аналізу потоків відео. Зокрема він згодом був використаний для розробки

нейронної мережі, яка класифікувала відео на основі 13 заданих категорій описаних в датасеті UCF Crime[6]. На вхід нейронна мережа приймала усереднені характеристики отримані від нейронної мережі, на виході отримувалася показник аномальності відео.

Основною задачею, яка вирішується є покращення існуючої архітектури ЗНН для її використання задля аналітики відео в реальному часі із застосуванням технологій обробки надвеликих масивів даних.

**2. Пов'язані дослідження**

Задачу знаходження аномалій на відео спеціалісти в сфері комп'ютерного зору вирішують вже не перший рік і змогли вирішити її для певних. Вузкоспеціалізованих її інтерпретацій. [7, 8, 9, 10, 11, 12, 13]. Наприклад є кілька спроб для визначення агресії та насильства [14, 15, 16, 17] на відео. Анкур Датта разом із співавторами визначати агресію шляхом використання руху та орієнтації кінцівок людей. Куї разом із співавторами [15] використали відео та аудіо інформацію для визначення агресивних дій на записах з відеоспостереження. Гао та співавтори[14] запропонували дескриптори для опису насильницьких дій, для їхнього знаходження на відео з натовпом. Нещодавно, Мохамаді разом зі співавторами у [17] запропонували



новий підхід заснований на евристиці поведінки для класифікації відео які містять чи не містять насильницьких дій.

Окрім розпізнавання шаблонів агресивної та неагресивної поведінки автори у [8, 9] пропонують використовувати стеження для моделювання нормального руху людей та вважати відхилення від нормального руху за аномалію. Проте цей підхід пов'язаний із складністю знаходження чітких траєкторій руху, оскільки вони можуть бути нечіткими чи перекриватися у випадку відслідковування декількох об'єктів. Крім того цей підхід вимагає значних обчислювальних ресурсів. Для уникнення цих проблем було запропоновано декілька інших методів, - вивчати загальні шаблони руху застосовуючи методи на основі гістограм [10], моделювання тем [11], прихованих моделей маркова на просторово-часових просторах [12]. Ці методи на основі тренувальної вибірки, яка складається з відео на яких знята нормальна поведінка вивчають розподіл нормального руху і сприймають за аномалії поведінку, яка має низьку ймовірність настання.

Слідом за успіхом словникового навчання та розрідженого представлення деякі дослідники у [21] застосували розріджене представлення для навчання словника нормальної поведінки. Під час тестування шаблони, які мали велику помилку реконструкції розглядалися як аномальні. Завдяки успішній демонстрації глибокого навчання для класифікації зображень, їх спробували застосувати для класифікації дій на відео [19, 20]. Проте проблема отримати мітки для тренувальної вибірки все ще залишається і потребує великих затрат часу у випадку відео.

Також [18, 7] використовували автоматичні кодувальники, які базуються на глибокому навчанні для того, щоб натренувати модель нормальної поведінки та застосували помилку реконструкції для знаходження аномалій. Підхід, який ми плануємо покращувати використовує як аномальні так і нормальні дані.

### 3. Постановка задачі

Для вхідного відео потрібно визначити відрізок протягом якого відбувається аномальна подія. Для цього було запропонований підхід з використанням багатопримірникового навчання.

Наведу спрощений опис методу навчання моделі запропонованої Ченом та співавторами. Нехай маємо два відео одне аномальне інше без аномалій. Ми знаємо, що в аномальному відео присутня аномалія та не знаємо її часових рамок. Розділяємо кожне відео на 32 часових сегменти, кожне відео в результаті утворить собою множину у якій 32 аномальних сегменти та 32 сегменти з відсутньою аномалією. Кожну з двох утворених множин надалі називатимемо рюкзаком. Для сегментів з кожного рюкзака розраховуються характеристики, з використанням С3D. Далі для кожного сегмента розраховується показник аномальності, він визначається за рахунок повністю з'єднаної нейронної мережі. Функцією штрафу для мережі в даному випадку є кусочно-лінійна функція втрат, задана як:

$$l(B_a, B_n) = \max(0, 1 - \max_{i \in B_a} f(V_a^i) + \max_{i \in B_n} f(V_n^i)) - f(V_a^i) -$$

показник аномальності сегменту (0 ~ 1) який генерує нейронна мережа.

В якості алгоритму для оптимізації моделі застосовувався Adagrad.

### 4. Пропонований підхід покращення

Оскільки для тренування моделі застосовувався оптимізації застосовувався алгоритм Adagrad, Заради пришвидшення роботи моделі та зменшення помилки тренування/валідації застосувати оптимізаційний алгоритм Adam або комбінацію Adam + SGD як запропоновано у [22]. Як альтернативний варіант для покращення додамо ще один проміжний шар в мережу розмірністю 256. Як показав приклад мережі С3D глибокі нейронні мережі працюють значно краще для обробки графічної інформації.

### Висновки.

У ході дослідження був проаналізований сучасний стан у сфері пошуку аномалій у відеопотоках даних. Найкращі результати показав метод із використанням згорткових мереж, він дозволив побудувати модель, яка здатна здійснювати пошук аномалій не тільки для класів для яких вона була натренована, але й для класів аномалій, які вона зустрічає вперше. Ми пропонуємо підходи для покращення методу для покращення точності методу та збільшення швидкості розрахунку показника аномальності сегменту відео шляхом застосування іншого алгоритму Adam для оптимізації моделі. Крім того комбінування моделі та технології Apache Spark Streaming дозволить обробляти декілька відеопотоків в реальному часі.

### Список літератури

1. Analyzing Video, the Biggest Data of Them All [Електронний ресурс] – Режим доступу до ресурсу: <https://www.datanami.com/2016/05/26/analyzing-video-biggest-data>.
2. I. Laptev and T. Lindeberg. Space-time interest points. In ICCV, 2003.
3. O. Boiman and M. Irani. Detecting irregularities in images and in video. IJCV, 2007.
4. D. B.Kris M. Kitani, Brian D. Ziebart and M. Hebert. Activity forecasting. In ECCV, 2012
5. Waqas Sultani, Chen Chen, Mubarak Shah, "Real-world Anomaly Detection in Surveillance Videos" arXiv:1801.04264, 2018.
6. UCF Crime [Електронний ресурс] – Режим доступу до ресурсу: <https://webpages.uncc.edu/cchen62/dataset.html>.
7. D. Xu, E. Ricci, Y. Yan, J. Song, and N. Sebe. Learning deep representations of appearance and motion for anomalous event detection. BMVC, 2015.
8. S. Wu, B. E. Moore, and M. Shah. Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes. In CVPR, 2010.
9. A.Basharat, A.Gritai and M.Shah. Learning object motion patterns for anomaly detection and improved object detection. In CVPR, 2008.
10. X. Cui, Q. Liu, M. Gao, and D. N. Metaxas. Abnormal detection using interaction energy potentials. In CVPR, 2011.
11. T. Hospedales, S. Gong, and T. Xiang. A markov clustering topic model for mining behaviour in video. In ICCV, 2009.
12. L. Kratz and K. Nishino. Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In CVPR, 2009.
13. W. Li, V. Mahadevan, and N. Vasconcelos. Anomaly detection and localization in crowded scenes. TPAMI, 2014.
14. Y. Gao, H. Liu, X. Sun, C. Wang, and Y. Liu. Violence detection using oriented violent flows. Image and Vision Computing, 2016.
15. J.Kooij, M.Liem, J.Krijnders, T.Andringa, and D.Gavrila. Multi-modal human aggression detection. Computer Vision and Image Understanding, 2016.
16. A. Datta, M. Shah, and N. Da Vitoria Lobo. Person-on-person violence detection in video data. In ICPR, 2002.
17. S. Mohammadi, A. Perina, H. Kiani, and M. Vittorio. Angry crowds: Detecting violent events in videos. In ECCV, 2016.
18. M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis. Learning temporal regularity in video sequences. In CVPR, June 2016.
19. A.Karpathy, G.Toderici, S.Shetty, T.Leung, R.Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In CVPR, 2014.
20. D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In ICCV, 2015.
21. C.Lu, J.Shi, and J.Jia. Abnormal event detection at 150fps in matlab. In ICCV, 2013.
22. Keskar N.S., Soher R. Improving Generalization performance by switching from Adam to SGD. In ICCV, 2017

УДК 004.021

ГУНАВАРДАНА Р.С.Д.  
СЕЛІН Ю.М.

## **СУЧАСНІ ПІДХОДИ ДО ВИРІШЕННЯ ЗАДАЧ РОЗМІЩЕННЯ НА ПРИКЛАДІ ОПТИМІЗАЦІЇ РОЗКЛАДУ РЕКЛАМНОГО КОНТЕНТУ**

У даному дослідженні описано алгоритм оптимізації розміщення медійного рекламного контенту для екранів та проекторів у людних місцях. Алгоритм зведений до задачі оптимізації певної сукупної характеристики «корисності» та врахування накладених обмежень. Інтелектуальність системи полягає в тому що, система збирає статистичні дані про поточний розмір аудиторії та адаптує свою поведінку для того щоб максимізувати корисність рекламного медійного контенту, який транслюється.

### **КЛЮЧОВІ СЛОВА**

алгоритм, теорія розкладів, інтелектуальні системи, оптимізація розміщення, планування, розклад, контент,

This research goal is the development of a multimedia advertisement placement optimization algorithm specifically for screens and multimedia projectors. Algorithm is converged to the optimization problem, where custom property “usefulness” serves as optimization criteria. There are few limitations. The intellectual property of the system comes from the ability to adapt multimedia content placement based on statistical data about the current people count. It allows to maximize the “usefulness” of the played multimedia advertisement content.

### **KEYWORDS**

algorithm, job shop scheduling, intellectual systems, optimization, placement, planning, schedule, content,

### **1. Вступ**

У сучасному світі досить гостро стоїть проблема рекламування своєї продукції. Це обумовлено, по-перше, великою кількістю конкурентів на ринках, по-друге, більшим асортиментом та доступністю рекламних рішень. Великий об'єм рекламного трафіку проходить через глобальну мережу Інтернет у мобільні додатки, веб-сторінки та програми прямо на машини кінцевих користувачів. Проте, не зважаючи на безсумнівний успіх Інтернет-реклами, вона має свої недоліки. По-перше, на даний момент майже половина Земної кулі не використовує Інтернет у своєму повсякденному житті, по-друге, у певних географічних локаціях досі є проблеми з доступом до Інтернету – наприклад, у метро, по-третє, в останні роки було винайдено багато способів блокувати небажану рекламу – браузерні додатки, що блокують рекламу, VPN, сервери-медіатори,

що фільтрують веб-сторінки від реклами. Ці недоліки та декілька інших змушують нас і досі використовувати офлайн-рекламу. Прикладом таких рекламних систем є бігборди, дошки оголошень, проектори та екрани з медійним рекламним контентом у людних місцях.

### **2. Постановка деталей до проблематики задачі, що розв'язується**

У даному дослідженні розроблено алгоритм оптимізації розміщення медійного рекламного контенту для екранів та проекторів у людних місцях. Такий вид дистрибуції рекламного контенту на даний момент є дуже поширеним у великих містах і використовується переважно у магазинах та ресторанах у центрі міста та громадському транспорті. Тобто, мультимедійний рекламний контент направлений переважно на багатолюдні громадські місця.

### 3. Сучасний стан проблеми

Кардинальні зміни та стрімкий розвиток технологій у сфері Machine Learning та Internet of Things надав можливість розв'язувати нові види задач при порівняно невеликих фінансових витратах. Серед них цікавою є задача пошуку об'єктів на зображенні. Ця можливість з'явилася як завдяки новітнім алгоритмам, які оптимізували вирішення задач Machine Learning, так і новим апаратним забезпеченням, що комплектується компактним, швидким та багатофункціональним системним програмним забезпеченням.

Існує два види стиснення документів – із втратами та без. Алгоритми стиснення із втратами більш продуктивні, але документи, які вони стискають, можуть втратити якість зображення. Алгоритми без втрат навпаки, зберігають початкову якість зображень, але не такі продуктивні.

Для вирішення завдань використовується ряд методів лінійного програмування, дискретного програмування, методи гілок і меж, мережевого планування і управління. Останнім часом особливий розвиток приймають наближені методи рішення, різко скорочують перебір варіантів, (метод Монте-Карло).

Складність таких задач можна проілюструвати прикладом: потрібно спланувати виготовлення чотирьох виробів, кожне з яких проходить обробку на кожному з п'яти верстатів. Існує 4! або майже 7962 тис. Різних варіантів обробки (послідовностей); деякі з них до того ж треба якось відсіяти, оскільки певні операції слід виконувати в заданому порядку. На практиці, зрозуміло, завдання ще набагато складніше. [1]

Простіше інших вирішуються так звані задачі одного верстата: пошук найкращої послідовності обробки на ньому певної кількості деталей (найкращою з точки зору мінімуму витрат на пролежування деталей до і після обробки, мінімуму часу затримки у видачі деталей у порівнянні з установленим терміном, мінімального обсягу незавершеного виробництва і т.п.).

Існує також ряд моделей планування роботи виробничої дільниці (методичну

основу для них дає модель Джонсона для п деталей і двох верстатів, але вона представляє лише теоретичний інтерес і малопридатна на практиці). Нарешті, урахування містить методи складання календарних планів роботи підприємств. Зазвичай завдання ставиться таким чином: скласти план виготовлення всіх виробів, в якому не порушувалися б технологічні обмеження, обмеження по потужності обладнання, а також терміни запуску і випуску продукції. [2]

### 4. Запропоноване вирішення проблеми

Алгоритм зведений до задачі оптимізації певної сукупної характеристики «корисності» та врахування накладених обмежень. Обмеженнями в даній системі виступають час, дисперсія показів мультимедійного рекламного контенту. На рисунку 1 зображено високорівнева блок-схема алгоритму вирішення проблеми розміщення медійного рекламного контенту. На рисунку 2 зображено апаратне забезпечення, що необхідне для повноцінного функціонування системи трансляції та розіщення мультимедійного рекламного контенту для людних місць.

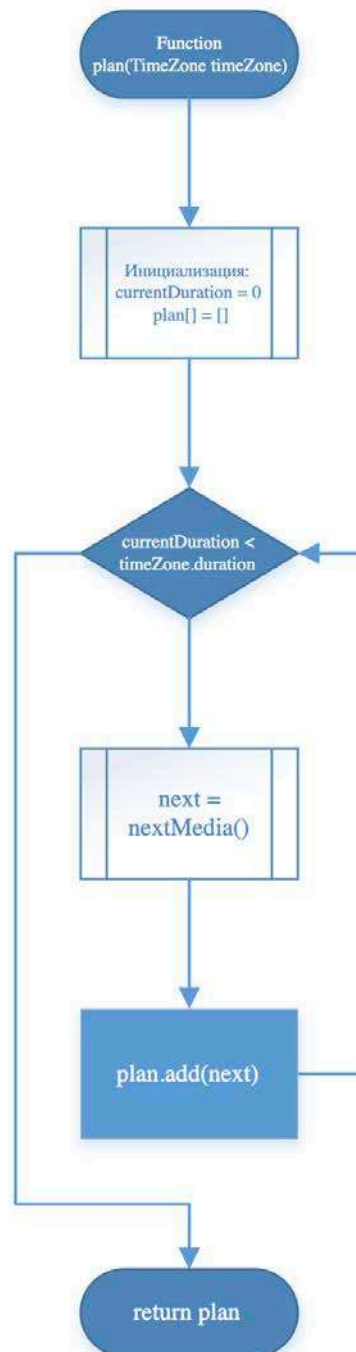
### 5. Адаптація розміщення під навколишнє середовище

У ході дослідження можливостей оптимізації розміщення медійного рекламного контенту було запропоновано використати технологію розпізнавання об'єктів для пошуку людей. Мається на увазі система, що використовує алгоритм Machine Learning для обробки зображення з камери у реальному часі та пошуку людей що знаходяться у людному місці. Головна характеристика яка може кардинально змінити підхід до розміщення медійного рекламного контенту – це кількість людей у місцях розміщення проекторів або екранів. Ця інформація за допомогою спеціального API може надходити до системи планування розміщення трансляцій медійного рекламного контенту для програвання на екранах або проекторах.

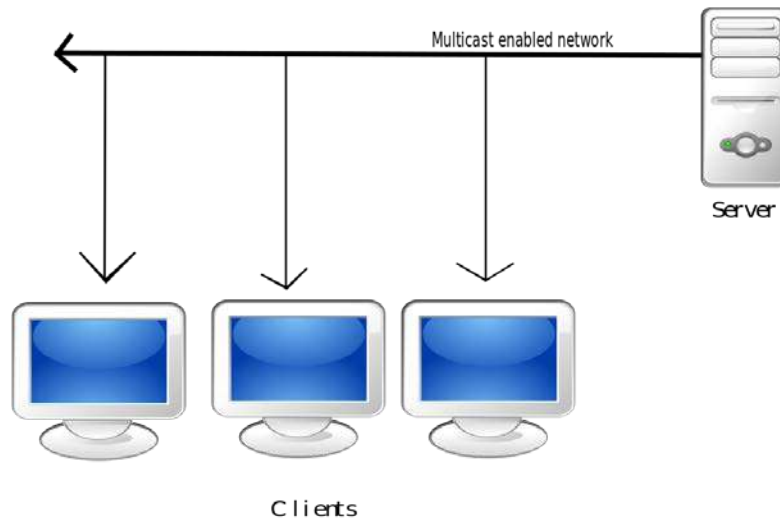
Інтелектуальність системи полягає в тому що, система збирає статистичні дані про поточний розмір аудиторії та адаптує свою поведінку для того щоб максимізувати

корисність рекламного медійного контенту, який програється. Під корисністю мається на увазі штучна характеристика, що моделює потенційну фінансову вигоду для

власника системи. Характеристика була введена, в якості функції, яку необхідно оптимізувати в ході роботи алгоритму.



*Рис.1. Високорівнева блок-схема системи мультимедійного рекламного контенту*



*Рис. 2. Апаратне забезпечення системи мультимедійного рекламного контенту*

### **Висновок**

В роботі запропонований алгоритм вирішення задачі оптимізації розміщення медійного рекламного контенту з урахуванням зміни цільової аудиторії. Програмне забезпечення, що використовує розроблений алгоритм, оптимізує корисність від медійного контенту, що транслюється на екрани та проектори у людних місцях. Для подальшої оптимізації запропоновано додати можливість адаптувати алгоритм, що виконується перед розміщенням рекламного медійного контенту. В якості параметрів адаптації системи обрано час та кількість людей. Алгоритм вирішує поставлену задачу.

### **Список літератури**

1. Теория расписаний – научная дисциплина [Електронний ресурс] / [Л. Костеляцький, К. Перепенко та ін.] – Режим доступу до ресурсу: [https://economic\\_mathematics.academic.ru/4526](https://economic_mathematics.academic.ru/4526)
2. Лазарев А. А. Теория расписаний. Задачи и алгоритмы. – 1997. – № 4. – С. 100-102.

УДК 683.519

ДЕМИДЕНКО М. О.

## ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС АНАЛИЗА УЯЗВИМОСТЕЙ УСТРОЙСТВ, НАХОДЯЩИХСЯ В ОДНОЙ ПОДСЕТИ С ДОСТУПОМ К СЕТИ

В настоящей статье рассмотрен программно-аппаратный комплекс, разработанный для выявления угроз, которые могут быть произведены на устройства посредством воздействия через сеть Интернет. Данный программно-аппаратный комплекс предназначен для устройств, которые находятся в пределах одной подсети. Описан архитектурный подход, с помощью которого производится защита, а также все шаги, которые могут быть применены для обнаружения, а также фильтрации вредоносного трафика.

КИБЕРБЕЗОПАСНОСТЬ, УЯЗВИМОСТЬ, ФАЙЕРВОЛ, ПРОГРАММНОЕ  
ОБЕСПЕЧЕНИЕ, ВЕБ-ЗАЩИТА

The subject of the article is a software and hardware complex developed to detect network vulnerabilities, which can be produced on devices through exposure via the Internet. Current software and hardware complex designed for devices that are within the same subnet. An architectural approach is described, with the help of which protection is performed, as well as all the steps that can be applied to detection and filtering malicious traffic.

CYBERSECURITY, VULNERABILITY, FIREWALL, SOFTWARE, WEB-  
PROTECTION

### 1. Вступление

Сегодня вопрос cybersecurity как никогда актуален. Bad rabbit, Petya, Not Petya заставили содрогнуться большую часть планеты. Была заблокирована большая часть данных устройств, которые были поражены. И в большинстве случаев не поддавалась восстановлению после нанесенного урона.

В настоящее время злоумышленники находят все больше уязвимостей в реализации программного обеспечения – будь то рабочий софт или, собственно, сами операционные системы - и пользуясь «дырками» ПО дестабилизируют работу системы, воруют приватные данные пользователей, вымогают выкупы за целостность данных. Стоит отметить, что основным

источником уязвимостей есть почитаемая всеми нами сеть Интернет.

Атаки на web-системы разумно разделить на атаки, направленные на программное обеспечение сервера, и атаки, направленные на web-приложение. Проанализировав bug-track, видно, что большинство уязвимостей именно в web-приложениях, такие как PHP-including, XSS, SQL-injection, DoS, DDoS и т.д. [1]

С развитием Internet of Things комфорт человека значительно повысился за счет автоматизации различных процессов, но так же повысился и фактор различных угроз безопасности, которым подвергается пользователь. Так как системы Internet of Things проводят сбор данных в реальном времени, то в случае наличия уязвимостей в такого рода устройствах это может привести к утечке

«sensitive» данных пользователей, например определенных показателей отдельного человека, фотографий и видео личного характера и так далее. Поэтому целью разработки является аппаратно-методологический комплекс, который выполняет роль firewall для устройств находящихся в одной подсети с доступом к Интернету, проводит анализ и фильтрацию не только входящего, но и исходящего трафика на пакетном уровне, распознаёт вредоносные пакеты и обезопасивает периметр каждого из устройств подсети, не пропуская «плохой ответ» (вредоносные пакеты) к конечному устройству и не выпуская «чувствительные» данные наружу.

## 2. Обзор литературы

Идея создания системы обнаружения вторжений возникла в 80-х годах, а в 90-х годах активно развивалась. В 1998 году начались такие известные проекты, как Snort и Prelude [2]. После этого разработчики начали внедрять многоуровневые системы безопасности [3]. Литературный анализ [3, 4, 5, 6, 7, 8] показывает, что одним из наиболее перспективных способов повышения уровня безопасности информационных систем является улучшение активных

средств защиты информации на основе атак систем обнаружения (злоумышленники) - системы обнаружения вторжений (IDS) и системы предотвращения вторжений (IPS) - системы обнаружения атак. Сегодня существует около сотни таких систем. Они весьма разнообразны с точки зрения принципов работы и технологии, которые используются в них [4, 8]. Наиболее распространенной системой обнаружения атак на основе хоста является Intruder Alert (компания Symantec) и сеть / сети Cisco Secure Scanner (компания Cisco Systems), а также некоммерческие системы для обнаружения атак ASAX (Университет Намюра, Бельгия), SHADOW (Naval Surface Warfare Center, Dahlgren Division) и NetSTAT (Калифорнийский университет в Санта-Барбаре [3, 4, 5, 6, 8, 9, 10, 11].

## 3. Инцидент безопасности

Имеется подсеть из независимых устройств и коммутатора (в нашем случае роутер). Каждый из устройств в произвольный момент времени делает запрос. Схема общения для отдельного устройства выглядит следующим образом (Рисунок 1):

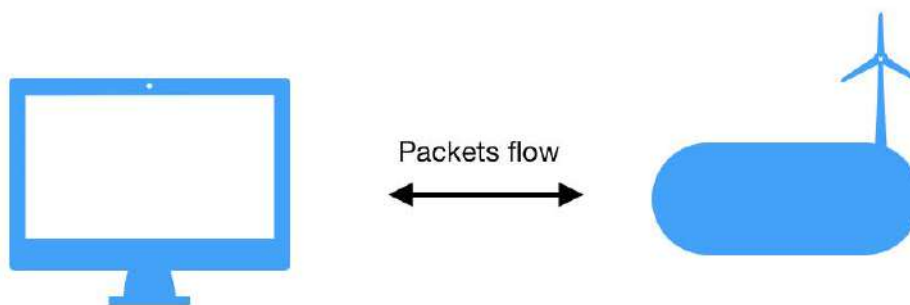


Рисунок 1. Схема взаимодействия отдельного устройства

В момент запроса начинается поток пакетов от устройства к роутеру. Роутер в свою очередь передает запрос в сеть Интернет и ожидает ответа. При получении ответа роутер направляет ответ запрашивающему устройству. После этого устройство собирает набор пакетов, интерпретирует его в понятный для компьютера формат и после этого

выводит его пользователю. Процесс интерпретации начинается после получения последнего пакета данных в

ответ. Если была произведена атака, то определенные пакеты либо были заранее умышленно фальсифицированы, либо подменены в процессе передачи. В идеальном случае firewall операционной системы распознает атаку и блокирует её.



Если firewall не обнаружил угрозу, которая прошла вместе с ответом от сервера, то устройство инфицировано и ответственность за дальнейшее выявление последствий атаки лежит на антивирусе, который при анализе состояния системы либо выявит, либо не выявит данную атаку.

#### 4. Процесс реагирования на инциденты и решения проблемы кибербезопасности

Разрабатываемое решение – это аппаратно-методологический комплекс (далее EMDS extendable malware detection system) – аппаратный, потому

что алгоритм имплементирован на отдельном устройстве, – подключив его к подсети, в которой находятся устройства, у пользователя будет возможность активации защиты как для каждого из них по отдельности, так и одновременной защиты всех устройств. При активации защиты производится атака man-in-the-middle на уровне маршрутизации трафика и весь трафик проходит через подключенное устройство. То есть в данном случае схема работы общения отдельного устройства выглядит следующим образом (Рисунок 2):

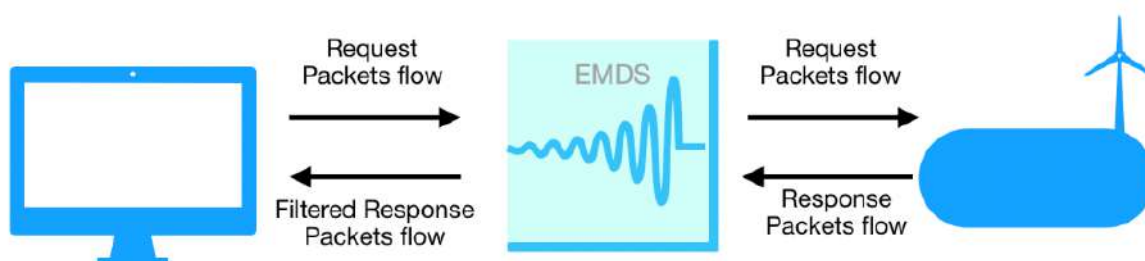


Рисунок 2. Схема взаимодействия отдельного устройства после внедрения EMDS

Теперь каждое из устройств считает, что EMDS - это роутер, а собственно сам роутер воспринимает EMDS как конечный девайс. В свою очередь EMDS берет на себя ответственность за доставку исходящих пакетов от устройства к роутеру и ответных пакетов от роутера к устройству. Так как теперь весь трафик проходит через EMDS, есть возможность анализировать каждый из пакетов. Для комплексного анализа нужно сохранять пакеты и их содержимое в базе данных, этот момент рассмотрим дальше.

Анализ пакетов будет проводиться с помощью трехфазовой проверки. На первом этапе проводится проверка на сигнатуры уже известных атак. Несмотря на стремительное развитие технологий, большая часть атак выявляется сигнатурным методом, и от того, насколько качественно пишутся сигнатуры, зависит точность работы всех методов, построенных на базе

сигнатурного анализа (в том числе машинное обучение).

Таким образом на данной фазе EMDS будет обнаруживать вторжение в реальном времени, то есть во-первых

будет выполнять роль Intrusion Detection System (IDS), во-вторых обеспечивать защиту от вторжений - Intrusion Prevention Sensor (IPS), а также проводить мониторинг сетевой безопасности - Network Security Monitor (NSM). Система отслеживает критические характеристики сети в реальном времени и формирует сигнал тревоги при обнаружении странного события, которое может свидетельствовать о наличии угрозы. Примеры таких характеристик включают объем трафика, использование полосы пропускания и использования протоколов. Проверка сетевого трафика проводится используя мощные и обширные правила и язык сигнатур, и имеется поддержка сценариев Lua для

обнаружения сложных угроз. Такой способ проверки позволяет интерпретировать ответ на EMDS и в случае выявления угрозы не пропустить её на конечное устройство.

Вторым этапом является поведенческий анализ. Этот метод базируется на последовательности различных запросов. Анализ поведения сети - это способность идентифицировать трафик, который является привычным в повседневном трафике сети. Иными словами, это попытка выявить нарушения в сети, если трафик превысил

настроенный ранее порог. Одним из наиболее известных нарушений сетевой безопасности является атака, известная как распределенный отказ в обслуживании (DDoS). Это серьезная угроза безопасности для интернет-провайдеров и крупных сетевых инфраструктур.

Рассмотрим данную фазу на примере. Пускай имеется последовательность запросов, приведенной в Таблице 1.

Таблица 1. Попытки SQL-инъекции

№	Request
1	/api/products?userId=1
2	/api/products?userId=3-2
3	/api/products?userId=- 1
4	/api/products?userId=1'
5	/api/products?userId=' 1
6	/api/products?userId=1 and sleep(5)

В данном случае наблюдается попытка выявления SQL-инъекции методом манипуляции характеристик, прибавления знака кавычки и вызова функции «sleep». Сами по для себя эти разрозненные симптомы не содержат очевидного вектора атаки, но их множество очевидно дает понять, собственно что преступник пробует «прощупать» веб-приложение. Математическая модель суммирует симптомы поведения злоумышленника за этап времени и на основании этих данных производится блокировка, собственно что разрешает не упустить начало атаки, при данном подходе нормальные запросы не блокируются.

Третьим этапом, как упоминалось ранее, служит эвристический анализ. Это модуль, который построен на нейронных сетях и имеет возможность отлавливать

атаки нулевого дня - то есть атаки, которые ранее не были замечены. Благодаря способности обучения, нейронные сети могут обучаться для выполнения различных инженерных задач. Некоторые из задач, которые могут быть идентифицированы, - это распознавание образов, ассоциация образцов, аппроксимация функций, системы управления. Первым преимуществом в использовании нейронной сети при обнаружении атаки будет гибкость, обеспечиваемая сетью. Нейронная сеть могла бы анализировать данные из сети, даже если данные являются неполными или неясными. Точно так же сеть будет обладать способностью проводить анализ с данными нелинейным образом. Кроме того, поскольку некоторые атаки могут быть предприняты против сети при

скоординированной атаке нескольких злоумышленников, особенно важна способность обрабатывать данные из ряда источников нелинейным образом.

Злоумышленное программное обеспечение либо проведенная атака всегда пытаются замести следы и максимально долго оставаться незаметными для совершения заложенных в них целей. Слабым местом является место логгирования пакетов. Иными словами, чтобы скрыть факт проникновения, вирус может пытаться поменять информацию о ранее зарегистрированных пакетах, чтобы ни сигнатурный метод, ни поведенческий анализ, ни машинное обучение не дало положительного результата в выявлении атаки.

С этой целью в качестве базы данных будет использоваться блокчейн, который будет гарантировать целостность и иммутабельность сохраняемых в нем данных. Так как запись постоянно проводится от имени одного и того же актера - пользователя, от имени которого запущено программное обеспечение по пакетному анализу, то это случай централизованного блокчейна с недоверительным центром (случай, когда вражеское ПО, попавшее на устройство после проведения атаки, используя текущую учетную запись на EMDS, пытается провести манипуляции по записи или изменению в базе данных). Данный подход организации хранения данных EMDS предостерегает от изменения уже хранимых данных в базе задним числом.

### 5. Заключение

В результате, данный аппаратно-методологический комплекс позволит изолировать устройство подсети от выявленной угрозы. Подводя итоги, можно выделить несколько кейсов, которые разрешаются в результате имплементации.

#### Случай 1. Блокирование выявленной атаки.

Проводится направленная атака на одно из устройств сети. В потоке пакетов

проходимых через EMDS на одной из трёх фаз выявлена угроза. Пакет, в котором была обнаружена атака, и последующие пакеты ответа блокируются. Таким образом ответ не будет интерпретирован полностью на конечном устройстве и атака не будет проведена.

#### Случай 2. Блокирование не выявленной атаки.

Проводится направленная атака на одно из устройств сети. В потоке пакетов проходимых через EMDS ни на одной из трёх фаз угроза не была выявлена. На последнем пакете ответа будет проводиться попытка интерпретации пакетов в понятный ответ и атака воспроизведется на EMDS, таким образом результата выполнения сервисом проверки не будет и EMDS не отправит ответ конечному устройству.

#### Случай 3. Попытка устройств IoT-системы «слить» данные в сеть Интернет.

Исходящие пакеты от устройств фильтруются с помощью устанавливаемых правил и не будут пропускать файлы определенных размеров заранее предустановленные администратором подсети. Это поможет избежать утечки данных личного характера, таких как аудио, видео, фото и так далее.

#### Случай 4. Попытка вредоносного ПО в процессе атаки скрыть факт проникновения.

Чтобы попытаться скрыть следы атаки, вредоносное ПО может пытаться манипулировать существующими данными о предыдущих пакетах в базе данных приложения, отвечающего за безопасность системы. Блокчейн гарантирует неизменяемость данных и их целостность, с помощью криптографических методов и электронно-цифровой подписи. Поэтому этот подход к хранению данных предотвращает возможность таких махинаций с данными

**СПИСОК ИСТОЧНИКОВ**

1. Смыков Г. Новый взгляд на обнаружение и предотвращение web-атак [Электронный ресурс] / Г. Смыков. – Режим доступа: <https://www.securitylab.ru/contest/290792.php>
2. Войтович О. П., Ювковецкий О. С. Класифікація вразливостей Web-ресурсів [Электронный ресурс] / О. П. Войтович, О. С. Ювковецкий. – Режим доступа: <http://itce.pu.if.ua/files/topics/Voytovych-Yuvkovetskyi.pdf>
3. Ленков С. В., Перегудов Д. А., Хорошко В. А. Методы и средства защиты информации. В 2 томах. Том 1. Несанкционированное получение информации [Текст] / С. В. Ленков, Д. А. Перегудов, В. А. Хорошко. – М. : Издательство: Арий, 2008 г. – 464 с.
4. Белова А. Л., Бородавкин Д. А. Сравнительный анализ систем обнаружения вторжений [Электронный ресурс] / А. Л. Белова, Д. А. Бородавкин. – Режим доступа: <https://cyberleninka.ru/article/n/sravnitelnyy-analiz-sistem-obnaruzheniya-vtorzheniy>
5. Корниенко А. А., Слюсаренко И. М. Системы и методы обнаружения вторжений: современное состояние и направления совершенствования [Электронный ресурс] / А. А. Корниенко, И. М. Слюсаренко. – Режим доступа: [http://citforum.ru/security/internet/ids\\_overview/](http://citforum.ru/security/internet/ids_overview/)
6. Лукацкий А.В. Обнаружение атак / А. В. Лукацкий. – СПб. : БХВ-Петербург, 2003. – 256 с.
7. Шаньгин В. Ф. Защита информации в компьютерных системах и сетях / В. Ф. Шаньгин. – М. : «ДМК пресс. Электронные книги», 2014. – 592 с.
8. Лукацкий А.В. Предотвращение сетевых атак: технологии и решения / А. В. Лукацкий. – СПб. : Экспресс Электроника, 2006. – 268с
9. Громов Ю.Ю. Информационная безопасность и защита информации: Учебное пособие / Ю.Ю. Громов, В.О. Драчев, О.Г. Иванова. – Ст. Оскол : ТНТ, 2010. – 384 с.
10. Жидков И. В., Кадушкин И. В. О признаках потенциально опасных событий в информационных системах / И. В. Жидков, И. В. Кадушкин. // Вопросы кибербезопасности. – 2014. – №1. – С. 40–48.
11. Норткатт С., Новак Д. Обнаружение вторжений в сеть / .С. Норткатт, Д. Новак. – М. : «Лори», 2002. – 416 с

УДК 510.8

*ДОВГАЛЬ Д.О.  
КОВАЛЮК Т.В.*

## КОГНІТИВНИЙ ПІДХІД В МОДЕЛЮВАННІ І УПРАВЛІННІ

В даній статті викладено огляд когнітивного підходу в моделюванні та управлінні. Наведено коротку історію розвитку когнітивного підходу і визначено клас проблем, які можливо і доцільно вирішувати, використовуючи когнітивний підхід.

Ключові слова: когнітивний метод, когнітивна карта, слабоструктуровані системи, орієнтований зважений знаковий граф.

In this article presents an overview of the cognitive approach to modeling and control. The brief history of the development of the cognitive approach is given and a class of problems is identified which may and should be dealt with using the cognitive approach.

Key words: cognitive method, cognitive map, weakly structured systems, oriented weighted sign graph.

### Вступ

Теорія прийняття рішень, що вивчає проблеми управління технічними, біологічними, соціально-економічними та іншими системами, за останні роки стала незалежною науковою галуззю. Теорія прийняття рішень активно використовує методи математики, психології, інформатики. Для аналізу, моделювання та управління слабоструктурованими системами та ситуаціями розроблений новий підхід сучасної теорії рішень – когнітивний підхід. На даний момент когнітивний підхід активно розвивається та використовується в міждисциплінарних науках (соціології, економіці тощо). Когнітивний підхід створений для розв'язання слабоструктурованих проблем, підвищення ефективності управління в організаційних, соціально-економічних та політичних системах.

Метою статті є визначення класу проблем, які можливо та доцільно вирішувати застосуванням когнітивного підходу.

### Історія еволюції когнітивного підходу

Термін «когнітивна карта» прийшов з психології. Когнітивна карта це суб'єктивне уявлення людини про організацію зовнішнього світу. Вивчення та аналіз когнітивних карт набули важливого значення в рамках вивчення

особливостей сприйняття людиною її середовища.

Когнітивна карта пов'язана зі збором, представленням, аналізом та обробкою інформації про навколишнє середовище. В даному процесі людина не тільки пасивно спостерігає, а й активно з ним взаємодіє. Початок роботи над дослідженням щодо когнітивного підходу в політології та соціології датується 1960-80 рр у роботах американського дослідника Р. Аксельрода та його колег з США та Скандинавії (Герадсвейн і Нарвесен [1978], Аксельрод [1976]). У політології та соціології когнітивне карта або «пізнавальна карта» є схематичним поданням зображення, що стосується конкретної проблемної ситуації.

Побудова та аналіз когнітивних карт дозволяють визначити причинну структуру думок, представлених у політичних текстах, і на вже на цьому робити висновок щодо авторського бачення політичної ситуації, визначаючи чинники, які враховують політики під час прийняття рішень. Аксельрод розробив метод когнітивного моделювання, який спирався на ідеї психологічної (Абельсон і Розенберг, [1958]), причинного висновку (Шапіро і Бонхем [1973]), а також графіку та теорії рішення (Аксельрод [1976]).

Головною проблемою прийняття рішень Аксельрод вважав проблеми пояснення ситуації та тестування гіпотези щодо ситуації, а також проблеми прогнозування і вибору рішень з декількох альтернатив. В складних ситуаціях людина, зазвичай, спрощує подання ситуації - пропускає зворотні зв'язки, не помічає вплив факторів тощо.

Так само, як будь-яка формалізація, когнітивна карта та методи її аналізу дають індивідуальний спосіб прийняття рішень у складних ситуаціях. Методи когнітивного моделювання знайшли застосування в колективній розробці з прийняття рішень (Харт [1976], Робертс [1976]).

При аналізі власних та інших когнітивних карт людина може з більшою точністю представити проблему, знаходити протиріччя, розуміти як інші представляють дану проблему. Також когнітивна карта є зручним способом для зміни стійких стереотипів, що сприяє формуванню нових точок зору.

Розвиток методів когнітивного моделювання обумовлений потребою в аналізі слабоструктурованих або неструктурованих систем і ситуацій, у тому числі великої кількості елементів різного характеру, а також взаємозв'язків між ними, які мають як кількісний, так і якісний характер. Поняття "погано структурована проблема (ситуація)" була введена Г. Саймоном. Когнітивний підхід до аналізу слабоструктурованих або неструктурованих ситуацій був запропонований Аксельродом та Робертсом. Аксельрод займався розробкою методології, а Робертс розробкою математичних методів.

В даний час когнітивний підхід знайшов застосування в різних сферах і в різних країнах та далі розвиваються.

#### **Проблеми, вирішення яких застосовує когнітивний підхід**

В науці традиційні теоретичні методи шукають оптимальні рішення з фіксованого набору альтернатив для

досягнення чіткої цілі. У реальних ситуаціях управління можна знайти проблеми, які полягають не тільки у виборі альтернативних рішень, а й у аналізі ситуації для виявлення реальних проблем та причин їх появи. При цьому погано підготовлені системи характеризуються проблемами, які важко або взагалі не можливо витягти з контрольної ситуації, яка аналізується. Такі проблеми обмежують застосування традиційних методів пошуку оптимального (або навіть задовільного) рішення для управління проблемами в таких систем.

Формалізація нечітких уявлень є однією з основних проблем, які потрібно вирішити при розробці моделей та методів прийняття рішень у слабоструктурованих ситуаціях. Також важливо враховувати, що суб'єкт контролю дуже часто повинен приймати рішення в умовах, що постійно змінюються та обмежені в часі. Також складність полягає в тому, що суб'єкт контролю повинен маніпулювати якісною інформацією у формі гіпотез (припущень), інтуїтивних концепцій та семантичних образів. Прийняття рішень в слабоструктурованих системах слід розглядати як складний груповий інтелектуальний процес вирішення проблем, який неможливо звести до виключно раціонального вибору

Когнітивний підхід в моделювання та управління слабоструктурованими системами спрямований на розробку формальних моделей та методів підтримки інтелектуального процесу вирішення проблем з урахуванням людського фактору та знань, накопичених і розвинутих в когнітивній науці.

#### **Основні концепції та моделі сучасного когнітивного моделювання**

В задачах аналізу і управління слабоструктурованими системами і ситуаціями когнітивне моделювання – це дослідження роботи та створення слабоструктурованої системи або ситуації шляхом побудови моделі даної системи або ситуації на основі когнітивної карти.

Когнітивна карта відображає індивідуальні або колективні суб'єктивні уявлення даної проблеми чи ситуації, пов'язаної з роботою та розвитком слабоструктурованої системи. Основними елементами когнітивної карти є фактори або концепти та причинно-наслідкові зв'язки між ними.

Когнітивна карта формально представляється у вигляді орієнтованого знакового графу. Орієнтований знаковий граф складається з вузлів (факторів) та ребер (взаємозв'язків) зі знаками «+» або «-». Частіше всього когнітивна карта представляється у вигляді орієнтованого зваженого знакового графу, де ребра відображають не тільки вплив (позитивний або негативний), а силу впливу одного фактору на інший. Тому можна стверджувати, що когнітивна карта у вигляді орієнтованого зваженого знакового графу є загальною у дослідженнях в контексті когнітивного підходу. Існують різні інтерпретації вузлів, ребер, ваг цих ребер, а також різні функції, що показують взаємозв'язки факторів. А це в свою чергу призвело до різних модифікацій когнітивних карт та інструментів для їх створення та аналізу. Виділяють 5 видів моделей на основі когнітивних карт:

- карти, які оцінюють увагу, асоціацію та значення поняття;
- карти, які показують розмір категорій та когнітивні таксономії;
- карти, що показують вплив, причинність та динаміку системи;
- карти, що показують структуру аргументів та висновків;
- карти, що визначають рамки та методи сприйняття.

Карты, що показують вплив, причинність та динаміку системи є найдоцільнішими для вивчення слабоструктурованих систем. Подання слабоструктурованих або неструктурованих систем та ситуацій у вигляді множини факторів або концептів та взаємозв'язків між ними обумовлений тим, що процеси роботи та розвитку слабоструктурованих системи включають різні події, що визначаються великою

кількістю чинників. Також кожний фактор впливає на деякі інші фактори.

Проблеми аналізу ситуації на основі когнітивних карт можна розділити на два типи - статичні та динамічні. Статичний аналіз або аналіз впливу - аналіз даної ситуації через вивчення структури взаємовпливу в когнітивній карті. Аналіз впливу вибирає фактори, які найбільше впливають на цільові фактори. Динамічний аналіз допомагає в створення можливих сценаріїв розвитку ситуації з часом.

В лінійній динамічній моделі, яка заснована на когнітивній карті, фактор формально визначається як змінна, що приймає значення з деякої чисельної шкали. Зміна фактору з часом дається у вигляді функції [5]:

$$x_i(t + 1) = x_i(t) + \sum_{j \in I_i} a_{ij} (x_j(t) - x_j(t - 1)), \quad i=1, \dots, N,$$

де  $x_i(t + 1)$  і  $x_i(t)$  - значення  $i$ -го фактору в момент часу  $t + 1$  і  $t$ ,  $a_{ij}$  - це сила впливу фактору  $x_j$  на фактор  $x_i$ ,  $x_j(t) - x_j(t - 1)$  - приріст фактору  $x_i$  в момент часу  $t$ .

У [8] запропонований підхід до аналізу стабільності слабоструктурованих систем на основі методу підписаних та функціональних графів та імпульсних процесів в них. Стабільність системи та пошук методів управління на основі модифікації структури для стабілізації імітаційних процесів є важливою проблемою дослідження. Розроблено методологію виведення сценаріїв слабоструктурованих систем, що дозволяє проводити дослідження їх поведінки для різних контрольних ситуацій. Аналіз динаміки роботи в неструктурованих системах ґрунтується на можливості модифікації структури моделі. Модифікація структури моделі пов'язана з розглядом обмежень з урахуванням здатності суб'єкта інтерпретувати отримані рішення. Досвід застосування моделей, побудованих на основі когнітивних карт для вирішення проблем в слабоструктурованих системах, показав,

що в деяких випадках модифікація структури моделі навряд чи інтерпретується в термінах предметної області. З іншого боку, роботи [3], [4], [5], [6] та інш. спрямовані на аналіз стійкого процесу в моделі, оскільки стабільні процеси дозволяють правильно сформулювати предметні інтерпретації подій, які розглядаються. В якості вхідних параметрів визначаються початкові значення факторів (початкові тенденції). Значення факторів не належним чином фіксуються, тобто моделюється динаміка змін ситуації на основі факторних

приростів. Ця група дослідників зосередила увагу на:

- пошуку та розробці методів структурування, спрямованих на побудову когнітивних карт.
- підвищенні ефективності підтримки вирішення проблем та завдань практичного управління.

Застосування даних підходів одночасно дозволяє проводити статичний та динамічний аналіз при вивченні слабоструктурованих або неструктурованих систем.

### Висновки

Досвід застосування різних методів та моделей на основі когнітивного підходу, а також підвищення інтересу експертів та практичних спеціалістів до нових розробок і подій у вивченні когнітивного підходу показує доцільність подальшого дослідження даного методу. Але при цьому досі залишаються не вирішені або частково не вирішені проблеми. Дослідники в цій галузі ставлять перед собою задачі:

- підвищення ефективності координації, розумінню між учасниками процесу вирішення складних проблем в слабоструктурованих системах;
- покращення формалізації слабоструктурованих або неструктурованих систем.

Сьогодні розробляються нові інструменти для вирішення різних завдань. Розробляються програмно-аналітичні системи для реалізації функції:

- побудови моделей на основі когнітивних карт;
- моделювання сценаріїв;
- оцінки та аналізу даних сценаріїв.

### Список літератури

1. R.P. Abelson, and M.J. Rosenberg. Symbolic psycho-logic: A model of attitudinal cognition. Behavioral Science, 3:1– 13, 1958.
2. Н.А. Абрамова. Субъект интеллектуальной деятельности под когнитивным контролем окружающего интеллекта, 2006.
3. З. Авдеева, С. Коврига. Определение и структура цели и анализ целей сложных систем и ситуаций, 2003.
4. З. Авдеева, С. Коврига. Когнитивный подход к решению проблем развития социально-экономических объектов, 2007.
5. В. Максимов, Е. Корнушенко. Аналитические основы построения графов и компьютерных моделей для сложных ситуаций, 2001
6. В. Максимов. Когнитивный анализ и моделирование ситуаций, 2001.
7. И.В. Прангишвили. Универсальные системные законы функционирования природных и социальных систем и проблемы контроля, 2002.
8. В.В. Кульба, Д.А. Кононов. Методы формирования сценария развития для социально-экономических систем, 2004



УДК 004.93

*КАПОРІН Р. М.  
СПЕРКАЧ М. О.*

## **ПРОГРАМНО-ІНФОРМАЦІЙНИЙ КОМПЛЕКС ДЛЯ МОНІТОРИНГУ ТА КЕРУВАННЯ ГРОМАДСЬКИМ ТРАНСПОРТОМ**

Громадський транспорт є невід'ємною частиною будь-якого великого сучасно міста. Кожен день на цю інфраструктуру доводиться велике навантаження, яке динамічно змінюється протягом доби.

У зв'язку зі зростаючою кількістю населення в містах, все частіше пасажери стикаються з перевантаженням окремих транспортних засобів на маршруті, що призводить до таких негативних явищ як травми і крадіжки.

Проблема, яка розглядається у роботі, пов'язана з оцінкою навантаження транспортного засобу пасажирами з метою покращення рівня обслуговування.

За основу проблематики береться пошук балансу між економічною стратегією роботи транспортного підприємства з обслуговування маршруту та рівнем якості послуг з перевезень. Для вирішення поставленої проблеми було запропоновано розв'язати наступну задачу: завантаження салону транспорту та часу очікування на наступний транспортний засіб.

Для вирішення проблеми оцінки пасажиропотоку на маршруті, пропонується вести підрахунок пасажирів громадського транспорту та передавати у реальному часі ці дані до віддаленого серверу. На основі таких даних можна корегувати роботу громадського транспорту для покращення якості послуг з перевезень.

Робота присвячена розв'язанню задачі оцінки кількості пасажирів на одиницю громадського транспорту.

### **КЛЮЧОВІ СЛОВА**

Географічні координати, розпізнавання образів, передача даних, громадський транспорт, маршрут, перевезення.

Public transport is an integral part of any large modern city. Every day, this infrastructure has a heavy load, which dynamically changes overnight.

Due to the increasing population in cities, passengers are increasingly facing congestion individual vehicles on the route that leads to such negative phenomena as injury and theft.

The problem considered in the work is related to the assessment of the load of the vehicle by passengers in order to improve the level of service.

The basis of the problem is the search for a balance between the economic strategy of the transport company's service of the route and the quality of transportation services. To solve this problem, it was proposed to solve the following problem: loading of the cabin and waiting time for the next vehicle.

In order to solve the problem of estimating the passenger flow on the route, it is proposed to count the public transport passengers and transfer this data to the remote server in real time. Based on these data, public transport can be adjusted to improve the quality of transportation services.

The work is devoted to solving the problem of estimating the number of passengers per unit of public transport.

### **KEYWORDS**

Geographical coordinates, pattern recognition, data transmission, public transport, route, transportation.

## 1. Введення

Громадський транспорт відіграє важливу роль у житті великого сучасного міста.

У зв'язку з підвищенням кількість населення міст, зростає й пасажирський потік, що перевозить громадський транспорт.

У цій статті наводиться ідея подолання перевантаження громадського транспорту шляхом оцінки пасажиропотоку на маршруті протягом дня та динамічного внесення коректив до розкладу роботи маршруту.

## 2. Аналіз роботи транспортних перевізників

Для оптимізації роботи маршруту громадського транспорту, транспортне підприємство повинно визначити критерії оптимізації та наявні обмеження для їх введення.

Підприємство повинно забезпечити належні умови перевезення, надавши необхідну кількість місць для сидіння та пристосувань для людей з обмеженими можливостями. З метою забезпечення належних умов перевезень, модернізується транспортний парк. Автобуси та тролейбуси обладнуються пандусами та кріпленнями для інвалідних візків.

При обслуговуванні маршрутів, підприємство має притримуватися триматися певної економічної стратегії. За неї підприємство повинно працювати не в збиток. Ця границя встановлюється у процентному значенні та прямо впливає на вартість проїзного квитка.

Перейдемо до складових економічного планування роботи та виділимо головні джерела доходів та розходів перевізника (рисунок 1 та 2).

Отримання грошей від надання послуги перевезення проходить через продаж абонементів та проїзних квитків. Існують категорії громадян, що сплачують лише частину вартості квитка або користуються безоплатним проїздом – пільговики. Компенсація за перевезення цієї групи людей надходить від держави.



**Рис. 1 – Джерела надходження доходів перевізника**



**Рис. 2 – Джерела основних витрат**

Оскільки КП «Київпастрас» є державною установою, то воно також отримує гроші з державного бюджету. До операційних доходів належать ті, що були отримані від оренди, відсотків по вкладах та ін.

Транспортне підприємство повністю залежить від його техніки, яка потребує обслуговування. З часом, техніка випрацьовує закладені у неї ресурси і вона починає вимагати заміни певних вузлів. Запчастини та паливно-мастильні матеріали відносяться до матеріальних витрат.

З метою закладання капіталу на ремонт та оновлення технічного парку, з доходів забирається певний процент та надходить до амортизаційного фонду.

Перевізник має декілька важелів впливу для регуляції доходів, але головним з них є вартість проїзду.

Знаючи кількість розходів, можна закласти процент виручки і, відштовхуючись від нього, закласти вартість квитка.

Як приклад таких розрахунків можна розглянути перерахунок тарифів столичного перевізника, комунального підприємства «Київпастрас»[1]. Відповідно до накопиченої раніше статистики та бажаного проценту виручки було обґрунтовано та встановлено нову ціну на проїзні квитки.

## 3. Процес оптимізації

Початковою точкою оптимізації є визначення змінних процесу роботи маршруту.

Для спрощення, у наших розрахунках будемо вважати, що на маршруті існують лише однотипні транспортні засоби.

Відомо, що протягом дня кількість користувачів громадського транспорту змінюється у декілька разів.

Очікуємо, що у робочі дні існують два пікових значення завантаження публічного транспорту для маршруту  $r$  вранці та ввечері.

Маючи точні дані, можна визначити необхідну кількість транспортних засобів, що виконують задачу перевезень та будуть найбільш економічно вигідними.

Однак необхідно також розуміти, що ця мінімальна кількість не забезпечить належної якості обслуговування. Таке завантаження транспортних засобів призведе до перенавантаження пасажирів транспортних засобів.

Тому перед задачею оптимізації роботи маршруту полягає задача пошуку балансу між економічною доцільністю обслуговування маршруту та його якістю за умови динамічної зміни кількості транспорту на маршруті протягом доби.

Якість обслуговування може бути оцінена за двома критеріями: завантаженість салону та час очкування транспортного засобу.

Основними характеристиками транспортного засобу є місткість та завантаженість.

Місткість  $C$  – це розрахункова (закладена виробником транспортного засобу) кількість осіб, що може перевозитися.

Завантаженість  $l$  – реальне значення кількості пасажирів у окремо взятому транспортному засобі у певний момент часу.

Нехай на маршруті  $r$  одночасно перебуває  $n$  транспортних засобів. Загальна кількість осіб, що перевозиться у поточний момент є пасажиропотік  $S$  і визначається за формулою:

$$S = \sum_{k=1}^n l_k$$

Введемо поняття *коефіцієнт завантаженості салону*  $P$ , що показує, на скільки відсотків салон зайнято

пасажирів від максимального завантаження.

За його допомогою можна обчислити необхідну кількість однотипних транспортних засобів, знаючи обсяг пасажирів, котрий необхідно перевести:

$$n = \frac{S}{P \cdot C}$$

Розглянемо критерію часу очікування на транспортний засіб  $t_g$ . У контексті поточної задачі, цей час повинен прямувати до мінімальних значень. Для обчислення значення, необхідно знати наступні параметри:

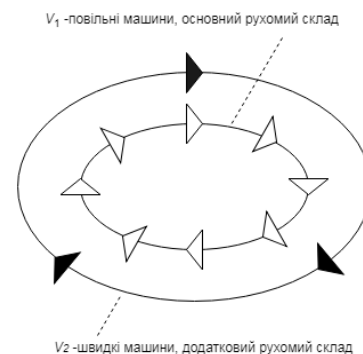
- протяжність маршруту у кілометрах  $L$ ;
- середня швидкість пересування  $V$ ;
- наявна кількість транспортних засобів  $n$ .

З цього маємо еталонний час очікування на транспорт:

$$t_g = \frac{L}{n \cdot V}$$

Корегування часу очікування можна здійснювати двома засобами:

1. зменшення / збільшення кількості транспортних засобів;
2. зміна типу транспортних засобів або їх комбінування (рисунок 3).



**Рис.3 – Комбінація типів транспортних засобів**

За умови існування двох різних типів (моделей) транспортних засобів, можна забезпечити основний показник середнього часу очікування повільним складом та корегувати його введенням транспортних засобів, що переміщуються з більшою швидкістю.

Повернемося до економічної обґрунтованості роботи. Маршрут має

скінченну довжина та буде проходитися кожний раз за приблизно однаковий час  $T$ . Усі розрахунки будемо вести для почасових робіт.

Розходи на обслуговування маршруту громадського транспорту можна звести до трьох основних груп:

- заробітна плата екіпажа машини;
- витрати на ТММ;
- ремонтні роботи транспортного засобу;

Екіпаж транспортного засобу складається з водія та касира. Означимо заробітну плату за годину як  $H^h$ .

За роботи транспортного засобу витрачається певна кількість паливно-мастильних матеріалів за годину ( $E^h$ ), яка зазначена виробником у нормативах.

Нехай  $Pr$  це вартість одиниці палива, тоді вартість роботи одного транспортного засобу на годину  $M^h$  дорівнюватиме:

$$M^h = E^h \cdot Pr$$

Останньою складовою розходів є відрахування до амортизаційного фонду. Гроші з нього будуть використовуватися на проведення ремонтних робіт, закупівлю запчастин та купування нової техніки.

Надходження до нього формуються з встановленого проценту амортизації  $Am$  від реалізованих квитків та компенсації проїзду пільгових пасажирів.

Візьмемо вартість проїзду як  $K$  і сформуємо кінцеву формулу підрахунку доходу від маршруту  $I$ :

$$I = K \cdot S \cdot (1 - Am) - n \cdot T \cdot (\sum H^h + E^h \cdot Pr)$$

За умови, що значення  $I$  додатне, маршрут повертає вкладені в нього кошти.

Маючи змогу обчислити усі три значення головних критеріїв роботи, сформуємо задачу оптимізації: для обмеженої кількості транспортних засобів, що курсують за маршрутом  $r$ , знайти оптимальну кількість транспортних засобів на маршруті за умови додержання оптимальних значень показників якості обслуговування пасажирів (часу очікування на транспорт —  $t_g$  і навантаження на окремий транспортний засіб  $S$ ) та економічної доцільності проведення перевезень, що виражено як (формула 1):

$$\begin{cases} t_g \rightarrow \min \\ P \rightarrow P_{const} \\ I \rightarrow \max \end{cases} \quad (1)$$

Задачу можна розбити на дві частини: збір інформації про поточну потужність пасажиропотоку на певному маршруті, що послугує фундаментом для майбутніх розрахунків, та підрахунок кількості транспортних засобів у залежності від обраних значень бажаної дохідності та проценту завантаження салону пасажирами.

У поточному документі розглядається саме перша частина задачі — збір актуальних даних про пасажиропотік.

#### 4. Принципи збору інформації

Підрахунок осіб у салоні транспорту можна виконати декількома шляхами:

- встановлення рамок-лічильників на входах транспорту;
- встановлення камер на зупинках громадського транспорту;
- встановлення відеокamer у салоні транспортних засобах.

Перший метод було відкинуто через вірогідну високу кількість помилкових спрацювань. Такі події можуть бути відтворені, якщо людина буде постійно стояти поміж цих рамок.

Порівняльний аналіз другого та третього варіантів [2] викриває головний недолік камер, встановлених на зупинках: складність виокремлення окремих осіб що входять та виходять з та до транспортного засобу.

Цей недолік спричинено зайвими об'єктами у кадрі, такими як: інші транспортні засоби, люди, які очікують інші машини, освітлення тощо.

Найбільш прийнятним є варіант встановлення відеокamer у салоні транспортного засобу.

На основі відео потоку з салону, можна визначити кількість людей за обличчями або контурами тіла.

Передача відео на віддалені обчислюючи потужності є недоцільним у зв'язку із повільним мережевим з'єднанням, яка може бути отримане через

встановлене мережеве обладнання стандарту 2G або 3G.

Тому обраний нами варіант пропонує встановлення камери відеоспостереження у салоні з під'єднаним до неї мікрокомп'ютером та модулем визначення географічних координат.

Перевагою техніки розпізнавання образів є її універсальність. Техніка задіяна у багатьох сферах життя: від розпізнавання надрукованих текстів до автоматичних систем забезпечення безпеки. За основу система бере попередньо підготовані дані, на підставі яких вона може навчатися.

Помилки розпізнавання образів можуть бути спричинені великою кількістю причин. Основними чинниками завад є незадовільне освітлення, задимлення, роздільна здатність зображення. Останній чинник повністю залежить від апаратних специфікацій камери, яка фіксує зображення.

## 5. Програмна реалізація клієнтського застосунку

Клієнтський прилад, який встановлено на рухомому складі, має збирати у реальному часі дані про своє місцезнаходження, визначати за відео кількість людей у салоні та передавати ці дані на віддалений сервер через заданий проміжок часу.

Перегляд поточних даних є можливим завдяки веб-сторінці, на якій постійно оновлюється інформація (рисунок 4).



Рис. 4 – Веб-сторінка клієнтської частини

На сторінці можна бачити мапу міста з декількома додатковими інформаційними

панелями. Опишемо за пунктами їх призначення.

*Панель інформації про транспортний засіб* (пункти 1, 2, 3 та 4):

1. тип транспортного засобу;
2. номер транспортного засобу;
3. кількість місць для сидіння;
4. загальна допустима кількість місць.

*Панель відображення поточної інформації* (пункти 5, 6 та 7):

5. назва панелі та індикатор роботи;
6. кількість осіб, розпізнаних у кадрі;
7. географічні координати транспортного засобу;

*Мапа міста* (пункт 8):

8. мапа міста з позначкою транспортного засобу;

*Відеопоток з камери пристрою* (пункт 9):

9. відео зображення з салону з позначеною людиною у кадрі.

Отримані від GPS модуля географічні координати та підрахована кількість осіб, компонуються та відправляються до трьох основних модулів обробки даних: локального журналу подій, веб-сторінці та модуля відправки даних на сервер.

Апаратна забезпечення клієнтської частини виконано на базі комп'ютера *Raspberry PI3*. Вибір платформи для реалізації клієнтської частини пав на мікрокомп'ютер через малий розмір, високу швидкодію та наявність інтерфейсів для під'єднання необхідної апаратури.

Програмне забезпечення написано на мові програмування *Golang*. Мова є компільованою, результатом зборки проекту є єдиний бінарний файл, який легко завантажити на віддалений хост та оновлювати його.

Підрахунок людей у кадрі виконується завдяки відкритій бібліотеці машинного зору *OpenCV*[3]. Обробка відео відбувається на основі каскадної класифікації зображення[4]. Процес виконується завдяки попередньо зібраним налаштуванням, які містять коефіцієнти співпадання частин зображення з шуканим патерном. Такий підхід надає змогу переключення між типами об'єктів які повинні бути розпізнаними: обличчя верхня частина тулуба або повний силует.

В якості локального журналу подій використовується нереляційна база даних *MongoDB*. Вона була обрана за низькі

вимоги до апаратного забезпечення та достатньої швидкості роботи записів / читання.

### Висновки

Головною проблемою оптимізації роботи підприємства з пасажирських перевезень є пошук балансу між економічною доцільністю роботи маршруту та якості обслуговування.

Опорними точками було визначено бажані показники завантаженості салону та процент отримання доходів від перевезень.

Для виконання розрахунків при пошуку оптимального режиму роботи маршруту з урахуванням цих показників необхідно брати до уваги періодичну змінюваність обсягів пасажирських перевезень протягом доби.

Пошук балансу між економічною доцільністю та комфортом перевезень засновано на отриманні реальних даних про завантаження транспортних засобів на маршруті.

Основною темою цієї статті є ідея підрахунку кількості осіб у транспортному засобі. Додатково у матеріалі викладено варіант програмно-апаратної реалізації приладу збору актуальних даних пасажирського потоку у транспорті.

Клієнтська частина заснована на мікрокомп'ютері *Raspberry PI3* до якого під'єднана відеокамера. Задачею цього пристрою є підрахунок кількості пасажирів, збір поточних географічних координат транспортного засобу та передача цих даних до віддаленого серверу.

Серверна частина опрацьовує ці дані та відображає їх у реальному часі. Додатково серверне програмне забезпечення надає функціонал для виконання експериментів над зібраними даними без внесень коректив до реально існуючих маршрутів.

### Література

1. Розрахунок планово-розрахункового тарифу на послуги з перевезення пасажирів за видами перевезень у звичайному міському режимі руху на 2018 рік. [Електронний ресурс] – Режим доступу до ресурсу: [https://kpt.kiev.ua/files/user/other/Rozrahunok\\_tarifu\\_2018\\_\\_zvichayniy\\_rejim\\_.pdf](https://kpt.kiev.ua/files/user/other/Rozrahunok_tarifu_2018__zvichayniy_rejim_.pdf).
2. Оцінка навантаження громадського маршруту транспорту. // Тринадцята міжнародна науково-практична конференція МОДС 2018. – 2018. – №13. – С. 100–103.
3. OpenCV Official Webpage [Електронний ресурс] – Режим доступу до ресурсу: <https://opencv.org/>.
4. Face Detection using Haar Cascades [Електронний ресурс] – Режим доступу до ресурсу: [https://docs.opencv.org/3.4.3/d7/d8b/tutorial\\_py\\_face\\_detection.html](https://docs.opencv.org/3.4.3/d7/d8b/tutorial_py_face_detection.html).

## СЕНТИМЕНТАЛЬНИЙ АНАЛІЗ КОРИСТУВАЦЬКИХ ТЕКСТОВИХ ДАНИХ У ВІДГУКАХ ПРО ТОВАРИ

В цій статті розглянуто задачу сентиментального аналізу користувацьких текстових даних у відгуках. Сформульовано цілі та актуальність дослідження. Виділено класифікацію користувацьких відгуків про проблеми з товарами. Описано правила створення словника оцінних слів та підхід з методами класифікації відгуків. Описано алгоритм виділення проблемних висловлювань.

This article is concerned with the task of sentimental analysis of custom text data in reviews. The goals and relevance of the research are formulated. The classification of user reviews about product problems has been highlighted. The rules for the creation of the vocabulary of evaluative words and the approach with methods of classification of reviews are described. The algorithm of identification of problem statements is described.

**Ключові слова:** СЕМАНТИЧНИЙ АНАЛІЗ, АНАЛІЗ ТОНАЛЬНОСТІ, СЛОВНИК КЛЮЧОВИХ СЛІВ, КЛАСТЕРИЗАЦІЯ

### 1. Задача сентиментального аналізу відгуків

Кількість контенту, який генерується користувачами контенту в Інтернеті виріс експотенціально за останнє десятиліття. Користувачі пишуть на форумах, блогах, залишають коментарі на множині сторінок в соціальних мережах. Весь контент несе в собі велику кількість інформації, яку ми регулярно отримуємо, аналізуємо і використовуємо. Для власників інформаційних ресурсів життєво необхідно знати думку клієнтів – будь це оцінка людьми нового продукту в інтернет-магазині чи відношення до новини на даному ресурсі.

Аналіз висловлювань користувачів, як ефективний засіб спостереження і оцінки думок користувачів, використовується в системах соціальних мереж, рейтингах при опитуваннях громадськості, торгівлі акціями, аналізі минулих та майбутніх подій, рекламних засобах (таргетинг, сервіси, що рекомендують товари), консультаційна та технічна підтримка клієнтів [1-6] і т.д.

В наукових дослідженнях описано наступні задачі сентиментального аналізу тексту відгуків користувачів:

- сентиментальний аналіз текстів у користувацьких висловлюваннях відносно аспектів [10;12];
- виділення оціночних словосполучень та слів [11];

– класифікація текстів на рівні документів і речень [13];

Поставлена задача: виділити переважаючі відгуки, що вказують на проблему товару і його цільовими об'єктами, використовуючи множину відгуків користувачів про даний товар. Задача поділяється на наступні підзадачі, що відповідають особливостям завдань сентиментального аналізу думок [9]:

1. Виділення відгуків, вказуючих на проблеми у використанні певних продуктів;
2. Виділення відгуків користувачів про проблеми, що залежать від предметної області;
3. Ідентифікація цільових об'єктів певної теми для визначення проблем описаних в множині відгуків відповідної предметної області.

### 2. Класифікація користувацьких відгуків при виникненні проблем з товарами

Огляд літератури показав, що не існує класифікації відгуків користувачів про неполадки чи недоліки продуктів. Тому постановка задачі виділення проблемних відгуків з метою класифікації тексту передбачає попередньо проведений вручну аналіз відгуків користувачів і аналіз схем класифікації, які зустрічаються в літературі, в якості додаткового знання про задачу.

Після аналізу речень з відгуків українською та англійською мовами виділено фрази чотирьох типів:

– Явна згадка проблем з продуктом. Тип містить пряму вказівку на недоліки і технічні неполадки в процесі використання продукту, такі як: «не відкривається додаток», «постійні проблеми з бортовим комп'ютером», «машині потрібний ремонт», «програма не працює належним чином».

– Неявна згадка проблем з продуктом. Тип цих фраз не містить згадування проблеми, але містить допоміжні слова і має на увазі проблему, наслідком якої є незадоволеність користувача продуктом. Приклади: «малий багажник», «слабкий сигнал мережі», «не підходящий дизайн», «заплутаний інтерфейс додатку».

– Заперечення складнощів при використанні. При використанні даного типу фраз, користувач заперечує раніше згадані або очікувані проблеми. Наприклад: «водіння не втомлює», «без претензій до якості», «шини частина трохи шумні, але зате добре тримаються на слизькій дорозі».

– Відсутність проблем з продуктом. Думка користувача не містить згадок очікуваних або фактичних труднощів. Приклади: «користувався додатком», «добре відношення ціни до якості», «служить вже десятий рік».

### 3. Словник оціночної лексики українською та англійською мовами

Описані типи висловлювань містять інформацію про існування проблем на основі конструкцій, що їх чітко ідентифікують. Одним з ключових завдань, що є основою при розробці методів для сентиментального аналізу думок текстових даних, є створення словників індикативних слів. В задачі ідентифікації проблемних слів чи висловлювань під індикативними словами розуміються слова, що виражають пряму або непряму вказівку на проблему. Вище згадані види слів та висловлювань містять інформацію про існування проблем з товаром (труднощі, недолік, проблема, корозія), негативних слів (поганий, дохлий, жахливий, кошмарний, виснажливий, неякісний), допоміжних слів (занадто, дуже, зовсім, надмірно) і заперечень дій (не їде, не працює, не відгукується, не чиниться).

Для створення словників індикативних конструкцій був використаний алгоритм описаний далі. Невеликий список індикаторів слів, що містить такі слова, як: проблема, помилка, зламатися, зависнути, для української мови і problem, fail, error, crash, bug для англійської мови, складений вручну на основі аналізу різних типів проблемних фраз. Лексичні одиниці зі словника проблемних слів можна класифікувати наступним чином: явні проблемні індикатори (дефектний, вм'ятина); оціночні слова з негативною тональністю і пов'язані зі зручністю використовуваного продукту (заплутаний, тупий, незрозумілий); дієслова, які вказують на проблемні ситуації в ході експлуатації (ламатися, промокати, розграбувати). Вихідний список містив близько 350 слів для кожної мови. Потім був застосований метод розширення списку слів шляхом додавання всіх синонімів лексичних одиниць багатомовного універсального словника, відповідними словами зі списку. Словник не проблемних слів, складений вручну, включає слова, що вказують на роботу, позитивну ситуацію або виправлення недоліків (налаштувати, зручно, несуттєво, комфортно). Таблиця 1 містить приклади проблемних індикаторів для двох мов.

Таблиця 1 – Приклади проблемних індикаторів, доданих в словники

Приклади проблемних слів і виразів			
Для української мови		Для англійської мови	
Проблема	Порушення	Error	Death
Ускладнення	Псувати	Defect	Garbage
Перешкода	Шкодити	Fault	Waste
Аварія	Пошкоджувати	Mistake	Refuse
Крах	Погіршувати	Difficult	Trash
Збиток	Ненадійний	Useless	Failure
Помилка	Сумнівний	Hard	Screw up

### 4. Підхід і методи класифікації

Для досягнення цілей задачі сентиментального аналізу пропонується підхід, заснований на знаннях. Даний підхід передбачає використання додаткових



експертних ресурсів у вигляді словників індикативних слів і виразів, складених вручну або автоматично, і написання правил, які відображають структуру фрагментів текстових даних. Перевагою цього підходу є здатність забезпечити ефективність класифікації текстів без втрат якості роботи для різних предметних областей. Пропонується два методи класифікації проблемних висловлювань:

1. Метод, що враховує умови входження слів або фраз зі словників;

2. Метод, що виконує аналіз граматичної структури складних речень щодо сполучників.

У даній статті виділяється клас проблемних висловлювань (problem клас) і клас висловлювань без проблем (no-problem клас).

### **Метод, що виконує аналіз граматичної структури складних речень щодо сполучників.**

Алгоритм виділення проблемних висловлювань складається з послідовності декількох умов, що враховують входження слів із створених словників в реченні  $S_{ij}$ :

1. Якщо знайдено входження дієслова зі словника дієслів разом із сполучним запереченням, то алгоритм виділяє речення  $S_{ij}$  як проблемне і присвоює  $r(S_{ij}) = 1$ .

2. Якщо знайдено входження проблемного індикатора зі словника проблемних слів без пов'язаного заперечення або знайдено входження слова з словника не проблемних слів разом із пов'язаним запереченням, то алгоритм виділяє все речення  $S_{ij}$  як проблемне і присвоює  $r(S_{ij}) = 1$ .

3. Якщо знайдено словосполучення, де першим словом є заперечення немає, ніякої, відсутність, відсутні, немає, другим словом є іменник, а саме словосполучення не виділено на кроці 1 і 2 і не знайдено входження слова з словника не проблемних слів разом із пов'язаним запереченням, то алгоритм виділяє все речення  $S_{ij}$  як проблемне і присвоює  $r(S_{ij}) = 1$ .

4. Якщо знайдено входження лексичної одиниці зі словника дієслів наказової форми, то алгоритм виділяє все речення  $S_{ij}$  як проблемне та присвоює  $r(S_{ij}) = 1$ .

5. Якщо на кроках 1–4 речення  $S_{ij}$  не класифікується як проблемне, воно виділяється як речення, що не містить описаної проблеми,  $r(S_{ij}) = 0$ .

Пов'язаним запереченням щодо дієслова зі словника дієслів вважається заперечення зі словника заперечень, якщо в реченні існує входження заперечення на відстані не більше, ніж  $N_{act}$  слів зліва від дієслова. Пов'язаним запереченням щодо українськомовного індикатора з словників проблемних і не проблемних слів вважається заперечення зі словника заперечень, якщо в реченні існує входження заперечення на відстані не більше, ніж  $N_{pw}$  слів (перед знаком пунктуації) від індикатора. Наприклад, у відгуку «не можу нормально завантажувати відео», заперечення знаходиться на відстані двох слів від дієслова «завантажувати». Додатково, сполучним запереченням для проблемного індикатора вважається словосполучення дієслова з запереченням зі словника дієслів, якщо словосполучення слідує після індикатора в реченні. Після ряду експериментів, відстань встановлено як  $N_{pw} = 2$  і  $N_{act} = 2$  для текстів українською мовою;  $N_{pw} = 2$  і  $N_{act} = 1$  для текстів англійською мовою.

### **Метод, заснований на аналізі граматичної структури складних речень щодо сполучників**

Користувач може описувати ситуацію використання продукту за допомогою декількох граматичних конструкцій з різними типами проблемних фраз в одному складному реченні. Виділимо наступні групи речень, щоб показати важливість ідентифікації проблемних фраз щодо сполучників в реченні:

– Перша граматична частина речення (до сполучника) має позитивну тональність, тоді як друга частина (після сполучника) відрізняється тональністю. Наприклад: «добре проходить нерівності і канави, хоча і вельми жорстка», «салон теплий і великий, але спартанський», «приємна стандартна комплектація, але немає кондиціонера».

– Перша граматична частина речення (до з'єднувального сполучника) підтверджує дефект або ускладнення при використанні, проте друга частина речення (після сполучника) заперечує проблемну або негативну ситуацію. Наприклад, «за такі гроші гарне не купити, тому ланос буде досить прийнятним варіантом», «салон, звичайно, простуватий, але це не найголовніше в машині», «користуюся цією

бляшанкою давно, але явних недоліків ще не виявив».

– Всі граматичні частини речення містять схожу інформацію про існування тих чи інших проблем у використанні. Наприклад: «гальмо залишає бажати кращого, тому сильно на ній не ганяю».

– Перша граматична частина речення містить умову виникнення проблеми, в той час як друга частина не вказує на скрутну ситуацію. Наприклад: «якщо не їздити бездоріжжю, то розганяється швидко».

Формальний опис запропонованого методу представлено у вигляді контекстно-вільної граматики – системи  $G = \langle V, \Sigma, S, R \rangle$ , заданої наступними елементами:  $V$  – множина нетермінальних (допоміжних) символів,  $\Sigma$  – множина термінальних символів,  $S \in V$  – початковий символ граматики,  $R$  – множина правил виведення вигляду  $A \rightarrow c$ , де  $A \in V$ ,  $c \in (V \cup \Sigma)$ . Правила виведення поділяються на кілька типів:

– правила виведення нетермінальних символів на основі словників;

– допоміжні правила об'єднання слів і нетермінальних символів;

– правила класифікації.

Множина термінальних символів визначена як  $\Sigma$  – алфавіт системи. Множина нетермінальних символів визначена як  $V = \{Z, WD, X, S, PS, \neg PS, clause1, clause2, conj\}$ , де  $S$  – речення,  $PS$  – проблемне речення,  $\neg PS$  – не проблемне речення,  $WD$  – множина словосполучень із запереченням (крок 3);  $X$  – множина слів з невідомою інформацією про тональність і недоліки (що не містить слів з  $N, P, IP, DP$ ).

Наведемо правила виведення нетермінальних символів  $Z \rightarrow$  ,  $Z \in \{N, P, AW, A, IP, DP, NDP, DDP, V DP\}$ , ,  $= w_0 \dots w_{k-1}$ , ,  $\in \Sigma$ , засновані на входженні зі словників слів з негативною тональністю, слів з позитивною тональністю, додаткових слів, дієслів (зі пов'язаним запереченням), проблемних слів (без пов'язаного заперечення) або не проблемних (з пов'язаним запереченням), входження явного індикатора проблеми зі словника проблемних слів , індикаторів з негативною тональністю, індикаторів дії помилкової або некоректної ситуації, відповідно. Конструкція виду  $\neg Z$  ( $Z \in V$ ) позначає заперечення  $Z$  (напр.,  $DP \rightarrow$  "проблема",  $\neg DP \rightarrow$  "без проблем").

## 5. Виділення висловлювань, що вказують на проблемні ситуації, щодо предметно-орієнтованих цільових об'єктів думок

Метою є визначення множини цільових об'єктів  $T_i = \{t_1, t_2, \dots, t_k\}$  для продуктів компанії  $P = \{P_1, P_2, \dots, P_m\}$  і класу висловлювання для всіх речень документів контрольної вибірки  $s_i, j \in d_i, j \in \{1, \dots, |d_i|\}$ ,  $i \in \{1, \dots, |D|\}$ . Актуальність поставленого завдання пояснюється необхідністю визначення інформації про товари, суттєвою для компанії. Подібна інформація повинна бути врахована системами технічної підтримки клієнтів компанії, дозволяючи виявити проблемні компоненти продукції. Наприклад, у фразах «багажник теж не відзначається простором» або «постійно пише про погане з'єднання, зайти практично неможливо» важливими для компанії цільовими об'єктами є багажник автомобіля і з'єднання телефону.

Метод визначення цільових об'єктів складається з 3 компонентів:

– визначення множини можливих цільових об'єктів, використовуючи синтаксичні зв'язки між проблемними індикаторами і іменниками в реченні;

– ідентифікації предметно-орієнтованих цільових об'єктів, використовуючи підрахунок семантичної пов'язаності між об'єктом і термінами предметної області;

– класифікації висловлювання до класу висловлювань, що вказують на проблемні ситуації, якщо метод витягує хоча б одну комбінацію (проблемний індикатор, предметно-орієнтований цільовий об'єкт).

Для ідентифікації зв'язків між проблемним індикатором і цільовим об'єктом фрази метод використовує два типи залежності між словами, визначені в роботах [7; 8]. Пряма залежність (англ. Direct dependency) між словами вказує на те, що одне слово безпосередньо залежить від іншого слова. Непряма залежність (англ. Indirect dependency) вказує, що одне слово залежить від іншого слова за допомогою слова-посередника. Слово-посередник – додаткове слово, пов'язане з проблемним індикатором і цільовим об'єктом. Приклад 1.1 містить фразу з прямою залежністю між обраними словами. Приклад 3.2 містить непряму залежність між словами. Позначення  $PW$  вказує на проблемний індикатор або заперечення дії,  $T$

позначає цільовий об'єкт фрази,  $S$  вказує на слово-посередник.

Приклад 1.1 Додаток навіть не запускається,  
 $PW$

вилітає миттєво.

Приклад 1.2 Ви коли помилку з  
 $PW$

розміщенням вкладів виправите?  
 $S$   $T$

### Алгоритм виділення предметно-орієнтованих проблемних висловлювань та цільових об'єктів

Алгоритм використовує результати аналізу текстового висловлювання запропонованими раніше методами: методом, заснованим на ряді умов, і методом, заснованим на аналізі складних речень. Загальний опис алгоритму складається з кількох кроків:

1. Витягнути з висловлювання  $s_{ij}$  входження індикаторів  $\{pw_{i1}, pw_{i2}, \dots, pw_{in}\}$ ,  $n \leq |s_{ij}|$  в залежності від пов'язаних заперечень з словників дієслів, проблемних слів, слів з негативною тональністю, додаткових слів, наказових фраз, використовуючи метод, що аналізує на ряді умов;

2. Для кожного  $pw_{ij}$  визначити множину можливих цільових об'єктів  $\{t_1, t_2, \dots, t_k\}$ , якщо цільовий об'єкт  $t_k$  синтаксично пов'язаний з  $w_{ij}$ , тобто існує пряма або непряма залежність між  $t_k$  і  $pw_{ij}$  в висловлюванні  $s_{ij}$ ; якщо множина об'єктів порожня, то  $w_{ij}$  виключається з множини індикаторів;

3. Для кожного  $t_k$  визначити чи є об'єкт предметно-орієнтованим на основі заходів пов'язаності термінів поняття  $t_k$  і термінів предметної області в лінгвістичному ресурсі;

4. Класифікувати висловлювання  $s_{ij}$  як висловлювання, яке вказує на проблемну

ситуацію про предметно-орієнтований цільовий об'єкт, якщо існує хоча б одна комбінація  $(pw_{ij}, t_k)$  і  $r(s_{ij}) \neq 0$  згідно результату аналізу методом, заснованим на аналізі складних речень; в іншому випадку, класифікувати висловлювання  $s_{ij}$  як таке, що не містить проблему.

### Висновки

В результаті для досягнення цілей дослідження в статті наводиться класифікація фраз користувачів, на основі якої побудовані словники індикативних слів і словосполучень. Запропоновано підхід, заснований на знаннях, представлених у вигляді правил і словників. В роботі створено англійський і російськомовний словники проблемних індикаторів незалежних від певної предметної області. В рамках підходу пропонується два методи вилучення фраз: Для перевірки ефективності запропонованих методів створені і розмічені контрольні вибірки відгуків користувачів, зібрані з онлайн сайтів про високотехнологічні, низькотехнологічні і механічні продукти компаній. Експериментальне дослідження показало, що найкращі результати класифікації фраз про проблеми у використанні продуктів показує запропонований метод, заснований на знаннях у вигляді словників і аналізі структури речень. Це підтверджує необхідність семантичного аналізу речень для виявлення проблем з продуктами. Аналіз результатів класифікації підтвердив, що подальше поліпшення результатів можливо за рахунок створення вузькоспеціалізованих словників і розробки умов входження лексичних одиниць в залежності від тематичний категорії обраного фрагменту тексту.

### Список літератури

1. Prentice S., Huffman E. Social medias new role in emergency management // Idaho National Laboratory. — 2008. — С. 1—5.
2. From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series. / В. О'onnor [та ін.] // ICWSM. — 2010. — Т. 11, 122-129. — С. 1—2.
3. Nguyen T. H., Shirai K. Topic modeling based sentiment analysis on social media for stock market prediction // Proceedings of the 53rd Annural Meeting of the Association for Computational Linguistics. — 2015.
4. Detecting sadness in 140 characters: Sentiment analysis of mourning michael jackson on twitter / E. Kim [та ін.] // Web Ecology. — 2009. — Т. 3. — С. 1— 15.
5. Fan T.-K., Chang C.-H. Sentiment-oriented contextual advertising // Knowledge and Information Systems. — 2010. — Т. 23, № 3. — С. 321—344.
6. Graphical modeling of macro behavioral targeting in social networks / Y. Xie [та ін.] // Proceedings of SDM. — SIAM. 2013.
7. Dem̃sar J. Statistical comparisons of classifiers over multiple data sets // The Journal of Machine Learning Research. — 2006. — Т. 7. — С. 1—30.
8. N. Loukachevitch [та ін.] // Proceedings of International Conference Dialog. Т. 2. — 2015. — С. 12—24.
9. Pang B., Lee L. Opinion mining and sentiment analysis // Foundations and trends in information retrieval. — 2008. — Т. 2, 1-2. — С. 1—135.
10. Pang B., Lee L. Opinion mining and sentiment analysis // Foundations and trends in information retrieval. — 2008. — Т. 2, 1-2. — С. 1—135.
11. Liu B. Sentiment analysis and opinion mining // Synthesis lectures on human language technologies. — 2012. — Т. 5, № 1. — С. 1—167.
12. Tsytsarau M., Palpanas T. Survey on mining subjective data on the web // Data Mining and Knowledge Discovery. — 2012. — Т. 24, № 3. — С. 478—514.
13. Sentiment analysis in Twitter / E. Martinez-C'аmara [и др.] // Natural Language Engineering. — 2014. — Т. 20, № 01. — С. 1—28.

УДК 004.023

КОГУЛЬКО О.С.  
ПОПЕНКО В.Д.

## НАДАННЯ РЕКОМЕНДАЦІЙ ЕЛЕМЕНТІВ НА ОСНОВІ ГІБРИДНОЇ ФІЛЬТРАЦІЇ

В даній статті розглянуто спосіб надання рекомендацій елементів на основі гібридної фільтрації. Сформульовано задачу та цілі з точки зору рекомендаційної системи книг. Виділено методи, які будуть використовуватися при застосуванні гібридної фільтрації. Описано алгоритм гібридної фільтрації.

In this article is concerned with way of recommendations processing on the basis of a hybrid filtering. Formulated the problem and its objectives in terms of books recommendations system. Described methods to be used when applying hybrid filtering. Described the algorithm of hybrid filtration.

**Ключові слова:** рекомендації, міра схожості, релевантність, колаборативна фільтрація, фільтрація за змістом, демографічна фільтрація.

### 1. Вступ

Рекомендаційні системи стали потужним інструментом для розширення та утримання аудиторії різноманітних веб-сервісів. Сьогодні можна отримати миттєвий доступ до величезної кількості інформації. Саме тому бажано використовувати рекомендаційні системи, які здатні пропонувати користувачу найбільш релевантні елементи і тим самим відсторонювати його від глибокого пошуку. Рекомендаційні системи успішно застосовуються сервісами з надання новин, книг, фільмів та інших послуг.

Рекомендаційні системи для реалізації коректної роботи застосовують комплексний підхід, який включає в себе застосування принципів та напрацювань з декількох галузей знань. Зокрема: когнітивістика, прогнозування, машинне навчання, інформаційний пошук і т.д. Також значного впливу рекомендаційні системи зазнали з боку бізнесу, адже результати маркетингових досліджень є не менш цінними, ніж інформаційний аналіз даних. Але завжди – рекомендаційні системи використовувалися для знаходження рейтингів об'єктів для конкретного користувача. Це може використовуватися як суто нова інформація, тобто передбачення вподобань користувача щодо нових елементів, так і для коригування роботи системи, за рахунок порівняння реальної оцінки користувача з прогнозованою.

Таким чином, рекомендаційна система – це система, націлена на передбачення вподобань користувачів щодо елементів, що виступають у якості основного контенту системи (фільми,

книги, музичні твори, новини та ін.), за рахунок відомої інформації користувача або елементів всередині системи.

### 2. Знаходження найбільш релевантного об'єкта

Математично, задачу знаходження найбільш релевантного об'єкта можна описати наступним чином:

$$\forall u \in U, s'_u = \operatorname{argmax}_{s \in S} h(u, s)$$

Де  $U$  – множина користувачів,  $u$  – вибраний користувач,  $S$  – множина елементів системи, які потенційно можуть бути рекомендовані користувачу,  $s$  – вибраний елемент,  $h$  – функція, яка показує ступінь зіставлення деякого елемента  $s$  з деяким користувачем  $u$ .

Таким чином, задача зводиться до знаходження такого елемента  $s' \in S$ , для якого ступінь зіставлення з користувачем  $u \in U$  є максимальним (у випадку рекомендації одного елемента).

При цьому на точність рекомендації суттєво впливає не лише інформація всередині системи, а і вибір функції  $h$ .

До основних методів вирішення поставленої задачі відносяться:

- Фільтрація на основі вмісту (item-based)
- Колаборативна фільтрація (user-based)
- Гібридний підхід

### 3. Фільтрація на основі вмісту

Як можна зрозуміти з назви підходу, фільтрація на основі вмісту перш за все

опирається на інформацію про контент системи, а не про користувачів. Тобто рекомендації базуються на знаходженні схожих елементів, до тих, що користувач вже оцінив в минулому. Для цього необхідно створити профіль користувача та профіль елемента. Після цього, на основі параметрів елементів системи, можна зробити висновок щодо відповідності конкретного елемента конкретному користувачу. Для опису елементів системи та створення їх профілю рекомендаційної системи ставлять у відповідність кожному елементу певний набір ключових слів.

Таким чином, користувацький профіль формується у вигляді параметрів, що характеризують кожен елемент  $s' \in S'$ , де  $S'$  – множина оцінених елементів. У якості таких параметрів найчастіше виступають ключові слова, також вагові коефіцієнти ключових слів для кожного з елементів системи. Одним з найчастіше використовуваних способів знаходження даних вагових коефіцієнтів є TD-IDF міра [1].

Її можна обчислити за наступною формулою:

$$TF_{ij} = \frac{f_{ij}}{\max_z f_{zj}}$$

Де  $f_{ij}$  - кількість входжень деякого ключового слова  $k_i$  в елемент  $d_j$ , при цьому ключове слово  $k_j$  зустрічається в  $n_i$  об'єктах.

Але при використанні вищезазначеної формули враховується лише частоти входження ключового слова. Це може призвести до випадку, коли максимальну вагу будуть мати найбільш розповсюджені ключові слова, що в подальшому може призвести до некоректного прогнозування вподобань конкретного користувача. Для уникнення такого випадку використовується величина  $IDF_i$ , яка є зворотною до частоти входження ключового слова в елемент.

$$IDF_i = \log \frac{N}{n_i}$$

Де  $N$  – кількість елементів, які потенційно можуть бути рекомендовані користувачу.

Виходячи з вищезазначених формул, можемо обчислити вагу  $w_{ij}$  ключового слова  $k_i$  в елементі  $d_j$  наступним чином:

$$w_{ij} = TF_{ij} * IDF_i$$

Тоді профіль елемента та профіль користувача можна представити у вигляді TF-

IDF векторів  $\vec{w}_s$  та  $\vec{w}_u$ , при цьому функція зіставлення елемента та користувача  $h(u, s)$  може бути представлена у вигляді косинусу кута між векторами  $\vec{w}_s$  та  $\vec{w}_u$ :

$$h(u, s) = \cos(\vec{w}_u, \vec{w}_s) = \frac{\vec{w}_u \vec{w}_s}{\|\vec{w}_u\| \|\vec{w}_s\|} = \frac{\sum_{i=1}^K w_{iu} w_{is}}{\sqrt{\sum_{i=1}^K w_{iu}^2} \sqrt{\sum_{i=1}^K w_{is}^2}} \quad (1)$$

Де  $K$  – загальна кількість ключових слів в системі.

#### 4. Колаборативна фільтрація

Методи колаборативної фільтрації для побудови рекомендацій використовують інформацію про оцінки інших користувачів, які належать до одної групи. В цьому випадку функція  $h(u, s)$  задає ступінь зіставлення елемента  $s$  з користувачем  $u$ , та обчислюється в залежності від зіставлення  $h(u_j, s)$  для елемента  $s$  для користувачів  $u_j \in U'$ , де  $U'$  - деяка група користувачів.

Для визначення ступеня схожості користувачів та формування груп, необхідно побудувати профіль користувача для кожного з них. При побудові профіля може використовуватися як дані про вподобання користувачів, так і більш загальна інформація, наприклад вік, стать, національність, профілі в соціальних мережах та ін. Подібна загальна інформація про користувача є дуже важливою на перших етапах роботи користувача з рекомендаційною системою, коли користувач ще не встиг оцінити достатньо елементів, для формування вподобань його профілю. Тобто маємо випадок, коли система нічого не знає про користувача, і не може побудувати для нього коректні рекомендації. Подібна проблема має назву “холодний старт”. Для зменшення впливу даної проблеми і необхідно використовувати загальні дані про користувача, адже на основі них також можна віднести його до деякої групи.

Методи колаборативної фільтрації можна поділити на 2 типи:

- методи, засновані на пам'яті;
- методи, засновані на моделях.

Методи, засновані на пам'яті, використовують дані про всі елементи, які

користувач оцінив в минулому, для передбачення рейтингу нового елемента:

$$r_{us} = \text{aggr} r_{u \in u' r_{u's}}$$

Де  $r_{us}$  - передбачений рейтинг елемента  $s$  для користувача  $u$ ,  $U'$  - множина користувачів, які оцінили елемент  $s$ , та входять в одну групу з користувачем  $u$ .

В якості функції агрегування можна використовувати різні підходи. Найпростіший з них – загальне середнє:

$$r_{us} = \frac{1}{N} \sum_{u \in U'} r_{u's}$$

Де  $N$  – потужність множини  $U'$ .

Частіше для агрегування застосовуються наступний вираз:

$$r_{us} = k \sum_{u \in U'} \text{sim}(u, u') * r_{u's} \quad (2)$$

Де  $k$  – параметр для нормалізації, який обчислюється наступним чином:

$$k = \frac{1}{\sum_{u \in U'} \text{sim}[u, u']}$$

Тут функція  $\text{sim}(u, u')$  - міра схожості (англ. "similarity"). Дана величина є зворотною до відстані і в більшості випадків використовується в якості вагових коефіцієнтів. Тобто, в залежності від схожості користувачів  $u, u'$ , ступінь впливу оцінок користувача  $u'$  на рекомендації для користувача  $u$  буде варіюватися.

В більшості систем схожість користувачів перш за все визначається схожістю оцінок, які вони дали окремим елементам. Власне в якості міри, найбільш використовуваних 2 методи:

- знаходження кореляції;
- знаходження косинусних коефіцієнтів.

При використанні колаборативної фільтрації  $S_{uu'}$  використовується у якості проміжного параметру, для виявлення найбільш схожих користувачів, та може обчислюватися багатьма методами, найпростіший з яких – перетин множин  $S_{u'}$  та  $S_u$ . Деякі методи [2] дозволяють знайти найближчих сусідів користувача  $u$  без обчислення  $S_{uu'}$ . Для цього використовується коефіцієнт Пірсона [3,4]:

$$\begin{aligned} \text{sim}(u, u') &= \\ &= \frac{\sum_{s \in S_{uu'}} (r_{us} - \bar{r}_u)(r_{u's} - \bar{r}_{u'})}{\sqrt{\sum_{s \in S_{uu'}} (r_{us} - \bar{r}_u)^2 \sum_{s \in S_{uu'}} (r_{u's} - \bar{r}_{u'})^2}} \quad (3) \end{aligned}$$

В методах, які використовують косинусний коефіцієнт [5,6] користувачі  $u, u'$  представлені у вигляді векторів в  $m$ -мірному просторі, де  $m = |S_{uu'}|$ . Таким чином міра схожості може бути обчислена за допомогою наступної формули:

$$\begin{aligned} \text{sim}(u, u') &= \cos(\vec{u}, \vec{u}') = \frac{\vec{u} \cdot \vec{u}'}{\|\vec{u}\| \cdot \|\vec{u}'\|} = \\ &= \frac{\sum_{s \in S_{uu'}} r_{us} r_{u's}}{\sqrt{\sum_{s \in S_{uu'}} r_{us}^2} \sqrt{\sum_{s \in S_{uu'}} r_{u's}^2}} \quad (4) \end{aligned}$$

## 5. Гібридний підхід

Деякі рекомендаційні системи застосовують гібридний підхід для формування рекомендацій. Гібридність полягає у поєднанні результатів двох різних підходів – фільтрації на основі вмісту та колаборативної фільтрації. Завдяки цьому можна, в деякій мірі, уникнути недоліків вищезазначених підходів. Існує декілька основних варіантів комбінування підходів до фільтрації:

- включення характеристик елементів в колаборативну фільтрацію;
- включення характеристик користувачів в фільтрацію за змістом;
- побудова єдиної моделі на основі колаборативної фільтрації та фільтрації за змістом.

## 6. Модифікація алгоритму

Модифікуємо алгоритм надання рекомендацій. Для цього розділимо метод колаборативної фільтрації на 2 окремих:

- фільтрація на основі сусідства;
- демографічна фільтрація.

В методі фільтрації на основі сусідства, рейтинг елемента  $s$  для користувача  $u$  прогнозується за рахунок рейтингів користувачів, які, на думку системи, мають схожі до даного користувача вподобання. Даний підхід залежить від обраної кількості сусідів користувача  $u$ , оцінки яких будуть враховуватися при прогнозуванні рейтингу. Власне ступінь сусідства обчислюється за

допомогою міри схожості (косинусна міра, кореляція).

Демографічні характеристики користувачів відіграють дуже важливу роль при формуванні рекомендацій. Рекомендаційні системи часто зіштовхуються з згаданою вище проблемою “холодного старту”, через недостатню кількість даних про оцінки користувачів [7]. В таких випадках необхідно використовувати дані, які доступні з самого початку роботи користувача з системою.

Найважливішою задачею рекомендаційної системи є прогнозування рейтингу. В нашій моделі прогнозування рейтингу буде обчислюватися за рахунок поєднання трьох описаних методів: фільтрація на основі змісту, фільтрація на основі сусідства та демографічна фільтрація.

Тобто, при побудові рекомендацій будемо використовувати 3 методи окремо, для підвищення точності рекомендацій. Кожен з методів буде обчислювати прогнозований рейтинг незалежно від інших. Після чого, отримані результати необхідно лінійно об'єднати, окремо враховуючи внесок результатів кожного методу в результуючому рейтингу. Для обчислення міри впливу результату кожного з методів, введемо ваговий коефіцієнт для кожного методу. Результуючим прогнозованим рейтингом, буде рейтинг, що обчислюється за допомогою наступної формули:

$$r_{us} = \frac{\alpha DF_{us} + \beta CB_{us} + \gamma CF_{us}}{\alpha + \beta + \gamma} \quad (5)$$

Де  $CB_{us}$ ,  $CF_{us}$ ,  $DF_{us}$  - прогнозовані рейтинги елемента  $s$  для користувача  $u$  отриманих за допомогою методів фільтрації на основі змісту, фільтрації на основі сусідства та демографічної фільтрації відповідно.  $\alpha, \beta, \gamma$  - відповідні вагові коефіцієнти.

Для обчислення значень вагових коефіцієнтів необхідно застосувати деяку функцію  $f(x)$ , яка дає значення 1 для великих значень аргументу та 0 для малих значень аргументу. Цим умовам чудово відповідає функція сигмоїда, яка задається за допомогою логістичної рівності:

$$S(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1}$$

Використовуючи цю функцію, задамо вагові коефіцієнти за допомогою наступної системи рівнянь:

$$\begin{cases} \gamma = \frac{1}{1 + e^{-\frac{x}{2}}} \\ \alpha = \beta = 1 - \frac{1}{1 + e^{-\frac{x}{2}}} \end{cases} \quad (6)$$

Де  $x$  – прогнозований рейтинг. Тобто значення параметрів  $\alpha, \beta, \gamma$  змінюються динамічно в залежності від доступних оцінок від користувача  $u$ .

## 7. Розробка алгоритму розв'язання

Крок 1. до демографічної фільтрації.

Крок 2. ОБЧИСЛИТИ вагові коефіцієнти для кожного користувача, на основі подібності демографічних даних (вік, стать, національність і т.д.) з заданим користувачем використовуючи міру схожості (3).

Крок 3. ВИБРАТИ зі списку користувачів, які є найбільш наближеними до даного користувача, групу визначеного розміру.

Крок 4. ОБЧИСЛИТИ прогнозований рейтинг елемента на основі оцінок користувачів з обраної групи (2).

Крок 5 . ПЕРЕЙТИ до фільтрації за змістом.

Крок 6. ОБЧИСЛИТИ вагові коефіцієнти для кожного елемента, на основі подібності властивостей елемента (жанр, рік видання, кількість сторінок та ін.) з заданим елементом використовуючи міру схожості (4).

Крок 7. ВИБРАТИ зі списку елементів, групу елементів, які є найбільш подібними до даного, на основі вагових коефіцієнтів.

Крок 8. ОБЧИСЛИТИ прогнозований рейтинг елемента на основі оцінок подібних елементів (1).

Крок 9. ПЕРЕЙТИ до колаборативної фільтрації.

Крок 10. ОБЧИСЛИТИ вагові коефіцієнти для кожного користувача, на основі подібності їх вподобань з вподобаннями заданого користувача, використовуючи міру схожості (3).

Крок 11. ВИБРАТИ зі списку користувачів, які є найбільш наближеними до даного користувача, групу визначеного розміру.

Крок 12. ОБЧИСЛИТИ прогнозований рейтинг елемента на основі оцінок користувачів з обраної групи (2).

Крок 13. ОБЧИСЛИТИ вагові коефіцієнти для кожного з методів фільтрації (6).

Крок 14. ОБЧИСЛИТИ результуючий прогнозований рейтинг елемента (5).

На цьому робота алгоритму завершується.



## 8. Результати експериментів

При проведенні експериментів використовувався набір даних з відкритим доступом Goodbooks 10K. Він містить в собі дані про 10000 книг та оцінки 50000 користувачів. В якості оцінок ефективності алгоритмів будемо використовувати наступні міри релевантності [8]:

- точність;
- повнота;
- F – міра (F1 score).

Експерименти будемо проводити використовуючи різні комбінації вищезазначених алгоритмів:

- 1) Колаборативна фільтрація (CF)
- 2) Фільтрація на основі вмісту (CB)
- 3) Демографічна фільтрація (DF)
- 4) **CB + CF**
- 5) **CF + DF**
- 6) **DF + CB**
- 7) Наш підхід

Табл.1 Результати експериментів

Алгоритм	Точність	Повнота	F-міра
<b>CF</b>	0.621	0.54	0.577
<b>CB</b>	0.719	0.69	0.704
<b>DF</b>	0.651	0.63	0.64
<b>CB+CF</b>	0.639	0.641	0.64
<b>CF+DF</b>	0.609	0.61	0.609
<b>DF+CB</b>	0.65	0.659	0.654
Наш підхід	0.722	0.7	0.71

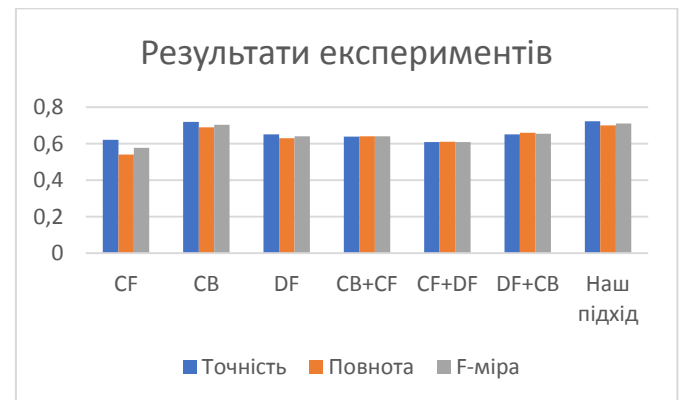


Рис.1 Порівняння підходів

Як видно з Табл.1 та Рис. 1 – наш підхід є більш ефективним за інші.

## 9. Висновки

На прикладі задачі обчислення рекомендацій елементів було застосовано гібридний підхід фільтрації. Був приведений аналіз існуючих методів розв'язання задачі та описаний модифікований алгоритм гібридної фільтрації. Була проведена серія експериментів з застосування різних підходів для прорахунку рекомендацій. Показана перевага модифікованого алгоритму перед рештою підходів.

## Список літератури

1. G. Salton Automatic Text Processing // Addison-Wesley. – 1989.
2. C.C. Aggarwal *Horting Hatches an Egg: A New Graph-Theoretic Approach to Collaborative Filtering* / C.C. Aggarwal, J.L. Wolf, K-L. Wu, P.S. Yu // Proc. Fifth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining. – 1999.
3. P. Resnick *GroupLens: An Open Architecture for Collaborative Filtering of Netnews* / P. Resnick, N. Iakovou, M. Sushak, P. Bergstrom, J. Riedl // Proc. 1994 Computer Supported Cooperative Work Conf. – 1994.
4. U. Shardanand *Social Information Filtering: Algorithms for Automating "Word of Mouth"* / U. Shardanand, P. Maes // Proc. Conf. Human Factors in Computing Systems. – 1995.
5. J.S. Breese *Empirical Analysis of Predictive Algorithms for Collaborative Filtering* / J.S. Breese, D. Heckerman, C. Kadie // Proc. 14th Conf. Uncertainty in Artificial Intelligence. – 1998.
6. B. Sarwar *Item-Based Collaborative Filtering Recommendation Algorithms* / B. Sarwar, G. Karypis, J. Konstan, J. Riedl // Proc. 10th Int'l WWW Conf. – 2001.
7. E. Aïmeur, G. Brassard, J. M. Fernandez, and F. S. M. Onana, *Privacy-preserving demographic filtering*, in Proceedings of the ACM symposium on Applied computing. New York, NY, USA: ACM, 2006, pp. 872–878.
8. Powers, David M. W. *Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation* // Journal of Machine Learning Technologies, 2011, pp. 37–63.

УДК 004.93(015.7)

*КОЛИЩАК Б.В.*

*БАКЛАН І.В.*

## **ІДЕНТИФІКАЦІЯ ТА АУТЕНТИФІКАЦІЯ КОРИСТУВАЧА НА ОСНОВІ ДИНАМІЧНИХ БІОМЕТРИЧНИХ ХАРАКТЕРИСТИК З ВИКОРИСТАННЯМ АПАРАТУ ПРИХОВАНИХ МАРКОВСЬКИХ МОДЕЛЕЙ**

У даній роботі розглядається рішення задачі активної аутентифікації з використанням історії трасування. Запропоновано метод ідентифікації користувачів за еталонними ознаками, заснований на динамічних біометричних характеристиках та застосуванні прихованих марковських моделей в якості апарату прийняття ідентифікаційних рішень.

**ПРИХОВАНА МАРКОВСЬКА МОДЕЛЬ, БІОМЕТРИЧНА ІДЕНТИФІКАЦІЯ, РОЗПІЗНАВАННЯ ОБРАЗІВ, ІДЕНТИФІКУЮЧІ ОЗНАКИ**

In this paper, we consider the solution of the problem of active authentication using trace history. The method of identification of users with reference features based on dynamic biometric characteristics and the use of hidden Markov models as a mechanism for identification decisions is proposed.

**HIDDEN MARKOV MODEL, BIOMETRIC IDENTIFICATION, PATTERN RECOGNITION, IDENTIFYING CHARACTERISTICS**

### **1. Вступ**

Технологічний розвиток обчислювальних технік привів до проникнення комп'ютерних систем у різноманітні галузі людської діяльності. В результаті значні і насамперед цінні ресурси стали доступними через локальні і глобальні мережі. Це стосується, зокрема, електронної комерції, інтернетбанкінгу, корпоративних застосувань, що містять конфіденційну інформацію,

або навіть звичайні домашні комп'ютери, що містять певну персональну інформацію і мають вихід до глобальних мереж. Тому постала нагальна проблема персоналізації користувача, котрий має право доступу до певних ресурсів. Вирішення цієї проблеми за допомогою логіна і пароля виявилось вразливим і не достатньо ефективним.

В останні роки все більше і більше пристроїв надають біометричні датчики, такі як відбитки пальців або малюнок на екрані в якості альтернативних варіантів аутентифікації для розблокування пристроїв, але як і раніше вимагають пароль в якості останнього довідкового ресурсу у разі, якщо біометричні рішення

не працюють з повторними спробами. Незважаючи на те, що ці нові параметри не можуть уникнути проблеми юзабіліті. Пристрій, як і раніше, вимагає від користувача багаторазового введення паролів або пінів, дотики до екрану з допомогою малюнка або розміщення пальця на датчику відбитків пальців. Тому, як звести до мінімуму проблему зручності використання незручностей є значним, важливим, але складним.

У цій роботі пропонується рішення пасивної і безперервної аутентифікації користувача на базі апарату прихованих Марковських моделей. Підхід не залежить від традиційних параметрів аутентифікації, таких як пароль або біометричних даних. Підхід не залежить від традиційних параметрів аутентифікації, таких як пароль або біометричних даних. Метод використовує історичну поведінку користувача при роботі з мишею, у зв'язку з тим, що у кожного користувача існує власна манера управління маніпулятором «миша»

## 2. Прихована марковська модель

Дискретна прихована марковська модель визначається набором станів і алфавітом вихідних символів (Rabiner 1989). Кожен стан характеризується двома розподілами ймовірності: розподілом переходу за станами і розподілом викидів над вихідними символами. Випадкове джерело, що описується такою моделлю, породжує послідовність вихідних символів наступним чином: на кожному кроці часу джерело знаходиться в одному стані, а після випуску символу виводу відповідно до розподілу викидів поточного стану, джерело переходить до наступного стану відповідно до перехідного розподілу його поточного стану. Оскільки діяльність джерела спостерігається побічно, через послідовність вихідних символів, а послідовність станів не спостерігається безпосередньо, стани називаються прихованими. Безперервна ПММ відрізняється від дискретної, оскільки вихідні символи викидаються з щільності ймовірності замість розподілу. Ми віддаємо перевагу дискретній моделі. Для дискретних та безперервних ПММ існують алгоритми динамічного програмування для:

- обчислення ймовірності спостереження послідовності;
- знаходження послідовності станів, що максимізує ймовірність даної послідовності, коли модель відома (алгоритм Viterbi);
- індукування ПММ, яке максимізує (локально) ймовірність даної послідовності (алгоритм Baum-Welch).

## 3. Отримання подій

Перш за все, необхідно виділити стійкі ідентифікаційні ознаки і створити еталон користувача. Еталонні значення ознак повинні бути характерно помітні у різних людей, а зразки ознак однієї людини, повинні бути схожі. Для виявлення інформативних ознак розроблений програмний модуль. Один з потенційних ознак заснований на оцінці середнього часу переміщення курсору миші між

елементами інтерфейсу. Чим далі або точніше виконується рух суб'єкта, тим більше корекції необхідно для його виконання, і відповідно, більше часу потрібно суб'єкту для внесення цієї корекції. Одна з ознак полягає в тому, що при внесенні корекції рухів проявляються індивідуальні особливості людини, що працює з маніпулятором «миша». Таким чином, фактичний час переміщення не повинен збігатися з оцінкою, що обчислюється за формулою (1), а повинно відрізнитися на величину  $\Delta T$ , яку вирішено використовувати як один з ідентифікуючих ознак в цій роботі. Іншими ознаками, використаними в роботі є максимальне  $C_{max}$  і середнє  $C_{mid}$  відхилення (в пікселях) від найкоротшого шляху  $S_{min}$  переміщення курсору між двома елементами інтерфейсу.

$$T = b \cdot \log_2\left(\frac{D}{W} + 1\right),$$

де  $b$  — величина, яка залежить від типової швидкості руху курсору миші (обчислюється як відношення середньої швидкості руху миші по екрану, здійснюваного суб'єктом, до встановленого в операційній системі коефіцієнту чутливості миші),  $D$  - дистанція переміщення курсору між елементами інтерфейсу (в пікселях),  $W$  – ширина елемента інтерфейсу, до якого направляєтся курсор (в пікселях).

Також в якості ознак використовуються амплітуди перших десяти низькочастотних гармонік функції швидкості переміщення курсору миші по екрану  $V_{xy}(t)$ . Розкладання функції проводиться за допомогою швидкого перетворення Фур'є, тим самим досягаючіється нормування ділянок шляху курсору за часом. Амплітуди  $A^* = A/E_s$  нормуються по енергії функції  $V_{xy}(t)$ , щоб привести всі траєкторії переміщень курсору між елементами інтерфейсу до єдиного масштабу. Аналогічні операції здійснюються по відношенню до функцій координат курсору  $x(t)$ , і  $y(t)$ , однак попередньо дані функції переводяться в іншу систему

координат, вісь абсцис розташовується в напрямку центру

елемента інтерфейсу, по відношенню до якого проводиться переміщення курсору. Це необхідно виконувати, щоб

позбутися від нахилу ліній, що зв'язують елементи інтерфейсу, щодо вихідної координатної площини (тобто залежності координат від кута нахилу).

$$V_{xy}(t) = \sqrt{(x_{(t+1)} - x(t_i))^2 + (y_{(t+1)} - y(t_i))^2},$$

де  $x$  і  $y$  – координати курсору,  $t_i$  –  $i$ -ий момент часу реєстрації координат курсору (реєстрація координат курсору здійснюється розробленим програмним модулем і залежить від продуктивності комп'ютера).

$$E_s = \int_{-\infty}^{\infty} A^2(\omega) dt,$$

де  $A(\omega)$  – амплітуди гармоніки з частотою  $\omega$  функції  $V_{xy}(t)$

#### 4. Створення еталонів і оцінка інформативності ознак

На підставі критерію Хі-квадрат Пірсона встановлено, що закони розподілу значень описаних ознак найбільш близькі до нормального

Тому для створення еталонів були розраховані параметри нормального закону розподілу – математичні очікування і середньоквадратичні відхилення значень кожної ознаки. Ці параметри являють собою еталон. За допомогою розробленого програмного модуля була зібрана база, що складається з набору не менш ніж 100 значень кожної ознаки. Так як швидкість реакцій суб'єкта істотно залежить від його темпераменту, то випробовувані підбиралися таким чином, щоб серед них була приблизно рівна кількість представників кожного типу темпераменту.

Аналіз амплітудного спектру функцій  $V_{mid}(t)$ ,  $x(t)$ , і  $y(t)$  показав, що найбільш інформативною є амплітуда першою гармоніки функції  $V_{mid}(t)$ . Вирішено в якості ознак використовувати тільки амплітуди перших чотирьох гармонік

функцій (з частотами 2, 4, 8 і 16 Гц), оскільки коливання руки випробованого з більшою частотою під час роботи з мишею є шум.

#### 5. Прийняття ідентифікаційних рішень

В якості способу прийняття рішень вирішено використовувати стратегію прихованих Марковських моделей, засновану на послідовному застосуванні формули. Кожна гіпотеза має на увазі, що дані про рухи належать конкретному користувачеві, тобто кожна гіпотеза асоціюється з певним еталоном. Приховані Марковські моделі дають дещо менший відсоток помилкових рішень у просторі малоінформативних ознак.

$$P_j(H/A) = \frac{P_{j-1}(H_i/A) \cdot P(A_j/H_i)}{\sum_{i=1}^n P_{j-1}(H_i/A) \cdot P(A_j/H_i)},$$

де  $P_j(H/A)$  – апостеріорна ймовірність  $i$ -ої гіпотези, що обчислюється при надходженні  $j$ -ої ознаки,  $P(A_j/H_i)$  – умовна ймовірність  $i$ -ої гіпотези при надходженні  $j$ -ої ознаки. Таким чином, на кожному кроці за апріорну ймовірність приймається апостеріорна ймовірність, обчислена на попередньому кроці. При кожному переміщенні курсора вимірюються значення ознак і розраховуються апостеріорні ймовірності гіпотез за 12 кроків. До здійснення переміщень курсору (на

нульовий крок) всі гіпотези є однаковими,

тобто,  $P_0(H_i/A) = \frac{1}{n}$ , де  $n$  – кількість гіпотез

(користувачів). Умовна ймовірність обчислюються виходячи із закону розподілу ознак (в даному випадку нормального), як щільності ймовірності значень ознак.

#### 6. Обробка даних

Біометрична система перевірки користувача – система розпізнавання образів, яка отримує біометричні дані від

людини, визначає набір ознак, і встановлює унікальний, призначений для користувача, підпис і будує модель перевірки, щоб класифікувати підписи між різними користувачами.

Система складається з чотирьох компонентів. Спочатку ми збираємо дані у вигляді часових серій, спостережуваних за

сеанс. Потім відбувається збір і обробка даних. Наступний етап супроводжується вивченням профілю користувача на основі апарату прихованої Марковської моделі, даними для якої є результуючі об'єкти. На рис. 1 зображена високорівнева архітектура пропонуваного рішення аутентифікації.

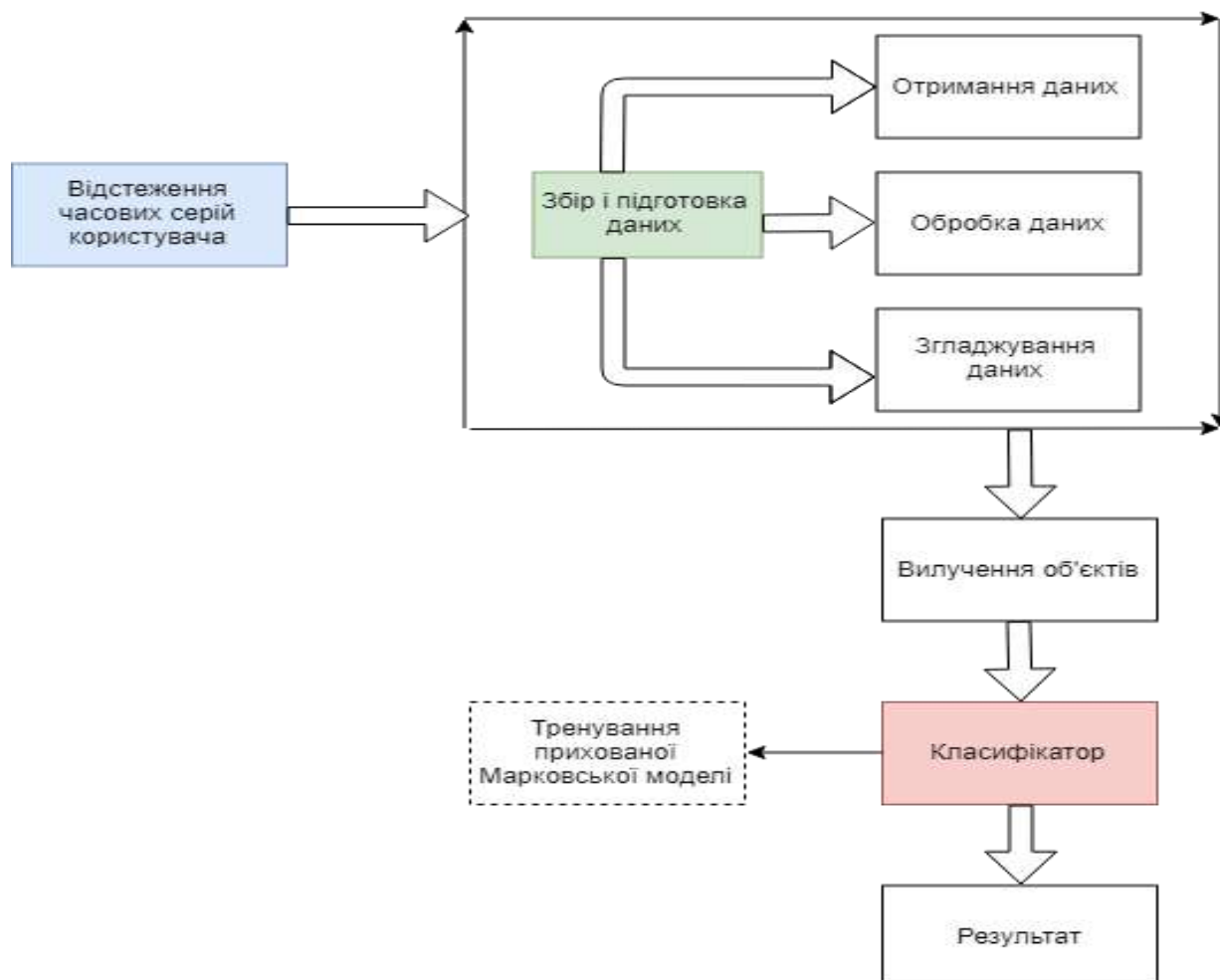


Рис. 1. Високорівнева архітектура аутентифікації

### Висновок

Підводячи підсумки, можна констатувати, що при кількості відомих системі користувачів понад 10 до 20 можлива ймовірність правильного розпізнавання в заданому просторі ознак перевищує 0,95, тобто сума помилок 1-ого та 2-ого роду може бути нижче 5%. При меншій кількості ідентифікованих образів теоретично можливо здійснити правильне розпізнавання з ймовірністю більше 0,99. Оданако на практиці для досягнення зазначених показників потрібно збільшити час на ідентифікацію. Це обумовлено тим, що в задачах ідентифікації в просторі малоінформативних часто відбуваються збої у вимірюванні ознак, внаслідок чого на етапі ідентифікації значення ознаки може «не потрапити» у свій розподіл. Тож на практиці доцільно використовувати модель прихованих Марковських моделей, описаних у роботі

**Список літератури**

1. B. H. Juang and Rabiner, L.R. Hidden Markov Models for Speech Recognition. American Statistical Association and the American Society for Quality Control, 33. 251-272.
2. Jorgensen Z., Yu T. On mouse dynamics as a behavioral biometric for authentication, in: Proceedings of the Sixth ACM Symposium on Information, Computer, and Communications Security (AsiaCCS), March 2011
3. A. A. E. Ahmed, I. Traore A new biometric technology based on mouse dynamics, IEEE Transactions on Dependable and Secure Computing, 4(3), pp. 165-179, 2007.
4. A. J. Mansfield and Wayman, J.L. Best Practices in Testing and Reporting Performance of Biometric Devices, Center for Mathematics and Scientific Computing, National Physics Laboratory, Queens Road, Teddington, Middlesex, TW 11 0LW, August 2002.
5. Saurabh Singh, Dr. K. V. Arya, «Mouse Interaction based Authentication System by Classifying the Distance Traveled by the Mouse» International Journal of Computer Applications (0975–8887) Volume 17. – № 1, March 2011.

УДК 519.68; 681.513.7; 612.8.001.57; 007.51/.52

*МАМАШКО Є.О.,  
ГАВРИЛЕНКО О.В.*

## **ДОСЛІДЖЕННЯ АЛГОРИТМІВ РОЗПІЗНАВАННЯ ЖЕСТИВ РУК ДЛЯ ВЗАЄМОДІЇ МІЖ ЛЮДЬМИ ТА МОБІЛЬНИМИ ПРИСТРОЯМИ**

В даній статті розглядаються можливі методи створення системи стеження за рукою на відео, отриманому з камери смартфона. Для досягнення цієї мети було використано апарат комп'ютерного зору та машинного навчання. Результати дослідження наведені в четвертому розділі.

**КЛЮЧОВІ СЛОВА:** ВІДСТЕЖЕННЯ РУКИ В РЕАЛЬНОМУ ЧАСІ, ВІДЕО, КОМП'ЮТЕРНЕ БАЧЕННЯ, ГІСТОГРАМА ОРІЄНТОВАНИХ ГРАДІЄНТІВ.

This article discusses possible methods for creating a hand tracking system on video from a smartphone camera. To achieve this goal, a computer vision and machine learning apparatus was used.

**KEYWORDS:** HAND TRACKING ON REAL-TIME VIDEO, COMPUTER VISION, HISTOGRAM OF ORIENTED GRADIENTS, SPEEDED UP ROBUST FEATURES.

### **1. Вступ**

Наразі простота взаємодії з сучасними пристроями є більш важливим питанням ніж навіть швидкість обробки даних черговим пристроєм, адже технологічність комплектуючих дозволяє достатньо дешево створювати потужні комплектуючі у потрібній кількості. Тому повсякчас виникають різні концепти керування та взаємодії з електронікою, адже від цього залежать враження та ефективність роботи з різноманітними пристроями.

Для людей очевидним способом взаємодії є жести, отже їх інтелектуальний аналіз дозволить керувати різними системами. Перевагою такого підходу є відсутність необхідного додаткового обладнання, потрібна лише камера, яка зараз є чи не в усіх мобільних пристроях (і смартфонах, і ноутбуках). Особливістю задачі розпізнавання жестів на відео з камери смартфона є обмеженість обчислювальних ресурсів смартфона. Для плавності розпізнавання необхідна частота близько 20 кадрів на секунду, але для складних алгоритмів (наприклад, багатошарових нейронних мереж) не буде достатньо обчислювальної здатності смартфона. Для розпізнавання жестів потрібно

отримувати відео з камери смартфона, що також потребує обчислювальних ресурсів. Також на смартфоні окрім розпізнавання жестів має працювати програма, в якій це розпізнавання жестів використовується, при чому в додатках віртуальної реальності основною задачею є рендер стереозображення, що при складних сценах є саме по собі затратною задачею.

В даній статті розглядаються можливі методи створення системи стеження за рукою на відео, отриманому з камери смартфона. Для досягнення цієї мети було використано апарат комп'ютерного зору та машинного навчання.

### **2. Огляд використаних алгоритмів**

Аналіз перемішень за допомогою методу Віоли-Джонса. Метод використовує техніку ковзного вікна – для пошуку об'єктів, які нас цікавлять, відбувається обхід зображення вікном, розмір якого менший, ніж розмір оригінального зображення. На кожному кроці відбувається розпізнавання чи знаходиться в даному вікні об'єкт. Особливістю методу Віоли-Джонса є використання для класифікації алгоритму AdaBoost.

Для розпізнавання використовуються ознаки Хаара – ознаки, які використовують суму пікселів з прямокутних областей. Однією з особливостей методу Віюлі-Джонса є використання інтегрального зображення – спеціальної форми представлення зображення, яка спрощує обчислення ознак Хаара[4].

Оскільки висока швидкість обчислень не компенсує значну кількість різних можливих ознак (так у нашому прикладі кількість ознак досягає приблизно  $32 \cdot 10^5$ ), тому в методі Віюлі-Джонса використовується модифікація алгоритму AdaBoost[5], яка обирає найкращі варіанти для тренування класифікатора на них. Цей класифікатор будує так званий “сильний” класифікатор як лінійну комбінацію “слабких” класифікаторів.

### 3. ЗАПРОПОНОВАНЕ РІШЕННЯ

В результаті проведеного дослідження літератури було вирішено створити нейронну мережу з частинами, які можна налаштовувати.

На даний час метод Джонса-Віюлі з використанням ознак Хаара каскадного AdaBoost класифікатора [6] для виявлення людських обличчя вважається одним з кращих алгоритмів для таких завдань. Метод Джонса-Віюлі був розроблений для завдання виявлення людських обличчя на зображенні. Головна перевага цього методу полягає в його високій швидкості виконання і точності виявлення. У даній роботі ми

### 4 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

На етапі генерації «словника» для кожного класу відбираються по 25 «чистих» зображень (без фону) з об'єктом, знятим в різних ракурсах і масштабах. Кількість виділених ознак залежить від значення порога Гессена. Якщо встановити мале значення порога, то кількість ознак збільшується, але при цьому потрібно тривалий час обробки.

модифікуємо детектор Джонса-Віюлі з використанням нового набору ознак Хаара для виявлення руки у відео потоці в режимі реального часу. У традиційному методі Джонса-Віюлі (і в модифікованому Lienhard і Maydt детекторі) застосовується стандартний набір ознак. У статті ми застосовуємо новий набір ознак, запропонований в роботі [7]

Основна ідея запропонованого алгоритму полягає в тому, що зображення розглядається як текстовий документ, в якому кожен візуальний ознака враховується як слово, що утворює даний документ. Клас документа визначається шляхом обчислення частоти появи деяких «ключових слів». Для застосування цієї ідеї в розпізнаванні об'єкта спочатку з набору зображень (бази даних для навчання) методом SURF (Speeded Up Robust Features) як на рисунку 5, виділяються характерні ознаки [8]. Потім всі виділені ознаки поділяються на групи, використовуючи K-means кластеризацію. Кожна група служить «ключовим словом», з яких будується так званий «словник ключових слів». Дескриптори для нейронної мережі створюються шляхом зіставлення виділених ознак вхідного зображення з ключовими словами з словника ознак за допомогою пошуку K найближчих сусідів. Набір дескрипторів і відповідний клас в подальшому використовуються для навчання нейронної мережі.

Якщо встановити великий поріг, то число виділених ознак може бути не придатним для розпізнавання. Для зображення заданого розміру поріг Гессена був обраний таким чином, щоб для зображень класу (а) SURF виокремлював в середньому 30 ознак; для класу (б) - 70; для класу (в) - 90; для класу (г) - 80. Тоді число слів в словнику дорівнює 270. Для навчання



нейронної мережі в базу даних додавали ще 75 зображень для кожного класу. Зображення для навчання були зняті на простому тлі для підвищення робастності класифікатора. Для тестування класифікатора веб-камерою були зняті 1000 зображень для кожного класу. Розмір кожного зображення становить 100×100 пікселів. Програма дає можливість генерації словника ознак, формування бази даних для навчання, створення і навчання нейронної мережі, а також має окремий інтерфейс для перевірки і тестування працездатності.

Результати тестування на вибірці з 1000 різних зображень при часі

оброблення одного кадру близько 90 мс показують, середня точність розпізнавання становить близько 82% на навчальній вибірці та 70% на тестовій. Найвищу точність отримано при розпізнаванні «відкритої долоні». Це можна пояснити тим, що для «відкритої долоні» число виділених ознак найбільше. Для «каменю» кількість ознак набагато менше інших. Час обробки для всіх зображень складає ~ 90 мс, що дозволяє працювати в реальному часі зі швидкістю близько 11 кадрів в секунду. Приклад розпізнавання жесту «fist», «palm» та «down palm» можна побачити на рисунках 1-5.

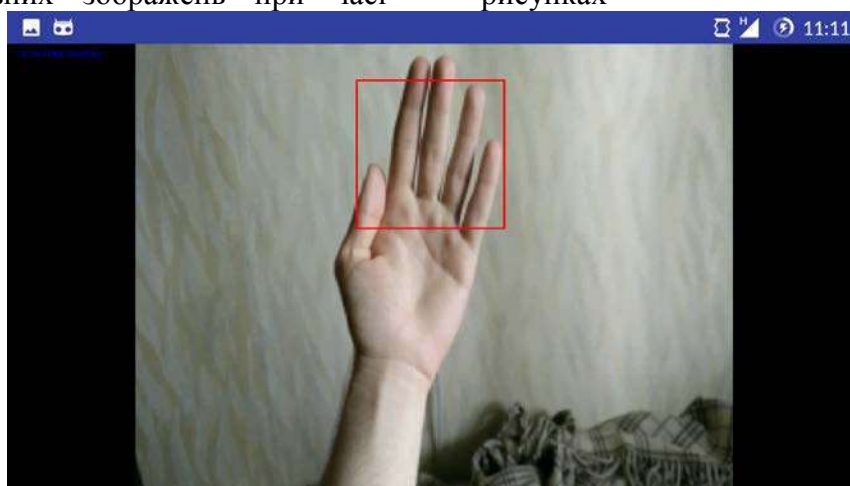


Рисунок 1 – Розпізнавання долоні



Рисунок 2 – Розпізнавання кулака



Рисунок 3 – Розпізнавання кулака



Рисунок 4– Розпізнавання повернутого кулака

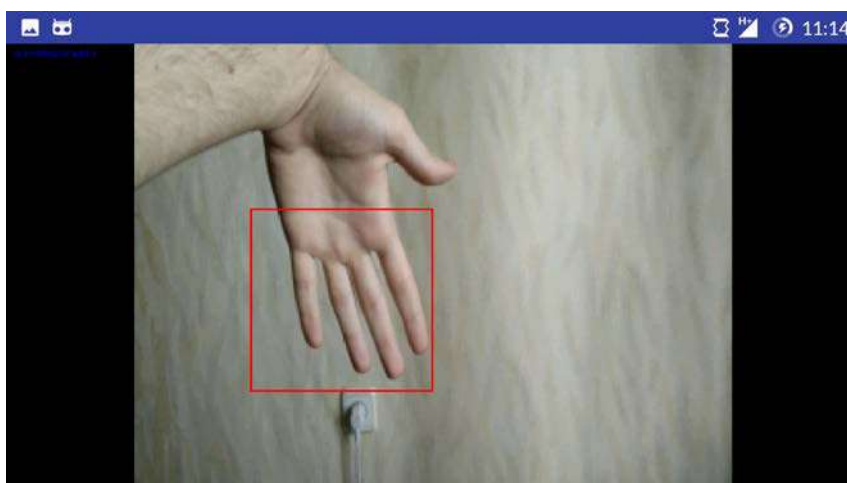


Рисунок 5 – Розпізнавання перевернутої долоні

## 5. Висновки

У даному розділі було розглянуто метод знаходження об'єктів за допомогою вдосконаленого метода SURF та нейроних мереж, було його реалізовано та портовано на смартфон. Для навчання було використано вибірку, в якій були зображення руки (приблизно 3000 позитивних зображень та 30000 негативних). Побудована модель має точність розпізнавання близько 75% на навчальній та 68% на тестовій вибірці. Швидкість розпізнавання близько 10 кадрів на секунду. Точність системи є досить високою але не ідеальною, що можна пояснити тим, що вибірка складається із зображень знятих статичною камерою, а не з кадрів рухомого відео з камери смартфона. Також, якщо розпаралелити алгоритм та перенести його виконання на відеочіп, можна отримати максимально швидку систему знаходження руки.

## СПИСОК ЛІТЕРАТУРИ

1. Handbook of natural language processing / Ed. by N. Indurkha, F.J. Damerau. // London: Chapman & Hall/CRC, 2010. – 692 p.
2. Douglas L.T.R., Laura M.G., David C.P. An improved method for deriving word meaning from lexical co-occurrence // Cognitive Science. – 2009. – V. 7. – № 2. – P. 573–605.
3. Foltz P.W., Kintsch W., Landauer T.K. The measurement of textual coherence with Latent Semantic Analysis // Discourse Processes. – 1998. – V. 25. – № 2. – P. 285–307.
4. Lund K., Burgess C. Producing high-dimensional semantic spaces from lexical co-occurrence // Behavior Research Methods, Instruments and Computers. – 1996. – V. 28. – № 2. – P. 203–208.

## ПОРІВНЯННЯ АЛГОРИТМІВ ДЛЯ ГЕНЕРАЦІЇ ЦИФРОВОГО ПІДПISУ

Безпека проведення будь яких фінансових операцій на даний час є дуже важливою і в сучасному світі для реалізації технологій, які зможуть допоготи її реалізувати придіється дуже багато ресурсів. Ідея генерування цифрового підпису не нова, та використовується дуже широко в різноманітних галузях, проте для більш ширшого її використання необхідно шлянути на цю техноголою з іншої сторони. Можливість генерації одноразового цифрового підпису для унікальної транзакції з мінімальними затратами ресурсів дозволить ще більш обширніше використовувати цю технологію не тільки для юридичних осіб а і для фізичних осіб в повсякдених потребах.

ФІНАНСОВА БЕЗПЕКА, ТРАНЗАКЦІЯ, ЦИФРОВИЙ ПІДПIS, КРИПТОГРАФІЯ.

В наведеній інформації нище буде проведено аналіз криптографічних алгоритмів, які зможуть бути реалізовані для нашої системи. Можливість генерації одноразового цифрового підпису для унікальної транзакції з мінімальними затратами ресурсів дозволить ще більш обширніше використовувати цю технологію не тільки для юридичних осіб а і для фізичних осіб в повсякдених потребах, таких як оплата товарів через інтернет, використання інтрнет банкінгу та підпис інших документів.

FINANCIAL SECURITY, TRANSACTION, DIGITAL SIGNATURE, CRYPTOGRAPHY.

### 1. Вступ

В наведеній інформації нище буде проведено аналіз криптографічних алгоритмів, які зможуть бути реалізовані для нашої системи. Можливість генерації одноразового цифрового підпису для унікальної транзакції з мінімальними затратами ресурсів дозволить ще більш обширніше використовувати цю технологію не тільки для юридичних осіб а і для фізичних осіб в повсякдених потребах, таких як оплата товарів через інтернет, використання інтрнет банкінгу та підпис інших документів.

### 2. виявлення недоліків та переваг алгоритмів

#### RSA

Це алгоритм шифрування з відкритим ключем. Особливості безпеки цього алгоритму зумовлені складністю проблеми факторизації. Алгоритм використовує два ключа - публічний та приватний, які формують відповідну пару. Публічний ключ використовується для цілей шифрування даних. Якщо повідомлення зашифровано за допомогою відкритого ключа, його можна розшифрувати лише за допомогою прив'язаного приватного ключа.

#### Переваги

- Зручний розподіл пари ключів. Не потрібно створювати додаткові компоненти.
- Використання в великих мережах провокує створення меншу кількість ключів ніж асиметричні криптоалгоритми.

#### Недоліки

- Низька робоча швидкість
- Високі витрати ресурсів для забезпечення "сили" шифрування відносно спроб фальсифікації
- Сприйнятливність до мультиплікативної атаки. Іншими словами, це дозволяє зловминому користувачеві, який не знає таємного ключа, генерувати підпис для документів, в якому результат хешування може бути розрахований як хеш-результати вже підписаних документів.

Система RSA використовується в гібридних криптосистемах із симетричними алгоритмами. RSA використовується в операційних системах, таких як Microsoft, Apple, Sun і Novell

#### EIGamal шифрування

Алгоритм шифрування EIGama - це тип алгоритму шифрування з відкритим ключем. Особливості безпеки цього алгоритму зумовлені труднощами обчислення дискретних логарифмів у кінцевому полі. Си-

стема шифрування ElGamal охоплює алгоритми шифрування та цифрового підпису.

#### Переваги

- Імовірнісний характер шифрування, що пропонує високий рівень безпеки
- Можливість генерувати цифрові підписи для великої кількості повідомлень, використовуючи лише один секретний ключ

#### Недоліки

- Подвоєння зашифрованої довжини тексту у порівнянні з початковим, що призводить до більш тривалого часу для передачі та більш жорстких вимог до безпеки каналу зв'язку.

Це рішення застосовується в сертифікатах відкритих ключів для захисту з'єднань у TLS (SSL, HTTPS, WEB), повідомленнях XML-підпису (XML Encryption) та цілісності IP-адресу та доменних імен (DNSSEC).

### DSA

Це алгоритм шифрування з відкритим ключем, призначений для створення електронного підпису. Підпис створюється "в приватному порядку", але його можна перевірити "публічно". Іншими словами, існує лише один предмет, який може створити підпис, доданий до повідомлення, але кожен зможе перевірити, чи підпис є правильним чи ні. Особливості безпеки цього алгоритму впливають з обчислювальної складності взяття логарифмів в кінці процесу генерування.

Алгоритм DSA є модифікацією алгоритму шифрування ElGamal і пропонує ряд переваг.

#### Переваги

- Коротша довжина підпису, незважаючи на однакові рівні міцності;
- Висока швидкість обчислення підписів;
- Невеликий необхідний обсяг пам'яті

#### Недоліки

- Верифікація підпису реалізовується за допомогою вирахування залишків від обчислень, тому швидкість роботи може бути різною для кожного випадку;

### ECDSA

Це ще один алгоритм шифрування публічного ключа, призначений для створення

електронного підпису та є модифікацією алгоритму DSA.

#### Переваги

- Можливість працювати у набагато менших даних, ніж у випадках, коли використовується алгоритм DSA
- Немає з непередбачуваною швидкістю роботи
- Швидкий підпис та його перевірка
- Відповідність сучасним вимогам захисту
- Робота відповідно національних стандартів захисту інформації

#### Недоліки

- Можна отримати помилку, яка дає змогу вибрати значення приватного ключа та ідентичні підписи для різних документів. Однак, для того, щоб отримати таку помилку, необхідні величезні ресурси обчислювальної машини.

Еліптичні криві алгоритми використовуються в TLS, PGP, SSH. Цей тип шифрування додатково застосовується на біткойнах та інших платформах блокчейн.

### Алгоритм підпису Шнорра

Будучи модифікацією системи шифрування ElGamal та схеми Fiat-Shamir, вона як і раніше виграє тим, що пропонує менший розмір згенерованого підпису.

Алгоритм використовується в національному стандарті Республіки Білорусь (STB 1176.2-99) та південнокорейських стандартах KCDSA та EC-KCDSA.

### Rapid Digital Signature

Принципи швидкого підписання спираються на алгоритми DS: BLS, Diffie-Hellman та алгоритм Fiat-Shamir. Ця реалізація використовується алгоритмами з меншою кількістю обчислень. Алгоритм, що розглядається, також включає в себе процеси створення пари користувацьких ключів, підрахунку підписів та функцій перевірки. Він спирається на реалізацію використовуючи дискретні логарифми.

#### Переваги

- Спрощення складності алгоритмів, що дозволяє використовувати без високої затрати ресурсів

**Недоліки**

- Недостатньо дослідженні можливості взлому
- Обмежений реалізацією, в якій в один момент часу може порівнюватись тільки одна параг.

Цей варіант доцільний у випадках, коли швидкість генерації ключа та розмір підпису є важливими, наприклад в мобільних системах, смарт-картках та мобільних телефонах.

**GMR алгоритм**

Це модифікація RSA. Рівні сили алгоритму обумовлені складністю цілісної факторизації. Його перевагою перед RSA є його захист від нападів адаптивно відібраних повідомлень.

**Rabin Cryptosystem**

Алгоритм підпису з різними рівнями міцності. Особливості безпеки цього алгоритму зумовлені складністю цілісної факто-

ризації. Цей алгоритм не є широковикористовуваним.

**Переваги**

- Вища операційна швидкість, ніж RSA

**Недоліки**

- Можливість бути взламаним, способом вибору даних з зашифрованого тексту.

**Алгоритм EdDSA**

Цей алгоритм є схемою підпису з використанням алгоритма Шнорра та еліптичних кривих. Алгоритм EdDSA спирається на схему підписів Ed25519 на основі SHA-512/256 та Curve25519.

**Переваги**

- Висока швидкість
- Незалежність генератора випадкових чисел
- Висока продуктивність

Ця методика використовується в OpenSSH, GnuPG, OpenBSD, Nacl / libsodium, протоколах криптовалют CryptoNote, WolfSSL та I2Pd.

**Висновок**

Сьогодні цифрові підписи використовуються у всіх сферах. Системи Blockchain, що представляють майбутнє, і в перспективі будуть впроваджені в всі області інформаційних технологій, використовують цифровий підпис. Ідея визначення оптимального для впровадження такого алгоритму в систему, яка допоможе звичайним користувачам генерувати одноразові цифрові підписи для унікальної транзакції. В подальшому необхідно виконати дослідження затрати ресурсів цими алгоритмами та впровадити його в таку систему, яка значно полегшить процес здійснення фінансових операцій та може стати новик кроком в розвитку цієї сфери. Як зрозуміло з вище наведеної інформації, технічні можливості є.

**Перелік посилань**

1. <https://dzone.com/articles/jwt-quick-reference-for-the-header-and-jws-paramet> - A Guide to Digital Signature Algorithms
2. M. O. Rabin, Digitalized Signatures, Foundations of Secure Computation (Richard A. Demillo, David P. Dobkin, Anita K. Jones, and Richard J. Lipton, eds.), Academic Press, 1978, pp. 155–168.
3. Kaur, Goyal V. "A Comparative Analysis of ECDSA VS. RSA Algorithm", International journal of computer science and informatics, ISSSN (Print):2231-5292, volume-3, Issue-1, 2003.
4. Khalique, Singh k., Sood S. "Implementation Of Elliptic Curve Digital Signature Algorithm", International journal of computer applications (0975-8887), Volume 2–No.2, 2010.
5. Jhonson, Menezes A., Vanstone S, "The Elliptic Curve Digital Signature Algorithm (ECDSA)", Certicom research, Canada, 2001.

УДК 004.93(015.7)

ЧИЖ Ю.М.,  
СТЕЦЮК В.З.

## ОПТИМІЗАЦІЯ ПРОЦЕСУ РЕЄСТРАЦІЇ ТА СПИСАННЯ ЛАБОРАТОРНОГО ВИТРАТНОГО МАТЕРІАЛУ І РЕАГЕНТІВ

У статті описана комп'ютерна десктопна програма для автоматизації ведення обліку хімічного посуду та реагентів та звітності про надходження та списання «Облік матеріалів та реагентів медичної лабораторії», розроблена для оптимізації процесу реєстрації та списання лабораторного витратного матеріалу та реагентів.

**КЛЮЧОВІ СЛОВА:** медична лабораторія, лабораторний витратний матеріал, реагенти, статистична звітність, база даних

The article describes the computer desktop program for automation of accounting of chemical vessels and reagents and reporting of admission and cancellation "Accounting of materials and reagents of the medical laboratory", developed to optimize the registration and write-off process of laboratory consumables and reagents.

**KEYWORDS:** medical laboratory, laboratory consumables, reagents, statistical reporting, database

### 1. Вступ

У статті описана комп'ютерна десктопна програма для автоматизації ведення обліку хімічного посуду та реагентів та звітності про надходження та списання «Облік матеріалів та реагентів медичної лабораторії», розроблена для оптимізації процесу реєстрації та списання лабораторного витратного матеріалу та реагентів.

### 2. Причини, що спонукали до створення програмного застосунку

Робота лікаря-лаборанта є важливим етапом діагностики та лікування пацієнта. Результати його роботи допомагають лікуючому лікарю встановити діагноз і, за потреби, направити пацієнта до спеціаліста відповідного профілю. Саме тому, висока точність та якість результатів аналізу є важливою складовою роботи лікаря-лаборанта.

Після вивчення проблематики сфери діяльності цієї професії виникла ідея створення комп'ютерної програми для автоматизації процесу реєстрації та списання лабораторного витратного матеріалу та реагентів. Дана програма передбачає полегшення роботи лікаря-лаборанта; контроль за кількістю хімічного посуду та реагентів та строком придатності останніх; зменшення імовірності виникнення помилок з обрахунками та втрат даних про надходження та списання

лабораторного посуду та реагентів. Головною перевагою впровадження програми є зменшення часу на рутинну роботу з медичною документацією та оформлення прийому/списання медичних матеріалів, що в свою чергу збільшує час для роботи з аналізами, що забезпечує кращу точність та якість результату.

Метою даної роботи і є створення комп'ютерної десктопної програми «Облік матеріалів та реагентів медичної лабораторії» для реалізації вказаних вище функцій.

Актуальність науково-дослідницької роботи полягає в розробці дієвого інструменту у вигляді комп'ютерної програми, що забезпечує обробку облікової медичної документації про медичні матеріали та реагенти з метою покращення рівня якості та точності результатів аналізів та полегшення роботи лікаря-лаборанта.

Предметом дослідження є реляційна локальна база даних формату SQL Server та основні компоненти архітектури Microsoft SQL Server. Об'єкт дослідження – база даних електронної медичної облікової документації медичної лабораторії генетики дитячої лікарні «ОХМАДИТ», способи її обробки та формування необхідної звітності (з використанням основних елементів архітектури Microsoft SQL Server).

Одним із завдань роботи є застосування бази даних медичної документації для

обліку медичних матеріалів та реагентів для отримання оперативної та періодичної статистичної звітності.

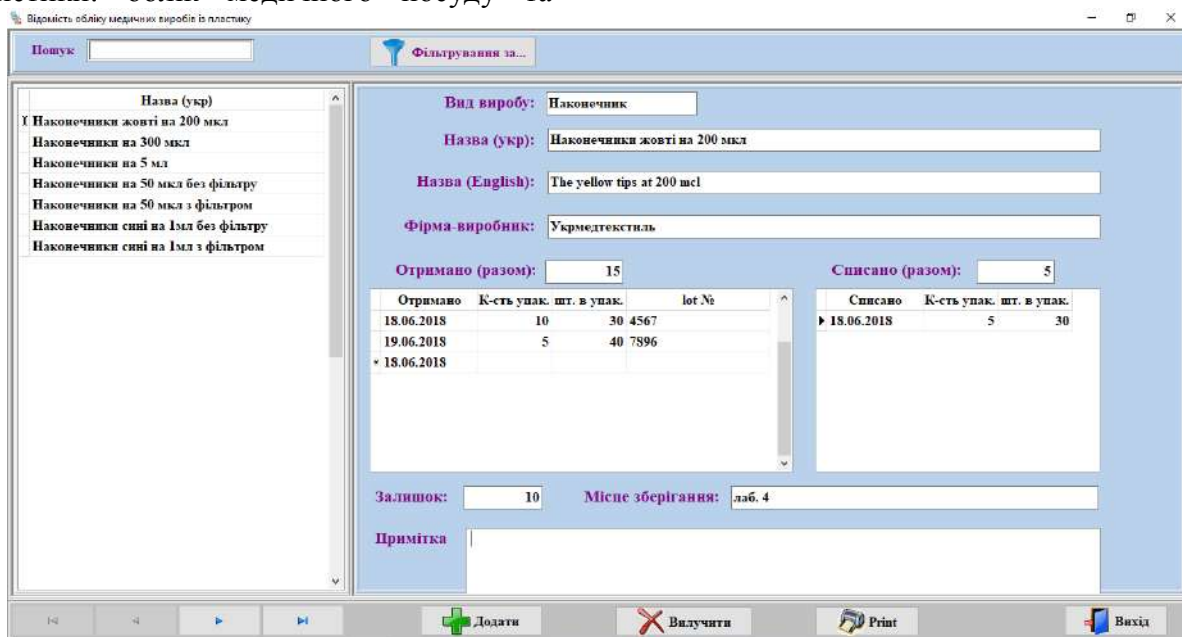
### 3. Мета дослідження

Метою даного дослідження є створення комп'ютерної десктопної програми, яка взяла б на себе виконання певних функцій медичного документообігу, звітності та статистики: облік медичного посуду та

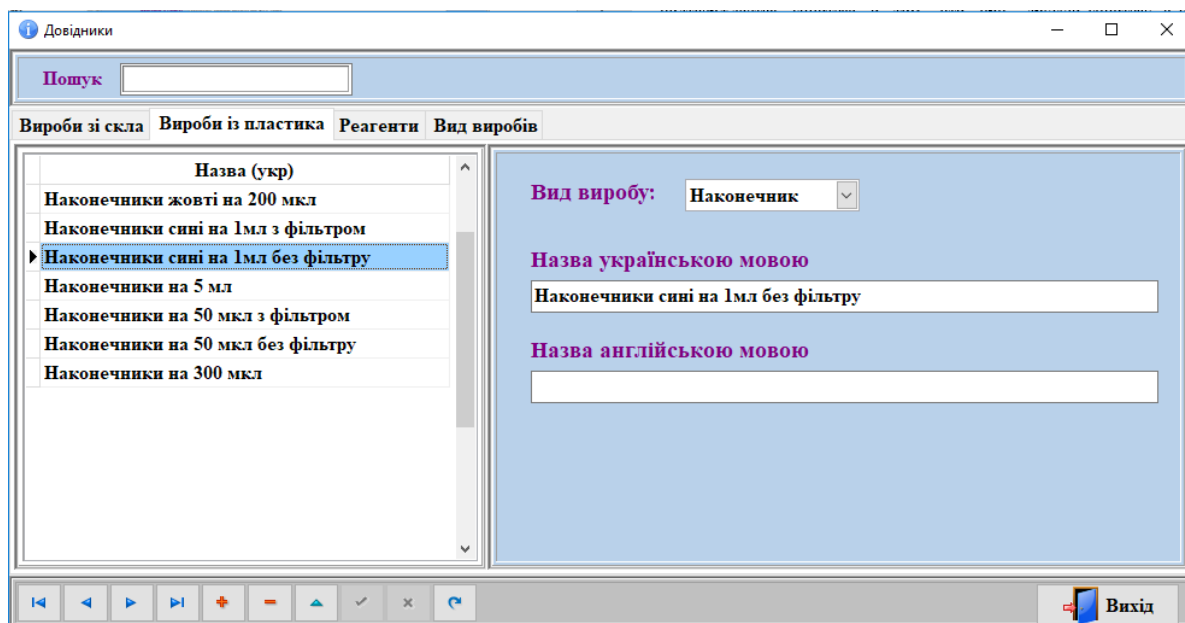
реагентів, основної інформації про кожен виріб, формування електронної звітності та інше.

### 4. Матеріали та методи дослідження

Матеріалом дослідження стала робота лікаря-лаборанта, який проводить біохімічні дослідження різноманітного біологічного матеріалу (крові, сечі, жовчі, тощо), веде лікарську документацію.



*Рис. 1 Відображення інформації про окрему одиницю лабораторного витратного матеріалу*



*Рис. 2 Довідники для кожного виду лабораторного витратного матеріалу або реагентів*



### 5. Результати та їх обговорення.

Результатом дослідження роботи працівників медичної лабораторії дитячої лікарні та аналізу необхідної облікової документації став програмний продукт, що оптимізує роботу персоналу лабораторії.

Прикладна комп'ютерна програма "Облік матеріалів та реагентів медичної лабораторії" (рис. 1) встановлюється на одному комп'ютері медичної лабораторії. Оператором даної програми є лікар-лаборант, що веде облік медичних матеріалів та реактивів. Ця прикладна програма передбачає отримання медичної та статистичної звітності.

Програма реалізує такі функції:

- облік виробів в залежності від його виду (скло, пластик) та підкатегорії;
- динамічне керування наявністю підкатегорій для кожного виду виробів (рис.2);
- оформлення надходження та списання кожного виробу та реагенту;
- збереження основної інформації про кожен виріб;
- автоматичний контроль над залишком в залежності від отримання/списання кожного виробу;
- контроль за строком придатності кожного отриманого реагенту;
- формування звітності за вказані періоди з можливістю перегляду та друку інформації за певними показниками.

### Реєстр надходження виробів із пластику за червень 2018 р.

<b>Назва</b>	Наконечники жовті на 200 мкл		
<b>Тип</b>	Наконечник		
<b>Виробник</b>	Укрмедтекстиль	<b>Місце зберігання</b>	лаб. 4
<b>Дата</b>	<b>К-сть упаковок</b>	<b>шт в упаковці</b>	<b>lot №</b>
18.06.2018	10	30	4567
19.06.2018	5	40	7896
<b>Залишок</b>			10

*Рис. 3 Фрагмент реєстру надходження виробів із пластику*

Користувач програми може виконувати наступні дії:

- перегляд відомостей по кожному виду виробів та реагентів;
- пошук виробу або реагенту за його назвою;
- фільтрація виробів за назвою (українською та англійською), видом, виробником;
- створення/видалення нової підкатегорії для кожного виду виробу;
- можливість друку наступних відомостей:
  - деталізований звіт (звіт, що містить всю інформацію про вироби даного виду);
  - реєстр надходження за вказаний період (рис. 3);
  - реєстр списання за вказаний період.

Періоди звітності:

- місяць;
- квартал;
- рік.

Крім цього, користувач має можливість обирати необхідну дату, яку треба взяти за точку відліку.

### 6. Методи вирішення завдання

Для розробки даного продукту використовувалась мова програмування C++ [1] та середовище програмування Embarcadero C++ Builder XE7 [2]. Для збереження даних розроблено відповідну реляційну локальну базу даних формату SQL Server [3, 4]. Для реалізації роботи з базою даних обрано Microsoft SQL Server 2012 Express.

### **Висновки**

Основним результатом роботи є програма, що забезпечує оптимізацію роботи лікарів-лаборантів та персоналу медичної лабораторії дитячої лікарні.

Результати проведеної апробації програми вказують на високу ефективність її впровадження.

### **Список літератури**

1. Брюс Эккель. Философия C++. Введение в стандартный C++ / Эккель Брюс.– : Питер, 2004. – 577 с.
2. В. С. Романчик, А. Е. Люлькин. Программирование в C++ Builder. – Минск, 2007. – 128 с.
3. Документация по SQL Server. – [Електронний ресурс]. – Режим доступу: <https://goo.gl/VrWkUY>
4. Алан Бьюли. Изучаем SQL / Бьюли Алан. – : Символ-Плюс, 2007. – 312 с.

УДК 004.021

*КОРОБОВА Е.В.,  
СПЕРКАЧ М.О.*

## **ІНФОРМАЦІЙНА СИСТЕМА ПІДТРИМКИ ДІЯЛЬНОСТІ РОБОТИ СТАНЦІЇ ШВИДКОЇ ДОПОМОГИ**

Служба швидкої невідкладної допомоги відіграє досить важливу роль в житті кожної людини, оскільки її завдання — надати першу медичну допомогу у випадках, що можуть загрожувати життю. Саме тому робота станції має бути швидкою, злагодженою та ефективною.

Одна із складових ефективності — швидкість прибуття бригади, що залежить від багатьох факторів, в тому числі й від навантаження на диспетчерський пункт. Багато часу диспетчери приділяють тим викликам, що є непрофільними, тобто причина до виклику не потребує оперативного втручання лікарів.

Люди викликають швидку на високу температуру, кров з носа чи щоб виміряти тиск. Кількість таких викликів становить майже 80% [1]. Саме через те, що диспетчер зайнятий цими викликами, життя людини, що екстрено потребує допомоги саме в цей момент може бути під загрозою.

Проблема, що розглядається у роботі, пов'язана із великим навантаженням на диспетчерів, що може призвести до загрози життю людей.

Для вирішення цієї проблеми пропонується розробити математичну модель, що автоматизує процеси прийняття рішень на станції швидкої допомоги та інформаційну систему, що її реалізує.

Робота присвячена аналізу процесів станції швидкої допомоги, постановці задач, вибору та аналізу методів вирішення поставлених задач, а також аналізу результатів.

### **КЛЮЧОВІ СЛОВА**

Станція швидкої медичної допомоги; диспетчер; інформаційна система; автоматизація диспетчерського пункту; математична модель;

Emergency services play a very important role in every person's life, since it's job is to provide first aid in cases that could endanger life. That is why the station's work must be fast, coherent and efficient.

One of the components of efficiency is the speed of arrival of the brigade, which depends on many factors, including the load on the control center. Much time dispatchers pay attention to calls which the reason for the call does not require the prompt intervention of doctors.

People call to Ambulance because of high temperature, blood from the nose, or even just to measure pressure. The number of such calls is almost 80% [1]. It is why when dispatcher is busy with these challenges, the human life that is in urgent need of assistance at this moment may be at risk.

The problem addressed in the work is related to the high load on the dispatches, which may endanger the lives of people.

To solve this problem, it is proposed to develop a mathematical model that automates decision-making processes at the ambulance station.

The work is devoted to the analysis of the ambulance station's processes, the task setting, the choice and analysis of methods for solving the tasks, as well as the analysis of the results.

### **KEYWORDS**

Ambulance Station; dispatcher; information system; automation of dispatcher's work; mathematical model.

## Вступ

Для збереження життя багатьох людей, які залежать лише від вчасно наданої медичної допомоги, робота станції швидкої допомоги має бути миттєвою, злагодженою і головне елективною.

У цій роботі розглядається один із варіантів пришвидшення роботи станції за рахунок розробки такої математичної моделі, що дозволить автоматизувати прийняття рішень на всіх етапах процесу виклику швидкої допомоги.

### 1. Завдання станції швидкої медичної допомоги

Для того, щоб правильно автоматизувати роботу станції, необхідно розуміти її завдання. Основне завдання станції — надавати екстрену медичну допомогу пацієнтам за викликами, переданими оперативно-диспетчерською службою Центру, а також у разі особистого звернення осіб, які перебувають у невідкладному стані та потребують надання екстреної медичної допомоги, або працівників правоохоронних та аварійно-рятувальних підрозділів до працівників станції, підстанції (відділення), пункту постійного або тимчасового базування бригад [2].

Швидке та безпомилкове виконання цього завдання є основною метою розробки математичної моделі.

Для цього розглянемо коротко розглянемо процеси діяльності станції.

### 2. Процеси діяльності станції швидкої допомоги

*Диспетчер головної станції:*

1. Визначає причину виклику.

Причина виклику може бути визначена за допомогою запитань диспетчера та відповідей пацієнта. При чому зміст кожного наступного запитання має змінюватися, в залежності від отриманої відповіді. Результатом

алгоритму стає попередній діагноз, реєструється у карті як причина до виклику.

2. Дізнається всі деталі про місцезнаходження пацієнта та його особисту інформацію; вносить цю інформацію до карти виклику.

3. Оцінює критичність виклику за порядковою (ранговою) шкалою.

На основі визначеної раніше причини до виклику, диспетчер отримує пріоритет.

4. Визначає профіль необхідної медичної бригади для виконання виклику.

На основі причини до виклику, а також його пріоритетності необхідно визначити профіль бригади.

5. Визначає підстанцію, якій необхідно передати даний виклик.

6. Передає карту на обрану підстанцію.

*Диспетчер підстанції:*

1. Отримує карту виклику із головної станції.

2. Визначає бригаду, а також планує розклад її роботи.

Бригада має визначатися з урахуванням вже складеного розкладу її роботи, та отриманий вхідний виклик повинен також бути внесеним до розкладу.

3. Передає виклик на мобільну станцію бригади.

*Медична бригада:*

1. Отримує оновлений розклад та карту виклику.

2. Приймає виклик, та оновлює свій статус на "зайнятий".

3. Після закінчення роботи над викликом заповнює карту виклику.

4. Відзначає свій статус як "вільний".

Навіть цей схематичний перелік свідчить про напружену цілодобову працю великої кількості людей для організації забезпечення найшвидшої медичної допомоги[3].

### 3. Математичні задачі

Таким чином, для розв'язку поставленої задачі необхідно вирішити наступні підзадачі:

1. Задача визначення причини виклику та його пріоритету
2. Задача вибору профіля бригади
3. Визначення бригади
4. Планування роботи бригади

Визначимо основний критерій ефективності роботи математичної моделі в цілому — найбільша швидкість прибуття бригади до потерпілого.

### 4. Визначення причини виклику

Основна мета розпізнавання полягає в тому, щоб визначити правило, відповідно до якого встановлюється до якого з класів відноситься об'єкт, що класифікується. Треба зазначити, що на даному етапі необхідно визначити лише первинний (попередній) діагноз, який впливає на профіль бригади та на пріоритет обслуговування виклику. Точний діагноз зможе визначити бригада, коли прибуде на виклик.

Визначення повного переліку симптомів (ознак), за якими необхідно класифікувати діагноз (об'єкт).

При цьому ознаки можуть бути детерміновані, ймовірнісні, логічні та структурні.

Детерміновані ознаки — ознаки, що можуть приймати лише конкретні числові значення.

Ймовірнісні ознаки — ознаки, чії значення є розподіленими по всіх об'єктах випадковим чином, при чому саме значення може бути визначено лише за допомогою експериментів.

Логічні ознаки — висловлювання, що можуть набувати лише двох значень — “Істина” чи “Брехня” з повною визначеністю. Це ті ознаки, що не мають кількісного вираження, вони можуть лише судити про наявність або відсутність ознак у об'єкті, що розглядається.

Структурні ознаки — непохідні елементи (символи) структури об'єкта.

У випадку визначення причини виклику, варто розглядати логічні ознаки, адже важлива лише наявність, або відсутність певного симптому, наприклад наявність втрати свідомості, та відсутність кровотечі. Втім, важливо зазначити, що наявність або відсутність симптомів у діагнозах повинна мати критерій важливості, що буде визначати важливість значення симптому в рамках цього діагнозу. Наприклад, у діагнозі “відкритий перелом кінцівки” можливий симптом “втрата свідомості”, але його наявність не є обов'язковою, в той час як для діагнозу “зупинка серця” він є одним із ключових симптомів.

Існує множина  $m$  можливих діагнозів  $D$ :

$$D = \{D_1, D_2, \dots, D_m\} \quad (1)$$

Існує множина  $i$  можливих симптомів  $X$ :

$$X = \{x_1, x_2, \dots, x_i\} \quad (2)$$

При цьому, кожен діагноз  $D_m$  характеризується наступним вектором, довжина якого рівна загальній кількості симптомів з множини  $X$ :

$$D_m = (v_1, v_2, \dots, v_i) \quad (3)$$

де  $v_i$  — значення, що визначає наявність чи відсутність симптому у складі діагнозу.

Кожному  $v_i$  відповідає коефіцієнт важливості симптому у складі поточного діагнозу  $\omega_i$ . Обмеження на коефіцієнт:

$$0 \leq \omega_i \leq 1 \quad (4)$$

Задачею диспетчера на даному етапі є визначення тих симптомів з множини  $X$ , що наявні у пацієнта, після чого отримуємо вхідний вектор симптомів виду:

$$X_p = (x_1^p, x_2^p, \dots, x_i^p) \quad (5)$$

для якого будуюмо вектор:

$$D_p = (v_1^p, v_2^p, \dots, v_i^p, \dots) \quad (6)$$

Наступним кроком визначимо новий вектор  $S$ , в якому кожний компонент вираховується за наступною формулою:

$$s_i = v_i^p \oplus v_i \quad (7)$$

Таким чином ми знаходимо такі симптоми між поточним діагнозом та діагнозом пацієнта, що відрізняються одне від одного. Тепер необхідно врахувати важливість симптому у складі діагнозу, для чого помножимо кожен компоненту вектора  $S$  на відповідний йому коефіцієнт  $\omega_i$ , та знайдемо суму всіх компонент вектора. В результаті треба обрати такий діагноз  $D_{\text{опт}}$ , що має мінімальне значення суми компонентів, тобто:

$$D_{\text{опт}} = \min \sum_{k=1}^i s_i \times \omega_i \quad (8)$$

Після того, як був встановлений  $D_{\text{опт}}$ , необхідно визначити пріоритет виклику  $w$ . Його значення знаходиться у межах:

$$0 \leq w \leq 1 \quad (9)$$

Для цього використовується таблиця відповідностей причини до пріоритету. Зміст цієї таблиці має бути складений медичними професіоналами.

### 5. Задача вибору профілю бригади

У структурі станції швидкої допомоги виділяють два типи профілів медичних бригад: лікарські та фельдшерські. При цьому лікарські поділяються на лінійні (загального профілю) і спеціалізовані. Бригада кожного профілю має чітко визначений список робіт (діагнозів), на виконання яких вони можуть виїхати. Позначимо множини робіт бригади:

$$P_{\text{лін}} = \{D_{\text{лін}1}, D_{\text{лін}2}, \dots, D_{\text{лін}k}\} \quad (10)$$

де  $P_{\text{лін}}$  — множина можливих робіт для лінійної бригади,  $k$  — кількість можливих робіт.

$$P_{\text{сн}} = \{D_{\text{сн}1}, D_{\text{сн}2}, \dots, D_{\text{сн}r}\} \quad (11)$$

де  $P_{\text{сн}}$  — множина можливих робіт для бригади загального спрямування,  $r$  — кількість можливих робіт.

$$P_{\text{ф}} = \{D_{\text{ф}1}, D_{\text{ф}2}, \dots, D_{\text{ф}n}\} \quad (12)$$

де  $P_{\text{ф}}$  — множина можливих робіт для бригади фельдшерського спрямування,  $n$  — кількість можливих робіт.

Таким чином, вибір бригади зводиться до того, щоб визначити в якій множині наявний визначений на попередньому кроці діагноз  $D_{\text{опт}}$ . Якщо на діагноз підходить декілька профілів, то обидва профілі обираються як прийнятні.

### 6. Визначення бригади

Для визначення бригади серед множини усіх бригад  $M$ , треба виділити лише ті, що підходять по профілям (одному чи декількох), що були отримані на попередньому кроці. Позначимо множину прийнятних бригад як  $M_{\Pi}$ :

$$M_{\Pi} = \{M_{\Pi\alpha}, M_{\Pi\beta}, \dots, M_{\Pi\alpha}\} \quad (13)$$

де  $\alpha$  — кількість бригад, що підходять за профілем для даного виклику. При цьому кожна бригада може мати вже складений розклад робіт (викликів), які вона має виконати  $M_{\Pi\alpha}^D$ :

$$M_{\Pi\alpha}^D = \{D_{\alpha 1}, D_{\alpha 2}, \dots, D_{\alpha l}\} \quad (14)$$

де  $l$  — кількість робіт, що вже заплановані у бригади.

Кожна робота  $D_{\alpha l}$  має свій пріоритет виконання  $w_{\alpha l}$ , а також орієнтовний час її виконання  $t_{\alpha l}$ .

Ще однією змінною, що знадобиться для визначення необхідної бригади — час прибуття на місце до потерпілого кожною бригадою  $t_{\alpha}^{Mn}$ , що визначається за допомогою онлайн-сервісів, що надають актуальну інформацію щодо маршрутів та стану доріг. Тоді загальний

орієнтовний час, що бригада має витратити на роботу  $D_{\alpha l}$ :

$$T_l^{Mn\alpha} = t_{\alpha}^{Mn} + t_{\omega} \quad (15)$$

Для того, щоб обрати бригаду, необхідно для кожної з них розрахувати час, який їй знадобиться для того, щоб дістатися потерпілого. Однак, у розрахунок часу необхідно враховувати лише ті роботи, що мають пріоритет вищий (або рівнозначний) за той, що був присвоєний новій роботі. Позначимо множину таких робіт  $D_w^{Mn\alpha}$ :

$$D_w^{Mn\alpha} = \{D_{w1}^{Mn\alpha}, D_{w2}^{Mn\alpha}, \dots, D_{ws}^{Mn\alpha}\} \quad (16)$$

Тоді час, що необхідний для бригади  $\alpha$  щоб дістатися до потерпілого:

$$T_p^{Mn\alpha} = \sum_{y=1}^s t_{ws}^{Mn\alpha} \quad (17)$$

де  $t_{ws}^{Mn\alpha}$  — час, що потрібен бригаді  $M_{P\alpha}$  для виконання роботи  $D_{ws}^{Mn\alpha}$ .

Розрахувавши час на прибуття для кожної бригади, обираємо ту, що має мінімальне значення загального часу:

$$T_{opt} = \min \{T_p^{Mn\alpha}\} \quad (18)$$

## 7. Планування роботи бригади

Таким чином, новий вхідний виклик повинен додатися до плану робіт бригади наступним за  $D_{ws}^{Mn\alpha}$ .

Саме такий підхід зможе забезпечити виконання викликів якнайшвидше та з урахуванням критерію важливості.

## 8. Розробка алгоритму розв'язання

**Крок 1.** ВИЗНАЧИТИ список симптомів.

**Крок 2.** ОБЧИСЛИТИ вектор характеристик для даного діагнозу, використовуючи значення 0 чи 1, в залежності від того, чи наявний симптом (5).

**Крок 3.** ОБЧИСЛИТИ на основі отриманого вектору попередній діагноз з кроку 2 (8).

**Крок 4.** ВИЗНАЧИТИ пріоритет виклику на основі інформації про попередній діагноз з кроку 2.

**Крок 5.** ВИЗНАЧИТИ профілі бригади, що можуть виконати даний вид робіт (10-12).

**Крок 6.** ВІДФІЛЬТРУВАТИ із загальної множини бригад ті, що не підходять по профілю.

**Крок 7.** ОБЧИСЛИТИ для кожної з бригад орієнтовний час на прибуття (17).

**Крок 8.** ОБРАТИ найменший з розрахованих часів (18).

**Крок 9.** ВИЗНАЧИТИ новий розклад робіт для бригади, що була обрана на кроці 8.

На цьому робота алгоритму завершується.

## Висновок

У даній роботі було наведено завдання станції швидкої допомоги, що потребують оптимізації. Розглянуті та проаналізовані бізнес процеси, з метою розробки математичної моделі для їх автоматизації. Була наведена математична модель для декількох процесів: визначення попереднього діагнозу, вибору профілю бригади, вибору бригади, а також визначення нового розкладу робіт медичної бригади. Тестування розробленої математичної моделі було проведено покроково, згідно з наведеним алгоритмом розв'язання задачі.

### Список літератури

1. Как работают фельдшеры "скорой" [Електронний ресурс] // Подробности. – 2018. – Режим доступу до ресурсу: <http://podrobnosti.ua/2251498-vyzovy-ot-skuki-i-ekskursii-dlja-pjanyh-kak-rabotajut-feldshery-skoroj.html>.
2. Служба екстреної (швидкої) медичної допомоги України [Електронний ресурс] – Режим доступу до ресурсу: [https://pidruchniki.com/76858/meditsina/sluzhba\\_ekstrenoyi\\_shvidkoyi\\_medichnoyi\\_dopomogi\\_ukrayini](https://pidruchniki.com/76858/meditsina/sluzhba_ekstrenoyi_shvidkoyi_medichnoyi_dopomogi_ukrayini).
3. Про заходи щодо удосконалення надання екстреної медичної допомоги населенню в Україні [Електронний ресурс] // МІНІСТЕРСТВО ОХОРОНИ ЗДОРОВ'Я УКРАЇНИ. – 2008. – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki>.



УДК 519.8

КРАСНИКОВ С.О.,  
ГУЛЯНИЦЬКИЙ Л.Ф.

## ПІДХІД ДО НАЛАШТУВАННЯ ПАРАМЕТРІВ АЛГОРИТМУ ІМІТАЦІЙНОГО ВІДПАЛУ ДЛЯ РОЗВ'ЯЗУВАННЯ ЗАДАЧ КОМБІНАТОРНОЇ ОПТИМІЗАЦІЇ

У статті розглянуто підхід до налаштування параметрів алгоритму імітаційного відпалу для розв'язування задач комбінаторної оптимізації. Наводиться опис алгоритму імітаційного відпалу з його ключовими аспектами а також підхід до налаштування. На прикладі задачі комівояжера знаходяться значення параметрів алгоритму імітаційного відпалу та порівнюється застосування таких параметрів у порівнянні з початковими значеннями параметрів за критерієм часу та точністю розв'язку.

**КЛЮЧОВІ СЛОВА:** алгоритм імітаційного відпалу, детермінований локальний пошук, задача комівояжера, параметри алгоритму.

The subject of the article is an approach to adjust the parameters of the simulated annealing algorithm for solving combinatorial optimization problems. The description of the simulated annealing algorithm with its key aspects as well as the approach to configuring is presented. In the example of the traveling salesman problem, the value of the parameters of the simulated annealing algorithm is found and the comparison of the application of such parameters with the initial values of the parameters by the criterion of time and accuracy of the solution is compared.

**KEYWORDS:** simulated annealing algorithm, deterministic local search, traveling salesman problem, parameters of the algorithm

### 1. Вступ

Задачі комбінаторної оптимізації є широко розповсюдженими в сферах життя: медицина, фабрика, перевозки товарів і т.д. Оскільки масштаби цих є великими для них потрібно мати алгоритм для знаходження оптимального розв'язку. Однак всі задачі комбінаторної оптимізації є NP-складними і для їх розв'язування точні методи не є практичними, оскільки вони є трудомісткими. Тому найбільшій популярності серед алгоритмів розв'язування задач комбінаторної оптимізації посіли наближені алгоритми. Серед наближених алгоритмів особливе місце посідають алгоритми стохастичного локального пошуку. Вони використовуються як допоміжні для інших алгоритмів, наприклад генетичний або мурашиний алгоритми. Однак для їх налаштування потрібно використовувати різні методи в тому числі і експертів. Дані методи є можуть потребувати немало часу для налаштування. В даній статті наводиться підхід для налаштування параметрів алгоритму імітаційного відпалу. Налаштування буде проводитись при розв'язуванні задачі комівояжера, як

поширеного представника задач комбінаторної оптимізації, постановку якої далі наведемо.

### 2. Постановка задачі комівояжера

Маємо  $N$  міст. Між кожним містом відома відстань  $C \in R^{N \times N}$ . Треба побудувати такий маршрут  $X_* \in R^{N \times N}$ , який проходить кожне місто 1 раз, крім початкового міста, та має мінімальну довжину.

Математична постановка задачі комівояжера може бути подана так:

знайти

$$X_* = \arg \min_{x_{ij} \in X \in R^{N \times N}} c_{ij} x_{ij}, \quad (1)$$

за обмежень

$$\sum_{i=1}^N x_{ij} = 1, j = \overline{1, N}, \quad (2)$$

$$\sum_{j=1}^N x_{ij} = 1, i = \overline{1, N}, \quad (3)$$

$$x_{ij} \in \{0,1\}, \quad (4)$$

де  $X_*$  - маршрут для обходу усіх міст;  
 $x_{ij}$  - наявність переходу від міста  $i$  до міста  $j$  у маршруті;  
 $c_{ij}$  - відстань (вага) переходу від міста  $i$  до міста  $j$ ;  
 $N$  – кількість міст.

Формула (1) вказує на цільову функцію – знаходження такого маршруту, при якому його довжина буде мінімальною.

Обмеження (2) та (3) вказують на умову відвідування міста тільки 1 раз.

Дана задача має багато модифікацій, серед яких є:

- задача кур'єра;
- кластерна задача комівояжера;
- задача комівояжера з декількома комівояжерами;
- задачі маршрутизації транспортних засобів.

Детальніше про модифікації задачі подано в [1].

Для розв'язування подібних задач використовуються наближені алгоритми, серед яких популярними є алгоритми локального пошуку. Ці алгоритми поділяються на детерміновані та стохастичні, перші з них розглянемо детальніше.

### 3. Алгоритм детермінованого локального пошуку

Детермінований локальний пошук є найбільш простим з класів наближених алгоритмів. Суть його роботи полягає у виборі тільки кращих варіантів розв'язку у околі поточного варіанту.

Типовим представником даного класу алгоритмів є метод вектору спаду, запропонований І.В. Сергієнком у 1964 році [2]. Вхідними даними для алгоритму є початковий варіант розв'язку  $x$ . Псевдокод алгоритму виглядає так.

```

procedure RecessionVectorMethod ( $x$ )
begin
     $x_{parent} = x$ ;
    while (не виконується умова завершення) do
         $x_{child} =$  Згенерувати Новий Варіант Розв'язку з  $x_{parent}$ ;

```

```

 $\Delta = f(x_{parent}) - f(x_{child})$ ;
if ( $\Delta > 0$ ) then
     $x_{parent} = x_{child}$ ;
endif
endwhile
return  $x_{parent}$ ;
end

```

Тут  $f(x)$  – функція визначення якості розв'язку  $x$ .

Ключовими аспектами реалізації алгоритму є:

- умова завершення роботи алгоритму;
- генерація нових варіантів розв'язку;
- обчислення функції  $f(x)$ .

В якості генератора нових варіантів розв'язку може використовуватись кільцевий генератор з використанням оператора 2-заміни [3].

Цей клас алгоритмів забезпечує в загальному випадку, знаходження до локального, а не глобального оптимуму. Для уникнення передчасної збіжності (потрапляння в локальний розв'язок) застосовують різні стратегії. Для досягнення глобального оптимуму будемо використовувати інший клас алгоритмів – алгоритми стохастичного локального пошуку, відомим представником якого є алгоритм імітаційного відпалу.

### 4. Алгоритм імітаційного відпалу

Алгоритм імітаційного відпалу (АІВ) відноситься до класу алгоритмів стохастичного локального пошуку. Цей клас алгоритмів використовує імовірнісну процедуру, що використовується для переходу не тільки до кращого варіанту розв'язку в околі, а і до гіршого.

В АІВ ймовірність переходу  $p$  з одного стану в інший базується на процесі відпалювання та формули Больцмана-Гіббса (5):

$$p = e^{-\frac{\Delta E}{kT}} \quad (5)$$

де  $\Delta E$  – зміна узагальненої енергії;  
 $T$  – температура;  
 $K$  – стала Больцмана.

Ймовірність переходу з одного варіанту розв'язку в інший  $p$  в АІВ виглядає наступним чином:

$$p = e^{-\frac{\Delta}{T}} \quad (6)$$

де  $\Delta = f(y) - f(x)$ .

В якості початкових даних для АІВ є початковий варіант розв'язку  $x$  та початкова температура  $T_0$ . Псевдокод алгоритму виглядає так.

```

procedure SA ( $x, T_0$ )
begin
   $T_{current} = T_0$ ;
   $x_{parent} = x$ ;
   $x_{best} = x$ ;
  while (не виконується умова завершення
роботи алгоритму) do
    while (не виконується умова
рівноваги) do
       $x_{child}$  = Згенерувати Новий
Варіант Розв'язку з  $x_{parent}$ ;
       $\Delta = (f(x_{child}) - f(x_{parent}))$ ;
      if ( $\Delta < 0$ ) then
         $x_{parent} = x_{child}$ ;
        if ( $f(x_{best}) > = f(x_{child})$ )
then
           $x_{best} = x_{child}$ ;
        endif
      endif
    else
       $p$  = Знайти Імовірність
Переходу За Формулою (6);
       $\xi = \text{rand}[0, 1]$ ;
      if ( $p >= \xi$ ) then
         $x_{parent} = x_{child}$ ;
      endif
    endelse
  endwhile
   $T_{current}$  = Зменшити
Температуру( $T_{current}$ );
endwhile
return  $x_{best}$ ;
end

```

де  $f(y)$  – функція визначення якості розв'язку  $y$ ;

$\text{rand}[a,b]$  – генератор випадкових чисел відрізка від  $a$  до  $b$ .

Ключовими аспектами для даного алгоритму є:

- умова закінчення роботи алгоритму;
- умова рівноваги;
- генерація нових варіантів розв'язку;
- обчислення функції  $f(x)$ .
- процедура зменшення температури.

В якості умови закінчення роботи алгоритму може використовуватись максимальна кількість ітерацій, при якій не було зміни рекорду, або мінімальна температура  $T_{min}$ .

В якості умови рівноваги можуть використовуватись такі умови:

- за визначеною кількістю пройдених варіантів розв'язку;
- за визначеною кількістю переходів;
- метод прогонів та переходів [4].

В якості процедури зменшення температури використовують поширену формулу мультиплікативного зменшення

$$T_{t+1} = \alpha T_t$$

де  $\alpha$  – коефіцієнт зміни температури  $\alpha \in [0.9; 0.99]$ .

3 ключових аспектів алгоритму впливають ключові параметри алгоритму:

- початкова температура  $T_0$ ;
- мінімальна температура  $T_{min}$ ;
- коефіцієнт зміни температури  $\alpha$ ;
- параметри умов рівноваги:
  - за визначеною кількістю пройдених варіантів розв'язку – кількість пройдених варіантів розв'язку;
  - за визначеною кількістю переходів – кількість переходів;
  - метод прогонів та переходів – кількість переходів та потрібна точність досягнення прогонів.

Після визначення ключових параметрів АІВ перейдемо до налаштування даних параметрів.

## 5. Підхід до налаштування параметрів АІВ

В якості підходу до налаштування параметрів буде використовуватись детермінований локальний пошук. Для налаштування параметрів алгоритму потрібно мати алгоритм  $A$ , для якого налаштовуються параметри, початкові значення параметрів алгоритму  $A$   $x_{begin} = (x_{1begin}, \dots, x_{nbegin})$ , де  $n$  – кількість параметрів, межі зміни параметрів алгоритму  $A$   $x_{min} = (x_{1min}, \dots, x_{nmin})$  та  $x_{max} = (x_{1max}, \dots, x_{nmax})$ , крок зміни параметрів алгоритму  $A$   $\Delta x = (\Delta x_1, \dots, \Delta x_n)$  та початкові варіанти розв'язків  $r = (r_1, r_2, r_3)$ .

Даний підхід базується на методі вектору спаду. Спочатку генеруємо нові значення параметрів алгоритму  $x$  після чого розв'язуємо задачу з використанням АІВ і передаємо по черзі початкові варіанти розв'язків  $r$ . Для кожного  $i$  варіанту розв'язку, отриманого з алгоритму  $r_{ibest}$ , визначаємо довжину  $l_i$ . Далі знаходимо середнє значення довжини варіантів розв'язку  $l_{average}$ . Якщо  $l_{average}$  менше за відоме середнє значення довжини варіантів розв'язку, то запам'ятовуємо його та параметри, при яких отримані дані середні значення. Так повторюємо доки не переберемо усі варіанти зміни кращих параметрів. Псевдокод алгоритму виглядає так:

```

procedure AlgorithmAdjustParameters(A,
   $x_{begin}$ ,  $x_{min}$ ,  $x_{max}$ ,  $\Delta x$ ,  $r$ )
  begin
     $x_{parent} = x_{begin}$ ;
     $l_{best\_average} = \infty$ ;
    while (не пройдені усі можливі
      варіанти зміни параметрів  $x_{parent}$ ) do
       $x_{child} =$  Згенерувати Нові Значення
      Параметру ( $x_{parent}$ ,  $x_{min}$ ,  $x_{max}$ ,  $\Delta x$ );
      for (i = 0; i < |r|; i++) do
         $r_{ibest} = A(r_i, x_{child})$ ;
         $l_i = f(r_{ibest})$ ;
      endfor

       $l_{average} = \sum_{i=0}^{|r|} l_i / |r|$ 

      if ( $l_{best\_average} > l_{average}$ ) then
         $x_{parent} = x_{child}$ ;
         $l_{best\_average} = l_{average}$ ;
  
```

```

endf
endwhile
return  $x_{best}$ ;
end
  
```

де  $f(y)$  – функція визначення якості розв'язку  $y$ ;

$|r|$  - довжина вектору  $r$ .

В якості генератора параметрів використовувався кільцевий генератор з одиничним радіусом.

Застосуємо наведений підхід налаштування параметрів алгоритму на відповідній задачі.

## 6. Експериментальні результати дослідження

Для експерименту налаштування параметрів алгоритмів використаємо задачу eil101.tsp (101 місто), взяту з міжнародної бібліотеки задач TSPLIB [5]. Для цієї задачі згенеровано 3 початкових варіанти розв'язку.

Експерименти проводились на ноутбучі ASUS S56C з характеристиками:

- процесор - Intel Core i5-3317U 1.7ГГц (розганяється до 2.5ГГц), 2 ядра, 4 потоки;
- оперативна пам'ять - 8Гб.

Алгоритми були реалізовані на мові програмування Java.

Налаштування відбувались для трьох варіацій в АІВ умови рівноваги:

- за кількістю пройдених варіантів розв'язку;
- за кількістю переходів;
- метод прогонів та переходів.

Для експериментального запуску алгоритму підбору параметрів АІВ використовуються наступні межі значень алгоритмів:

- початкова температура  $T_0$ : від 1 до 10 з кроком 0.5;
- коефіцієнт зміни температури  $\alpha$ : від 0.9 до 0.99 з кроком 0.01;
- мінімальне значення температури  $T_{min}$ : від 0.001 до 0.01 з кроком 0.001;
- кількість ітерацій: від 55 % від кількості сусідів для варіанту розв'язку до 100% з кроком 5%;

- кількість переходів: від 2 % від кількості сусідів для варіанту розв'язку до 20% з кроком 2%;
- потрібна точність досягнення прогонів: від 2 % від кількості сусідів для варіанту розв'язку до 20% з кроком 2% та точність досягнення прогонів від 0.001 до 0.01 з кроком 0.001.

В якості початкових значень параметрів алгоритму взяті середні значення із вказаних вище меж, які наводяться в таблиці 1.

**Табл. 1. Початкові параметри реалізації AIB для налаштування**

Параметр	SA з пройденими варіантами розв'язку	SA з переходами	SA з прогонами та переходами
$T_0$	6	6	6
$\alpha$	0.95	0.95	0.95
$T_{min}$	0.006	0.006	0.006
Parameter 1*	0.8	0.12	0.12
Parameter 2*	-	-	0.006

\*Примітка: в якості Parameter 1 для AIB з пройденими варіантами розв'язку взято відношення пройдених варіантів розв'язку при одному значенні температури до кількості сусідів для одного варіанту розв'язку, для AIB з переходами та AIB з прогонами та переходами – це відношення

кількості переходів при одному значенні температури до кількості сусідів для одного варіанту розв'язку, Parameter 2 – потрібна точність досягнення прогонів.

Результати налаштування параметрів трьох реалізацій AIB для задачі eil101.tsp наведено в таблиці 2.

**Табл. 2. Налаштовані параметри реалізації AIB**

Параметр	SA з пройденими варіантами розв'язку	SA з переходами	SA з прогонами та переходами
$T_0$	9	6	3
$\alpha$	0.99	0.99	0.98
$T_{min}$	0.001	0.001	0.001
Parameter 1*	0.85	0.14	0.14
Parameter 2*	-	-	0.008
Average Length	636.3333	629	629
Time finding	34s	2m 3s	4m 13s

Отримані параметри використаємо для розв'язування задач меншої розмірності та порівняємо з початковими значеннями параметрів для алгоритмів. Для цього будемо використовувати мультистарт (запускати задачу по 10 разів) та обирати серед них найкращий варіант за довжиною маршруту та часом. Результати наведені в таблиці 3.

Тут: SA (1) – AIB з пройденими варіантами розв'язку;

SA (2) – AIB з переходами;

SA (3) – AIB з методом прогонів та переходів;

$n$  – кількість міст в задачі комівояжера;

$f^*$  – найменша довжина варіанту розв'язку;

$\varepsilon$  – відносна похибка яка обчислюється за формулою (8);

$$\varepsilon = (f - f^*)/f^* \quad (7)$$

$t$  – час знаходження даного розв'язку (в секундах);

$\varepsilon_{mean}$  – середнє значення похибок  $\varepsilon$ .

Як впливає з таблиці 3, алгоритм покращує свою точність за допомогою оптимізованих параметрів. Це можна побачити за параметром середнього значення похибок  $\varepsilon_{mean}$ . Найкращим серед трьох варіацій AIB є AIB з умовою рівноваги за метод прогонів та перегонів.

Табл. 3. Отримані результати для двох варіантів параметрів

n	f*	SA (1)				SA(2)				SA(3)			
		Початкові		Налаштовані		Початкові		Налаштовані		Початкові		Налаштовані	
		$\varepsilon$	t	$\varepsilon$	t	$\varepsilon$	t	$\varepsilon$	t	$\varepsilon$	t	$\varepsilon$	t
17	39	0,0	0,01	0,0	0,1	0,0	0,01	0,0	0,09	0,0	0,01	0,0	0,05
29	2020	0,3	0,01	0,0	0,17	0,0	0,01	0,0	0,06	0,0	0,05	0,0	0,05
33	1286	24	0,01	12,8	0,16	14,5	0,04	7,5	0,06	8,1	0,07	5,4	0,89
35	1473	19,7	0,03	6,4	0,17	17,4	0,02	2,7	0,09	6,6	0,06	3,8	0,58
42	1273	1,6	0,03	0,0	0,19	0,0	0,02	0,0	1,26	0,0	0,08	0,0	1,27
43	5620	4,8	0,02	0,8	0,21	5,2	0,01	0,7	0,11	5,1	0,01	0,9	0,05
44	1613	41,0	0,02	25,0	0,23	25,0	0,04	9,5	1,57	7,5	0,13	3,6	1,59
47	1776	50,4	0,03	25,8	0,25	21,7	0,03	14,7	1,72	12,8	0,14	16,9	2,44
48	14422	11,9	0,02	1,8	0,25	10,5	0,01	0,6	0,19	4,9	0,05	0,6	0,28
51	426	16,2	0,02	0,0	0,23	3,8	0,01	0,0	2,32	3,8	0,05	0,0	2,20
52	2059	312,5	0,02	266,3	0,23	297,2	0,01	266,3	0,43	290,1	0,04	266,3	0,66
53	6905	61,1	0,03	23,9	0,31	29,4	0,03	17,1	0,51	35,6	0,1	17,0	0,35
55	1608	60,7	0,05	25,1	0,33	45,3	0,02	18,2	2,23	17,8	0,19	18,2	2,76
64	1839	98,0	0,05	41,9	0,45	54,4	0,03	28,3	4,59	29,7	0,15	16,8	0,41
76	538	68,0	0,04	0,6	0,43	49,1	0,01	0,0	3,30	39,6	0,06	0,0	3,60
101	629	121,1	0,10	1,6	0,79	134,5	0,02	0,0	4,38	111,8	0,09	0,0	5,07
$\varepsilon_{mean}$		55,71		27		44,25		22,85		35,84		21,84	

### Висновки

У результаті проведеного дослідження запропоновано підхід до налаштування параметрів алгоритмів локального пошуку. Цей підхід застосований для налаштування параметрів АІВ для розв'язування задачі комівояжера. В подальшому планується використати підхід для налаштування інших алгоритмів комбінаторної оптимізації, застосувати цей підхід при розв'язуванні інших задач комбінаторної оптимізації, а також використати в підході модифікацію того ж АІВ.

### Перелік посилань

1. Gutin G., Punnen A. P. The traveling salesman problem and its variations. – Springer, USA, 2006. – 830 p.
2. Сергиенко И. В., Математические модели и методы решения задач дискретной оптимизации. – Киев, Наукова думка, 1988. – 472 с.
3. Talbi El-Ghazali Metaheuristic From Design to implementation. – Wiley, 2009. – 618 с.
4. Гуляницький Л.Ф., Мулеса О. Ю. Прикладні методи комбінаторної оптимізації. – Київський університет ВПІ, 2016. – 146 с.
5. TSPLIB - IWR, Heidelberg - Uni Heidelberg. – Режим доступа: <https://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/> – дата звернення 14.11.2018. – Заголовок с екрану.

УДК 004.942

ЛАНЬКО В.В.

## ВИРІШЕННЯ ЗАДАЧІ АВТЕНТИФІКАЦІЇ КОРИСТУВАЧА ЗА РУХАМИ КУРСОРА МИШІ З ВИКОРИСТАННЯМ ЛІНГВІСТИЧНОГО ПІДХОДУ

У даній статті розглянуто вирішення задачі автентифікації користувача за рухами курсора миші з використанням лінгвістичного підходу. Було проаналізовано лінгвістичний алгоритм, а саме підзадача отримання різницевих рядів, інтервалізація, лінгвістизація та підзадача побудови матриці переходів для біометричної автентифікації користувача на основі траєкторії руху курсору миші користувача. Головною метою даної системи є підвищення надійності та захисту електронно цифрових сервісів від зловмисників.

**КЛЮЧОВІ СЛОВА:** АВТЕНТИФІКАЦІЯ, БІОМЕТРИЧНА АВТЕНТИФІКАЦІЯ, ЛІНГВІСТИЧНЕ МОДЕЛЮВАННЯ, ІНТЕРВАЛЬНИЙ ПІДХІД, РОЗПІЗНАВАННЯ ОБРАЗІВ, МАТРИЦЯ ПЕРЕХОДІВ.

This article discusses the problem of user authentication by mouse movement using a linguistic approach. The linguistic algorithm was analyzed, namely, the subtask of obtaining difference lines, intervaling, linguistics, and the subtask of constructing a matrix of transitions for biometric authentication of the user based on the trajectory of the user's mouse movement. The main objective of this system is to increase the reliability and protection of electronic digital services from intruders.

**KEYWORDS:** AUTHENTICATION, BIOMETRIC AUTHENTICATION, LINGUISTIC MODELING, INTERVAL APPROACH, RECOGNITION OF CHARACTER, TRANSITIONS MATRIX.

### 1. Вступ

В наш час, велика кількість сервісів які поліпшують життєдіяльність людини, зокрема робота з банківськими операціями, документообіг, взаємодії між працівниками на підприємстві, та інші, перейшли в цифровий простір для спрощення та пришвидшення процесу взаємодії з користувачами. Разом із цим переходом, зросла необхідність у надійному захисті від зловмисників, які різними способами намагаються нанести шкоди, як тому чи іншому сервісу, так і звичайним користувачам. Найважливішим етапом у побудові непорушної системи безпеки є ідентифікація користувачів системи. Стандартні методи розпізнавання людини за допомогою логіну та паролю з кожним днем продовжують показувати свою вразливість до атак, зловмисники винайшли безліч способів обходу цієї системи. На порятунок приходять біометричні методи ідентифікації особи, які основані на її фізіологічних та поведінкових

характеристиках. На сьогодні розрізняють два методи біометричного розпізнавання, статистичний та динамічний. Перший базується на фізіологічних, даних від народження, характеристиках людини, у той час, як другий ґрунтується на поведінкових характеристиках, тобто на підсвідомих рухах людини.

### 2. Мета роботи

Метою даної статті є опис методу розв'язання задачі автентифікації користувача за рухами курсора миші з використанням лінгвістичного підходу.

### 3. Аналіз попередніх досліджень та публікацій

На даний момент існує невелика кількість досліджень присвячених розв'язанню задачі розпізнавання та ідентифікації користувача за траєкторіями руху курсору миші. Далі буде зроблений огляд, аналіз та виявлення недоліків декількох із них.

***Робота Шивані Хаши, Кріса Поллета й Марка Стампа в державному університеті Сан Хосе, США.***

Ця робота присвячена активному методу розпізнавання та автентифікації користувача [1-2]. Процес побудови моделі користувача полягає в тому, що на екрані одна за одною з'являються десять точок, у які користувачу потрібно перемістити курсор. Точка зникає після потрапляння курсора в неї, після чого з'являється наступна в довільному місці на екрані. Цей сценарій потрібно повторити 20 разів.

По отриманим координатах курсора обчислюються середнє значення координат, дисперсія, мінімальне та максимальне значення кожного параметру, які зберігаються у файлі. Потім, під час повторного входу користувачем у систему ці дані використовуються для верифікації.

Процес верифікації проводиться за тим самим сценарієм із десятима точками, який використовувався на початку. Під час чого формується вектор параметрів, які перевіряються з тими стандартними відхиленнями, що були сформовані на етапі реєстрації.

Система була протестована на 15 користувачах. Похибка склала 20%.

Автори цього дослідження також повідомили, що тестували модель під час методу пасивної автентифікації, але результатів цієї спроби так і не опублікували.

Недоліки дослідження:

- проблема активної автентифікації в тому, що користувачу треба пройти деякий сценарій, що негативно впливає на процес роботи користувача із системою
- розпізнавання користувача проходить лише на початку роботи із системою
- використано всього два статичні параметри

- модель було протестовано на невеликій кількості користувачів
- велика похибка

***Робота Адама Вайса, Еніла Рампанікера, Пранава Шаха, Шінеса Нобла й Ларі Іммора (Adam Weiss, Anil Ramapanicker, Pranav Shah, Shinesa Noble and Larry Immoehr), університет Пасе, США.***

Автори провели дослідження можливості активної автентифікації користувача за визначеним сценарієм [3]. Загальний вид системи дуже простий. У вікні відображались 25 кнопок у вигляді матриці, розмірність якої 5 на 5, і користувач повинен натискати на ці кнопки за деяким сценарієм.

Під час проходження сценарію формувалися такі параметри, як кількість точок траєкторії, її довжина, сумарна тривалість, середня швидкість руху курсору, кут його руху, прискорення та тривалість натискання кнопки. Модель користувача створювали із середніх значень та дисперсії вхідних параметрів.

Для ідентифікації користувача серед зареєстрованих відомих моделей використовувався метод найближчого сусіда, на основі Евклідової метрики.

Недоліки:

- невелика кількість користувачів під час тестування - 5
- проблема активної автентифікації в тому, що користувачу треба пройти деякий сценарій, що негативно впливає на процес роботи користувача із системою
- розпізнавання користувача проходить лише на початку роботи із системою
- неможливе використання на великій кількості користувачів
- під час тестів, не використовувалися дані від нових або незареєстрованих користувачів



#### 4. Побудова лінгвістичної моделі

Задача знаходження лінгвістичного образу траєкторії руху курсору миші користувача включає в собі вирішення таких підзадач [4-6]:

- обчислення різницевого ряду траєкторії руху курсора
- інтервалізація ряду
- побудова лінгвістичного ланцюжка
- побудова матриці переходів для кожної пари символів в отриманому ланцюжку

Вхідні дані - часовий ряд пар координат курсора по осі абсцис та ординат.

Вихідні дані - множини інтервалів для кожного параметру та відповідні матриці переходів.

Методів розв'язання задач розпізнавання можна розділити на два класу. Перший клас - дискримінантний підхід, у процесі якого, об'єкти класифікують за деякими ознаками які представлені у вигляді набору чисел. Другий клас використовує синтаксичний підхід, у якому важлива структура образу, а під час розпізнавання визначається класифікація об'єкта та надається інформація про неможливість віднесення його до іншого класу.

Використання першого класу починає програвати другому на складних об'єктах, які мають велику кількість ознак та характеристик, що ми бачимо при розпізнаванні динамічних образів.

У синтаксичному підході представлення образу формується з простих часткових образів, за аналогією із синтаксисом мов, у яких речення складаються зі слів, а слова із літер. У процесі розпізнавання виявляється, чи є "речення", яке описує об'єкт, синтаксично правильним згідно з заданою граматику.

#### 5. Підзадача отримання різницевого ряду

Призначенням цієї підзадачі є формування рядів, які характеризують динаміку зміни траєкторії руху курсору миші. Зокрема, переміщення, швидкість, прискорення тощо.

Дано: Вектор цілих чисел  $\bar{X}$  з потужністю  $n = |\bar{X}|$ .

Результати: Вектор цілих чисел  $\bar{D}$  з потужністю  $k = |\bar{D}|$ .

Обмеження:

$$\forall d_i \in \bar{D} : d_i = x_{i+1} - x_i \quad (1.1)$$

$$\text{де } i \in [0; n - 1]; x_{i+1}, x_i \in \bar{X} \\ k = n - 1 \quad (1.2)$$

#### 6. Підзадача інтервалізації

Призначенням цієї підзадачі є побудова алфавіту шляхом розбиття отриманого різницевого ряду на множину непересічних інтервалів, у якому кожний інтервал відповідає певній літері алфавіту.

Дано:

- гіпотетична потужність алфавіту  $a$ ;
- вектор цілих чисел  $\bar{D}$  з потужністю  $k = |\bar{D}|$ .

Результати:

Вектор пар цілих значень  $\bar{I}$  з потужністю  $n = |\bar{I}|$ .

Обмеження:

$$\forall x \in \bar{I} : x^1 \leq x^2 \quad (1.3)$$

$$\forall x_i, x_{i+1} \in \bar{I} : x_i^2 < x_{i+1}^1, \\ \text{де } i \in [0; n - 1] \quad (1.4)$$

$$n \leq a \quad (1.5)$$

$$a \ll k \quad (1.6)$$

$$\exists x \in \bar{I} : \forall d \in \bar{D}, d \in [x^1; x^2] \quad (1.7)$$

$$\forall d_i, d_{i+1} \in \bar{D} : d_i \leq d_{i+1}, \\ \text{де } i \in [0; k - 1] \quad (1.8)$$

$$x_0 \in \bar{I} : x_0 = (-\infty; x_1^1) \quad (1.9)$$

$$x_n \in \bar{I} : x_n = (x_{n-1}^2; +\infty) \quad (1.10)$$

Алфавіт ділиться на 3 підмножини:

- підмножина літер, які відповідають від'ємним значенням різницевого ряду;

- нуль-символ, який відповідає нулям різницевого ряду;
- підмножина літер, які відповідають додатнім значенням різницевого ряду.

Існує декілька правил згідно з якими відбувається формування інтервалів алфавіту:

- рівнозначне розбиття, при якому кожний інтервал має однакову довжину;
- рівномірне розбиття, при якому відбувається рівномірний розподіл елементів ряду по інтервалах.

### 7. Підзадача лінгвістизації

Призначенням цієї підзадачі є формування лінгвістичного ланцюжку методом підстановки відповідної літери з раніше отриманого алфавіту для кожного елементу різницевого ряду. Важливо зауважити, що виникатимуть такі ситуації, коли значення різницевого ряду не будуть потрапляти в жоден інтервал алфавіту, тобто, виходити за його межі, у такому випадку відбувається розширення множини інтервалів, як на додатній площині (від максимального значення до плюс нескінченності), так і на від'ємній (від мінімального значення до мінус нескінченності).

Дано:

- вектор цілих чисел  $\bar{D}$  з потужністю  $k = |\bar{D}|$ , що обмежений за формулою 1.7;
- вектор пар цілих значень  $\bar{I}$  з потужністю  $n = |\bar{I}|$  з обмеженнями, що наведені у формулах 1.3, 1.4, 1.9 та 1.10.

Результати:

Вектор цілих чисел  $\bar{A}$  з потужністю  $k$ .

Обмеження:

$$\forall x_i \in \bar{A} : \exists d_i \in \bar{D}, \exists y_j \in \bar{I}, d_i \in [y_j^1; y_j^2], x_i = j, \text{ де } i \in [0; k), j \in [0; n) \quad (1.11)$$

### 8. Підзадача побудови матриці переходів

Призначенням цієї підзадачі є формування матриці переходів між сусідніми літерами алфавіту в раніше отриманому реченні.

Дано:

- вектор цілих чисел  $\bar{A}$ , що відповідає обмеженню 1.11 з потужністю  $k = |\bar{A}|$ ;
- потужність множини інтервалів  $n$ , отриманої як результат задачі інтервалізації.

Результати:

Квадратна матриця раціональних чисел  $\bar{P}$  розмірністю  $n$ .

Обмеження:

$$\forall x_{ij} \in \bar{P} : x_{ij} \in [0.0; 1.0], \text{ де } i, j \in [0; n) \quad (1.12)$$

### Висновок

В цій статті було проаналізовано вирішення задачі автентифікації користувача за рухами курсора миші з використанням лінгвістичного підходу. Був проведений аналіз попередніх робіт та виявлено їхні недоліки. Були сформовані математична та змістова постановки задачі. Поетапно описано спосіб застосування методу лінгвістичного моделювання для розв'язання даної задачі.

### Література

1. Hashia S., Pollett C., Stamp M. On Using Mouse Movements As a Biometric // Proceeding in the International Conference on Computer Science and its Applications, Volume 1. – 2005. – 5 p.
2. Hashia S. Authentication by Mouse Movements // CS 297 Report. – 2004. – 20 p.

3. Weiss A., Ramapanicker A., Shah P., Noble S., Immohr L. Mouse Movements Biometric Identification: A Feasibility Study // Proceedings of Student/Faculty Research Day, CSIS, Pace University. – 2007. – 8 p.
4. Баклан І. В. Лінгвістичне моделювання: основи, методи, деякі прикладні аспекти. // Систем. технології. — 2011. — № 3. — С. 10-19.
5. Баклан І. В. Структурний підхід до розпізнавання образів у системах безпеки. Національна безпека України: стан, кризові явища та шляхи їх подолання. // Міжнародна науково-практична конференція (Київ, 7-8 грудня 2005 р.). Збірка наукових праць. К.: Національна академія управління — Центр перспективних соціальних досліджень, 2005. — С.375-380.
6. Баклан І. В. Аналіз поведінки економічних часових рядів з використанням структурних підходів. // Сборник МКММ-2006. — Херсон: ХГТУ, 2006

## ІНФОРМАЦІЙНА ПІДСИСТЕМА ПОШУКУ ШЛЯХУ ДЛЯ МОДЕЛЮВАННЯ ПОВЕДІНКИ ВЕЛИКОЇ КІЛЬКОСТІ ЛЮДЕЙ В ОБМЕЖЕНОМУ ПРОСТОРИ

В рамках задачі створення системи комп'ютерного моделювання поведінки натовпу у обмеженому просторі виникла задача удосконалення методів пошуку найкоротшого шляху. Однак мета цієї модифікації не вдосконалення існуючих методів по складності, а їх адаптація для роботи в динамічному середовищі з постійно змінюваним оточенням та положенням агента. А також наблизити поведінку агента до поведінки реальної людини.

КЛЮЧОВІ СЛОВА: ПАНІКА, ДИНАМІКА НАТОВПУ, ЕВАКУАЦІЯ, АЛГОРИТМИ ПОШУКУ ШЛЯХУ, АЛГОРИТМ А\*

In the framework of the task of creating a system for computer modeling of the behavior of the crowd in a confined space, the task of improving the methods of searching for the shortest path has emerged. However, the purpose of this modification is not to improve the existing methods of complexity, but their adaptation to work in a dynamic environment with a constantly changing environment and the position of the agent. And also bring the agent's behavior closer to the behavior of the real person.

KEYWORDS: PANIC, CROWD DYNAMICS, EVACUATION,

### 1. Вступ

У якості базового алгоритму було обрано А\*. Це популярний алгоритм пошуку найкоротшого шляху на площині, що фактично являє собою пошук в ширину по графу з евристикою.

Основна ціль створення системи - розробити швидкий динамічний алгоритм пошуку шляху (не обов'язково найкоротшого) для механічної моделі поведінки людей в обмеженому середовищі, який буде забезпечувати знаходження допустимого розв'язку, якщо він існує, та буде максимально точно імітувати поведінку реальних людей. Під поняттям динамічний мається на увазі такий, що буде адекватно реагувати (перебудувати або змінювати маршрут) в залежності від зміни навколишнього середовища або стану агента

### 2. Постановка задачі

Задача підсистеми полягає у пошуку шляху для агента від поточної позиції до виходу на двовимірній площині. Алгоритм повинен реагувати на зміни геометрії простору та місцеположення інших агентів.

В будь-яких момент часу він повинен видавати допустимий розв'язок, якщо він існує. Розв'язок не повинен бути оптимальним, але повинен адекватно імітувати поведінку людини. «Адекватність» алгоритму повинна регулюватися певними параметрами його роботи (евристична функція, ваги ребер графу та ін.).

### 3. Математична модель

На вхід алгоритму подається матриця дійсних чисел, що відображає геометрію простору. Чим більша вага комірки, тим важче туди дістатися.

Перешкоди моделюються комірками з нескінченною вагою.

Вільний простір моделюється одиничною вагою.

Простір з прохідними перешкодами позначається вагою :

$$w \in (1; \infty) \quad (1)$$

Також на вході ми маємо дві координати  $a(x_1; y_1)$  та  $b(x_2; y_2)$ , координату початку шляху та його кінця відповідно.

На виході ми повинні отримати масив координат комірок вхідної матриці, що представляє собою шлях. Перший елемент  $(x_1; y_1)$  масиву - це вершина, що слідує за початковою, а останній  $(x_n; y_n)$  - координата кінця шляху:

$$p = \{(x_1; y_1), \dots, (x_n; y_n)\} \quad (2)$$

#### 4. Метод розв'язання

В якості основного методу розв'язання задачі візьмемо популярний та широковідомий алгоритм  $A^*$ .

Алгоритм пошуку  $A^*$  (« $A$  зірочка» або англ. « $A$  star») — належить до евристичних алгоритмів пошуку. Використовується для пошуку найкоротшого шляху між двома вершинами графу з додатними вагами ребер.

Алгоритм використовує допоміжну функцію (евристику), аби скеровувати напрям пошуку та скорочувати його тривалість. У даній реалізації алгоритму використовується евклідова евристика – відстань між двома точками розраховується як довжина відрізка, що їх сполучає.

Алгоритм повний в тому сенсі, що завжди знаходить оптимальний розв'язок, якщо він існує.

Алгоритм  $A^*$  спершу відвідує ті вершини, які ймовірно ведуть до найкоротшого шляху до мети. Аби розпізнати такі вершини, кожній відомій вершині  $x$  співставляється значення  $f(x)$ , яке дорівнює довжині найкоротшого шляху від початкової вершини до кінцевої, який пролягає через обрану вершину. Вершини з найменшим значенням  $f$  обираються в першу чергу.

Функція  $f(x)$  для вершини  $x$  визначається так:

$$f(x) = g(x) + h(x), \quad (3)$$

де  $g(x)$  - функція, значення якої дорівнюють вартості шляху від початкової вершини до  $x$ ,  $h(x)$  - евристична функція, оцінює вартість шляху від вершини  $x$  до кінцевої.

Принцип дії [3].

Алгоритм ділить вершини на три класи:

- невідомі вершини: ці вершини ще не були знайдені. Ще не відомий шлях до них. На початку роботи алгоритму всі вершини, окрім початкової, належать до класу невідомих;
- відомі вершини (OpenList) вже відомий (можливо не оптимальний) шлях до цих вершин. Всі відомі вершини разом зі значеннями  $f$  зберігаються в списку. З цього списку вибираються, в першу чергу, найперспективніші вершини. Конкретна реалізація цього списку має істотний вплив на швидкість алгоритму, і зазвичай має вигляд черги з пріоритетом (наприклад, бінарна купа). На початку роботи алгоритму до відомих вершин належить лише початкова вершина;
- повністю досліджені вершини (ClosedList) до цих вершин вже відомий найкоротший шлях. Повністю досліджені вершини додаються до так званого замкненого списку, аби запобігти багаторазовому дослідженню вже досліджених вершин. Список повністю досліджених вершин на початку роботи алгоритму порожній.

Кожна відома або повністю досліджена вершина має вказівник на попередні вершини. Завдяки цьому вказівникові, можна пройти шляхом від цієї до початкової вершини.

Коли вершину  $x$  буде повністю досліджено (або розкрито, релаксовано), суміжні з нею вершини додаються до списку відомих вершин, а сама вершина додається в список повністю досліджених. Вказівники на попередню вершину встановлюються на  $x$ . Суміжні вершини, які вже знаходяться в списку повністю досліджених вершин, до списку відомих не додаються, а зворотні вказівники не змінюються. Суміжні вершини, які вже знаходяться в списку відомих, лише оновлюються (значення  $f$  та вказівник на попередню вершину), якщо знайдений до них шлях коротший за вже відомий.

Алгоритм зупиняється коли кінцева вершина потрапляє до списку повністю досліджених вершин. Знайдений шлях відтворюється із допомогою вказівників на

попередню вершину. Якщо список відомих вершин порожній, то розв'язку задачі не існує, і алгоритм припиняє пошук.

Відтворений за зворотними вказівниками знайдений шлях починається з кінцевої вершини та прямує до початкової. Аби одразу отримати шлях в правильному напрямі, з початкової вершини до кінцевої, в умовах задачі слід переставити місцями початок та кінець. Якщо шукати шлях починаючи з кінцевої вершини, відтворений список починатиметься з початкової вершини й прямуватиме до кінцевої.

### 5. Адаптація алгоритму

Як видно описаний алгоритм знаходить найкоротший шлях за певною евристикою в заданому просторі. Фактично являє собою пошук в ширину, який скеровується значенням евристики та відсікає неефективні зони.

Керувати роботою алгоритму ми можемо двома способами:

- змінювати ваги комірок вхідної матриці відповідно до поточної ситуації;

– змінювати евристику  $h(x)$  оцінки вартості шляху від поточної вершини до кінця.

Найбільш гнучкий спосіб – це змінювати ваги матриці. Визначимо наступні категорії комірок:

- комірки з нездоланими перешкодами – для них вага буде дорівнювати нескінченності  $w = \infty$ ;
- комірки без перешкод – для них вага буде дійсним числом яке представляє собою характеристику рельєфу,  $w = 1$  – плаский рельєф,  $w > 1$  – підйоми, сходи та інші важкопроходимі ділянки,  $w < 1$  – спуски, ескалатори вниз та ін.;
- комірки без перешкод, в яких є агенти – для них вага буде функцією від кількості агентів, чим менше агентів, тим легше пройти комірки з точки зору агента:

$$w_{xy} = w^r + Cn_{xy}, \quad (4)$$

де  $w_{xy}$  – вага комірки  $(x; y)$ ,  $w^r$  – вага рельєфу комірки,  $C$  – ваговий коефіцієнт функції залежності ваги від кількості агентів. Цей коефіцієнт налаштовується у відповідності до поточної ситуації.

### Висновки

В процесі роботи на системою було розроблено методи адаптації алгоритму А\* до механічної моделі симуляції поведінки натовпу. Адаптація мала на меті зробити поведінку агентів максимально подібною до поведінки реальних людей в умовах паніки. Реалізовано метод у вигляді програмного продукту. Проведені експерименти з використанням реальних даних та на їх основі відкалібровані параметри алгоритму.

### Список літератури

1. Белкарян А. Л., Акопов А. С. «Моделирование поведения толпы на основе интеллектуальной динамики взаимодействующих агентов» // Бизнес-информатика № 1 (31), 2015 С. 69-77
2. А. М. Аптуков, Д. А. Брацуна, А. В. Люшнин «Моделирование поведения паникующей толпы в многоуровневом разветвленном помещении» // КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ И МОДЕЛИРОВАНИЕ 2013, Т. 5, № 3 С. 491–508
3. Red Blob Games - Implementation of A\* [Електронний ресурс] // <http://redblobgames.com/pathfinding/a-star/implementation.html>

УДК 51-74

МЕХЕДОВ Є. В.

## Принципи побудови мап рельєфу для наземної навігації

Дана стаття присвячується огляду основних принципів побудови мап рельєфу для наземної навігації, зокрема за допомогою використання даних SRTM. Актуальність дослідження полягає в повсякденному використанні цифрових моделей рельєфу в різних сферах життєдіяльності людей. В нинішній час к цифровим моделям рельєфу пред'являються жорсткі вимоги щодо точності. Багато досліджень направлено саме на методах підвищення точності отриманих цифрових моделей рельєфу.

**КЛЮЧОВІ СЛОВА:** навігація, рельєф, дані, SRTM, цифрові моделі, точність, ландшафт, залежність, геоморфологія, лінії.

This article provides an overview of the basic principles of terrain mapping for ground navigation, in particular through the use of SRTM data. The relevance of the study is the daily use of digital terrain models in various spheres of human life. At present, digital elevation models are subject to strict accuracy requirements. A lot of research is aimed at methods to improve the accuracy of digital elevation models.

**KEYWORDS:** navigation, relief, SRTM data, digital models, accuracy, landscape, dependence, geomorphology, lines.

### 1. Вступ

Рельєф є одним з головних факторів, що обумовлює розвиток різних природних процесів на поверхні Землі. Так, він визначає особливості формування поверхневого стоку і схилових процесів, істотно перерозподіляє сонячну радіацію, що приходить на землю, при цьому формуючи ландшафтну диференціацію навіть на локальних територіях.

Отримання інформації про рельєф традиційними методами завжди досить трудомістке, часто передбачається побудова ряду морфометричних карт з топографічних основ того чи іншого масштабу.

Тому, в даний час широко використовують електронне представлення рельєфу, яке в контексті географічних інформаційних систем (ГІС) має векторне або растрове уявлення [1].

Одним із методів, за допомогою якого можна побудувати цифрові моделі рельєфу, є використання методів SRTM.

SRTM (Shuttle radar topographic mission) - радарна інтерферометрична зйомка поверхні Землі, здійснена в лютому 2000 р радіолокаційними сенсорами SIR - C і X-SAR з борту космічного корабля "Шаттл". Дані SRTM поширюються у вигляді сіток з розміром комірки 1 кутова секунда і 3 кутові секунди.

Більш точні дані (SRTM1) доступні на території США, на іншу поверхню Землі доступні тільки трисекундні дані (SRTM3). Файл (SRTM3) являє собою матрицю з 1201x1201 значень, яка може бути імпортована в різні програми побудови карт і ГІС.

Матриця SRTM являє собою цифрову модель рельєфу (ЦМР), яка далі засобами ГІС може піддаватися просторовому аналізу і інтерпретації. Безсумнівна перевага використання програмних засобів ГІС- це автоматизація контурного дешифрування об'єктів дослідження, яка не тільки зменшує трудовитрати, але також практично виключає суб'єктивний (людський) фактор [2]. Саме це й показує актуальність даного дослідження.

### 2. Літературний огляд

Рівень пізнання морфологічних властивостей земної поверхні і картографічна вивченість рельєфу, яка виступає в якості об'єкта дослідження, явно недостатні. Морфологічна структура рельєфу, опис сполучень його геометричних форм і елементів до сих пір залишалися поза областю регіональних геоморфологічних досліджень [3].

Незважаючи на те, що дослідження просторово-часової структури геосистем є традиційними в географічній науці ( цим

займалися такі вчені, як Н.А. Сонцев, В.С. Преображенський, А.Г. Ісаченко, І.І. Мамай, А.Ю. Ретеюм і багато ін.) , питання ієрархії просторово-часової організації географічних систем і суворо алгоритмізація її моделювання до теперішнього часу відносяться до актуальних завдань географічної науки.

При проведенні подібних досліджень доцільне подання рельєфу в ГІС як цифрової моделі рельєфу (ЦМР) у вузлах регулярної сітки значень абсолютних висот. Такий підхід при адекватній побудові моделі дозволяє провести морфометричний аналіз, автоматизований кількісний аналіз рельєфу засобами ГІС технологій.

До найбільш відомих і фундаментальних, з аналізу рельєфу, як поля висот, можна віднести роботи таких фахівців, як: П. К. Соболевський; А. Н. Ласточкін; Червяков; П. А. Шарий; J.Krcho; I. S. Evans; J. D. Wood [4].

Точність матриці SRTM вивчалася вченими різних країн. Так А. К. Корвеул, І. Евіак оцінюють помилку матриці SRTM наступними величинами: для рівнинної місцевості - 2,9 м, горбистій – 5,4 м. На їхню думку, матриця SRTM підходить для створення горизонталей на картах масштабу 1: 50 000 і дрібніше.

Приблизно схожі результати призводять і інші зарубіжні дослідники. Проводилося тестування матриці SRTM і в РФ Л. А. Муравйовим на підставі вивчення 3-х територіально роз'єднаних тестових ділянок: "Катока" (Ангола), Фгеоконсалтинг (М. Тюмень), МП Електра (М. Южносахалінськ). Муравйов приходить до висновку, що дані SRTM можуть бути використані для оновлення топооснови території, де відсутні детальні топографо-геодезичні матеріали.

Ю. І. Каріонов також використовував територіально роз'єднані ділянки: острів Ольхон на Байкалі (гірський рельєф), район міста Саратов (рівнинний рельєф), район міста Сочі (високогірний рельєф) [5].

З матеріалів порівняння слідував висновок про відповідну точність матриці SRTM і матриці карти масштабу 1:100 000.

На думку автора, матриця SRTM може бути використана при створенні ортофотопланів у масштабі 1:25 000 і дрібніше на райони з рівнинним рельєфом. У районах з гірським рельєфом для попереднього складання ортофотопланів необхідна додаткова корекція космічних знімків, що враховує умови зйомки.

У високогірних районах для виготовлення ортофотопланів масштабу 1:25 000 розмір комірки занадто великий.

### 3. Мета та задачі дослідження

Метою даного дослідження є огляд основних принципів побудови мап рельєфу для наземної навігації, зокрема за допомогою використання даних SRTM.

При виконанні даного дослідження були поставлені наступні задачі:

1. Виконати літературний огляд за темою дослідження;
2. Визначити методи побудови мап рельєфу для наземної навігації;
3. Розглянути принципи використання даних SRTM для побудови карт наземної навігації.

### 4. Принципи побудови мап рельєфу для наземної навігації

У визначенні рельєфу як основного об'єкта геоморфології існують дві системи поглядів. В одному із загальноприйнятих трактувань рельєф утворює певну сукупність форм земної поверхні, яку можна описати у вигляді набору просторово координованих висотних відміток. Такий підхід до дослідження форм земної (фізичної) поверхні називають геометричним. В альтернативному трактуванні рельєфу враховуються і його геологічні тіла.

У геоморфології «геометричний підхід» в дослідженні земної поверхні, при якому саме ця поверхня є самодостатнім об'єктом аналізу, давно отримав самостійний розвиток, для чого існували передумови, закладені ще Гауссом у вигляді диференціальної геометрії поверхонь.

Побудова моделі рельєфу, що забезпечує виділення ієрархічних одиниць його поверхні, достовірно різниться за природними екологічними властивостями і



потенційним геофізичними характеристиками, дає можливість порівняння потенційних і реальних (в термодинамічних умовах конкретних морфологічних систем) геофізичних полів. Така дискретизація дозволяє будувати векторні поля градієнтів геофізичних полів геосистем, виявляти параметри порядку самоорганізації геосистем і досліджувати закономірності складання їх вертикальної і горизонтальної структури [6].

В нинішній час ЦМР можна отримати з двох видів джерел: до перших відносяться дані радарної зйомки поверхні землі; до других видів джерел відносяться горизонталі топографічних карт різного масштабу, які можна використовувати як вихідні дані в різних методиках апроксимацій для отримання ЦМР [1].

Логічна структура даних цифрової інтегрованої моделі географічних систем відображає геодинамічну єдність внутрішніх і зовнішніх сфер Землі. Ієрархічна структура даних всього комплексу компонентів (кліматичної системи, ґрунтів, гідрографічної мережі, рослинності) будується відповідно до класифікації і формує структуру об'єктів факторів. Така система даних масштабована і конструктивна, оскільки орієнтується на розуміння генетичної природи ієрархії. При цьому об'єкти клімату на регіональному рівні додатково ідентифіковані по однотипним величині і напрямку градієнтів основних факторів: температури і опадів - кліматичне поле регіону, також як і рельєф, характеризується на основі відображення просторової енергетики процесів його формування.

Кожен об'єкт характеризується параметрами структурно-формуєчих (інваріантних) факторів, що мають достовірні відмінності з аналогічними параметрами інших об'єктів того ж рангу. Такий підхід забезпечує функціонально узгоджене структурування простору за його вертикального і горизонтального вимірах і надає фактичні дані для верифікації моделей будови і розвитку геосистем, побудови спеціалізованих моделей.

В області класифікації ієрархічних рівнів геосистем існує невирішена проблема суворої алгоритмізації системної упорядкованості вертикальної і горизонтальної структури.

Досвід визначення точними кількісними методами полі - і моно системної організації ландшафтів на основі інваріантних параметрів компонентів показав, що в умовах стійкості ці моделі не працюють [6].

В загальному вигляді термін «цифрове моделювання» в геоінформаціі використовується при використанні можливостей математичних методів і програмних засобів для моделювання об'єктів земної поверхні [7].

Основним елементом моделювання є цифрова модель місцевості, яка отримується за допомогою різноманітних технологій.

Цифрові моделі оперують з різними типами інформації.

Метрична інформація визначає вимірювальні характеристики об'єктів, тобто координати, розміри.

Ця інформація відносно проста і однорідна по структурі, в силу чого вона стає сильно типізованою.

Атрибутивна інформація включає в себе атрибутивні дані та метадані. Вона може поділятися на семантичну, технологічну та інші види.

Даний тип інформації визначає приналежність об'єктів до певного класу, описує властивості об'єктів та їх частин, задає взаємозв'язки і умови обробки, умови відтворення тощо.

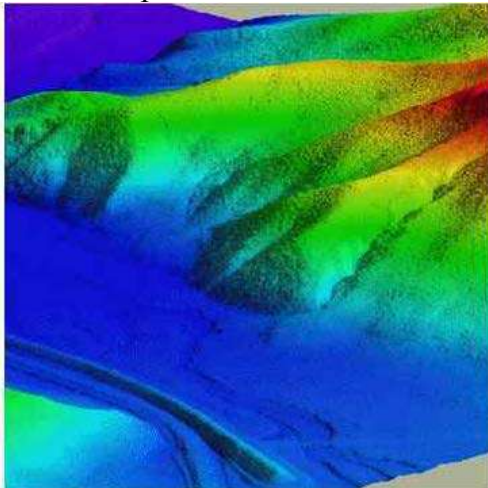
Синтаксична інформація визначає послідовність роботи при коректуванні чи оновленні цифрових моделей, правила їх побудови і представлення.

Технологічно можна виділити наступні види моделювання:

1. Семантичне;
2. Інваріантне;
3. Геометричне;
4. Евристичне;
5. Інформаційне.

Вони проявляються в різному ступені на різних системних рівнях обробки інформації.

Приклад цифрової моделі рельєфу представлений рис. 1.



**Рис. 1. - Приклад цифрової моделі рельєфу**

Цифрове моделювання рельєфу, результатом якого є створення цифрових моделей рельєфу (ЦМР), - одна з потужних і широко поширених функцій ГІС.

Під ЦМР (digital elevation model, digital terrain model, DEM, DTM) прийнято розуміти засіб представлення тривимірних просторових об'єктів (поверхонь або рельєфів) у вигляді тривимірних даних, що утворюють безліч висотних відміток і інших значень аплікату в вузлах регулярної або нерегулярної мережі або сукупність записів горизонталей чи інших ізоліній.

Горизонталі можна уявити як набір точок з абсолютними висотами і використовувати методи моделювання по точках: метод Шепарда; крігінг; радіальні функції; середньозважена інтерполяція і т. д. [8].

Однак при такому підході втрачається велика кількість інформації, закладеної в структурі горизонталей як геометричних об'єктів. Таким чином, для подібних даних повинні розроблятися власні методи, що враховують їх особливості.

Для створення ЦМР по горизонталях можна виділити три найбільш часто вживаних методи моделювання: на основі триангуляції, аналітичних сплайнів, кусково-поліноміального згладжування.

Всі три методи мають як переваги, так і недоліки. При використанні триангуляції виходять спотворення на краях, звідси слідує, що треба використовувати

інформацію, яка представляє внутрішні трикутники, однак це рідко вдається, часто територіального запасу інформації немає.

Сплайн-інтерполяція мінімізує кривизну поверхні. Вважається, що для цієї мети зручні і практичні бікубічні функції.

Кусково-поліноміальне згладжування здійснюється, як видно з назви, поліномами різного ступеня. Їх «кусоквість» полягає в тому, що на кожен конкретну ділянку апроксимації підбираються нові коефіцієнти по новому набору даних.

Різні шматки поверхні побудовані за допомогою різних поліноміальних функцій. Поверхня, побудована таким чином, виходить найбільш гладкою [9].

Альтернативним джерелом даних відносно висот є вільно поширювані дані SRTM (Shuttle radar topographic mission), доступні на більшій частині території земної кулі з роздільною здатністю моделі 90 м.

У 2011 році за даними зйомки сенсора X-SAR Німецьке аерокосмічне агентство (DLR) створило ЦМР SRTM X-band з розміром комірки 30x30 м. Дана ЦМР містить значні прогалини між смугами зйомки: чим південніше, тим більше пропусків.

Альтернативним джерелом даних відносно висот є вільно поширювані дані SRTM (Shuttle radar topographic mission), доступні на більшій частині території земної кулі з роздільною здатністю моделі 90 м.

У 2011 році за даними зйомки сенсора X-SAR Німецьке аерокосмічне агентство (DLR) створило ЦМР SRTM X-band з розміром комірки 30x30 м. Дана ЦМР містить значні прогалини між смугами зйомки: чим південніше, тим більше пропусків.

Як вже вказувалося, дані SRTM існують в декількох версіях: попередні (версія 1, 2003 р.) і остаточна (версія 2, лютий 2005 р.). Остаточна версія пройшла додаткову обробку, виділення берегових ліній і водних об'єктів, фільтрацію помилкових значень. Дані поширюються в декількох варіантах - сітка з розміром вічка 1 кутова

секунда (~30 метрів) і 3 кутові секунди (~90 метрів).

Досить часто виготовляються планово-картографічні матеріали, зазвичай починаючи з масштабу 1:25 000 (планшети). У масштабі 1:50 000 і дрібніше виготовляються плани і укрупнені плани різного роду ландшафтів.

Зіставляючи наведені дані про точність матриці SRTM і рекомендації щодо їх використання, слідє висновок про її застосованість для вирішення значної кількості науково-практичних завдань.

### 5. Результати досліджень

Апарат досліджень сучасної географії поповнився активним поєднанням геоінформатики та аерокосмічного зондування. Основним методом геоінформатики вважається цифрове моделювання та тісно пов'язане з ним геоінформаційне картографування.

Методично дослідження просторових об'єктів (систем) в геоінформатиці здійснюються шляхом створення цифрових інформаційних моделей місцевості, окремим випадком яких є ЦМР.

Карти як джерело вимірювань кількісних показників різних просторових об'єктів в даний час представлені

переважно цифровими картографічними продуктами.

### 6. Висновки

Таким чином, в наш час повсюди використовуються цифрові моделі рельєфу, які постійно намагаються вдосконалити. Серед численних сфер використання цифрової моделі рельєфу можна виділити наступні:

- вивчення та кількісна оцінка сучасного стану природного середовища;
- територіальне планування (міське, ландшафтне та ін.);
- моделювання екологічних ситуацій;
- прогнозування ландшафтних процесів та ін.

Даний перелік науково-практичних завдань може бути продовжений, що дозволяє оцінити сферу застосування матриці SRTM при побудові цифрових мап рельєфу для наземної навігації.

Зрозуміло, що всі такого роду завдання вирішуються на основі спеціальних матеріалів і відомостей, але залучення програмних засобів аналізу рельєфу місцевості внесе важливі додаткові переваги, серед яких особливо слід виділити об'єктивність контурного дешифрування об'єктів вивчення, часткову автоматизацію картографічних робіт і, що є дуже важливим, точність.

### Перелік посилань

1. Мальцев К.А. Построение цифровых моделей рельефа при помощи кубических парабол. Геоморфология. 2006;(3):30-36.
2. The shuttle radar topography mission. / Farr Tom G., Hensley Scott, Rodriguez Ernesto, Martin Jan, Kobrick Mike. // CEOS SAR Workshop. Toulouse 26-29 Oct. 1999. Noordwijk. 2000. С. 361-363.
3. Гук П.Д. Технология создания карт фототопографическими методами. Учебное пособие. Новосибирск, 1990г.
4. Shary P.A. Land surface in gravity points classification by a complete system of curvatures. Mathematical Geology 27. 1995. pp. 373-390.
5. Карионов Ю. И. Оценка точности матрицы SRTM. [Электронный ресурс]. – Режим доступа: [www.racurs.ru/?page=506](http://www.racurs.ru/?page=506).
6. Wood J.D. Scale-based characterisation of Digital Elevation Models, Proceedings, 3rd National Conference on GIS Research UK (GISRUK '95), Newcastle. 1995.
7. Бугаевский Л.М., Цветков В.Я. Геоинформационные системы. Учебное пособие для вузов. М.: 2000.
8. A. K. Karwel, I. Ewiak, Estimation of the accuracy of the SRTM terrain model on the area of Poland, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII. Part B7. Beijing 2008. – pp. 169-172
9. Sun W., Xu G., Gong P., Liang S. Fractal analysis of remotely sensed images: A review of methods and applications. International Journal of Remote Sensing, Vol.27, No.22, 2006.

УДК 683.519

*НЕВМЕРЖИЦЬКИЙ Д. О.,  
ОЛІЙНИК Ю.О.*

## **СИСТЕМА ОБЛІКУ РЕГУЛЬОВАНИХ ЛІКАРСЬКИХ ЗАСОБІВ ІЗ ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ ЛАНЦЮЖКІВ БЛОКІВ ТРАНЗАКЦІЙ**

В даній статті розглянуто декілька концепцій системи обліку регульованих лікарських засобів із використанням технології ланцюжків блоків транзакцій та запропоноване покращення існуючих систем за рахунок використання технології ланцюжків блоків транзакцій. Також проведений порівняльний аналіз централізованих та децентралізованих систем обліку.

**КЛЮЧОВІ СЛОВА:** Блокчейн, Ланцюжки Блоків Транзакцій, Децентралізовані Системи, Розумні Контракти, Ефіріум

In this article, several concepts of the accounting system of regulated medicinal products using the technology of chain of transaction blocks are considered, and the proposed improvement of existing systems through the use of technology of chain of transaction blocks. A comparative analysis of centralized and decentralized accounting systems was also conducted

**KEYWORDS:** Blockchain, Transaction Blocks Chains, Decentralization, Smart Contracts, Ethereum

### **Вступ**

На сьогоднішній день більшість системи обліку регульованих лікарських засобів в Україні мають мережеву архітектуру для моніторингу відразу декількох складів (автономні рішення систем складського обліку зустрічаються все рідше і рідше). Мережа значно прискорює рух даних усередині системи, дозволяє відстежувати рух ліків відразу за кількома складах в реальному часі. Загалом такі системи являють собою програмне забезпечення «клієнт-серверної» архітектури, отже, адміністрація має повний контроль та сама система ніяк не захищена від незаконної діяльності адміністраторів. Завдяки чому можливі маніпуляції із даними в базі даних, несанкціонований доступ до серверів, формування «необхідного» бухгалтерського обліку, що можуть призвести до корупції, безкарної крадіжки різноманітних медичних препаратів, «відмивання грошей» та іншої незаконної діяльності [1].

Основною задачею, яка вирішується є покращення системи обліку регульованих лікарських засобів за рахунок значної надійності збереження та незмінності даних про операції із ліками. Завдяки чому система

стане більш стійкою до незаконної діяльності зі сторони адміністрації.

### **1. ТЕХНОЛОГІЇ ЛАНЦЮЖКІВ БЛОКІВ ТРАНЗАКЦІЙ**

У загальному випадку, технологія ланцюжків блоків транзакція (блокчейн) - це база даних, в яку записуються факти і повні копії якої містяться на всіх комп'ютерах, об'єднаних в peer-to-peer (P2P). База даних хронологічно лінійно розширюється. Факти, які заносяться в базу за допомогою транзакції, можуть бути різними, наприклад, грошові транзакції, підписи вмісту або зміна стану. Члени мережі - анонімні суб'єкти, звані вузлами мережі. Новий вузол, приєднується до мережі, зобов'язаний завантажити повну копію блокчейну. Всі комунікації всередині мережі використовують інструменти криптографії для точної і безпечної ідентифікації відправника і одержувача. В разі, якщо вузол хоче додати факт в базу, в мережі повинен бути сформований консенсус, який визначає, де буде розміщений факт. Даний консенсус називається блоком [2].

Самі по собі P2P-мережі аж ніяк не новинка, вони існують вже довгий час. P2P-мережі - це розподілені системи, яким необхідно вирішити одну дуже складну проблему інформатики - вирішення

конфліктів або узгодження. На той випадок, коли два несумісних факту надходять одночасно, система повинна володіти правилами, що визначають, який з цих фактів вважати дійсним. У той час як в централізованих системах (за рахунок застосування централізованого консенсусу (тобто є тільки одна база даних, яка визначає дійсність транзакції)) і реляційних базах даних (за рахунок посилальної цілісності) ця проблема вирішена, механізми, застосовувані в таких системах, не можуть бути застосовані в P2P-мережах. Щоб гарантувати цілісність всередині P2P-мережі, необхідно ввести систему консенсусу - спосіб, щоб всі вузли мережі погодилися з порядком фактів. Деякі блокчейн платформи пропонують для цього алгоритм "докази роботи" (Proof-of-work consensus), який використовує блоки [2] (також існують інші алгоритми [3]). Цей децентралізований консенсус руйнує стару парадигму централізованого консенсусу.

Блоки - це спритний спосіб організації фактів в мережі недовірених користувачів. В їх основі лежить проста ідея: згрупувати факти в блоки, скласти з блоків єдину ланцюжок, який буде реплікований по всіх вузлах мережі. Цей ланцюжок блоків видно для всіх вузлів в мережі. Кожен блок посилається на попередній, так може бути простежено походження кожного факту. Криптографія (в особі хеш-кодів) використовується для забезпечення безпеки аутентифікації джерела транзакції і усуває необхідність центрального посередника. Комбінація криптографії та технології блокчейн гарантує, що жодна з транзакцій не буде записана двічі. Факти, ще не додані до блоку, називаються ті що очікують [2].

Отже, ми маємо надійну децентралізовану базу даних у якій інформація зберігається в блоках. Та із використанням даної технології система обліку лікарських засобів стане більш надійною щодо збереження та незмінності даних про операції із ліками.

## 2. Розумні контракти

Розумний контракт - різновид угоди в формі закодованих математичних алгоритмів, укладення, зміна, виконання і розірвання яких можливо лише з

використанням комп'ютерних програм (Блокчейн платформ) в рамках мережі Інтернет. Тобто таке програмне забезпечення яке виконується на Блокчейн-платформі та стан, якого зберігається у блоках [4].

Проаналізувавши сучасні Блокчейн-платформи із підтримкою розумних контрактів [5, 6, 7, 8, 9] найоптимальнішим рішенням, щодо надійності, безпеки, пропускну здатності мережі та «потужності» розумних контрактів буде платформа Ethereum (Ефіріум) із її віртуальною машиною та мовою програмування розумних контрактів Solidity [5].

## 3. Вирішення проблем програмного забезпечення на базі розумних контрактів

Згідно з МОЗ України від 07.08.2015 р. № 494 «приміщення, сейфи та металеві шафи в закладах охорони здоров'я, де зберігаються наркотичні засоби, психотропні речовини та прекурсори, після операцій з ними повинні замикатися, а після закінчення робочого дня — опечатуватися чи опломбовуватися і здаватися під охорону.» [10] - отже, можна припустити, що складські операції з регульованими лікарськими засобами проводяться рідко, тому для системи обліку немає потреби у значній пропускну здатності для бази даних. Мережа; Ethereum підтримує 7-8 транзакцій за секунду [5], цього показника вистачить для безперебійної роботи системи.

Одна з найголовніших властивостей системи обліку – агрегація інформації. Проте у блокчейн базі, як і в будь-яких «ключ—значення» базах даних, агрегація є складним процесом оскільки підтримка зв'язків «багато-до-одного» і «багато-до-багатьох» є дуже складним процесом [11]. Вирішенням цієї проблеми є використання журналів. Для запису в журнал у мові програмування Solidity використовується конструкція подій (events). Події є успадкованими учасниками контрактів. Коли вони викликаються, вони викликають збереження аргументів у журналі транзакції - спеціальної структури даних блокчейну. Ці журнали пов'язані з адресою контракту та будуть включені в блокчейну базу даних і залишатимуться там, доки будет доступний

блок (до припинення роботи мережі) - назавжди. Дані журналу та події недоступні з-поміж інших договорів (навіть з контракту, який їх створив). До журналу можна записувати інформацію будь-якого розміру – на скільки вистачить газу [5]. Також дозволяється записати до трьох індексованих полів для зручного пошуку інформації. І одна з найголовніших властивостей журналу це незмінність інформації – те що записали залишається незмінним назавжди [5].

Блокчейн база даних є публічною, кожен може її переглянути. Проте для виконання певного роду записів до бази, необхідно мати обліковий запис та невелику кількість ефіру (обмінна одиниця мережі [5]). Оскільки, обліковий запис може мати будь-хто та будь-яку кількість, то для реалізації розмежування доступу в мові програмування Solidity є патерн «Restricting Access», який використовує конструкцію модифікаторів (modifiers). За допомогою даного патерну можливо обмежити доступ до будь-яких операцій із системою для всіх крім спеціально вказаних. Також розширивши його можливо створити різні рівні доступу, наділяти / відбирати права.

Майже будь-яка автоматизована інформаційна система потребує внесення змін до її функціонування, наприклад, у зв'язку зі зміною бізнес процесів або виправлення помилок інформаційної безпеки. Система складського обліку лікарських засобів – не виняток. Проте байт код розумних контрактів в мережі Ethereum являється незмінним [5], тобто будь-яке оновлення коду рівноцінне створенню нового контракту з новою публічною адресою і новою «базою даних». Наприклад, після оновлення коду контракту всі його клієнти мають дізнатися нову адресу та працювати тільки з нею, а це дуже незручно. Для вирішення проблема оновлення в мові Solidity слід використати патерн архітектури розумних контрактів «Обновлюваний контракт (Upgradable Contract)». Основна ідея якого це розділення архітектури на 3

компоненти: проксі контракт, контракт для збереження даних та контракт із функціональністю. Адреса проксі контракту являється вхідною точкою для клієнтів, тобто всім клієнтам необхідно знати адресу тільки цього контракту для успішної взаємодії з ним. Сам проксі контракт наслідує контракт для збереження даних та зберігає посилання (публічну адресу) контракту із функціональністю, який також наслідую контракт для збереження даних. Таким чином при зміні коду контракту, все що потрібно це оновити застаріле посилання в проксі контракті. А клієнти взагалі нічого не помітять.

#### 4. Пропонований підхід покращення

Отже, з вирішеними проблемами програмного забезпечення на базі розумних контрактів. Можна розглянути концепції системи обліку регульованих лікарських засобів. Пропонується дві концепції: розумний контракт, як самостійний децентралізований програмний додаток складського обліку або як доповнення до звичайного централізованого програмного забезпечення. Розглянемо варіант самостійного додатку. У такому випадку вся бізнес-логіка проходить безпосередню в розумному контракті, тобто всі актори звертаються напряму до контракту в блокчейні через клієнтський додаток (рис. 1 - а). Переваги підходу із самостійним додатком:

- простота системи;
- легкий технічний супровід;
- прозора історія всіх операцій на складі;
- можливість працювати із системою з будь-якого місця.

Недоліки підходу із самостійним додатком:

- необхідність плати за кожну операцію;
- ускладнення клієнтських додатків;
- збільшення часу відгуку від додатку для клієнтів.

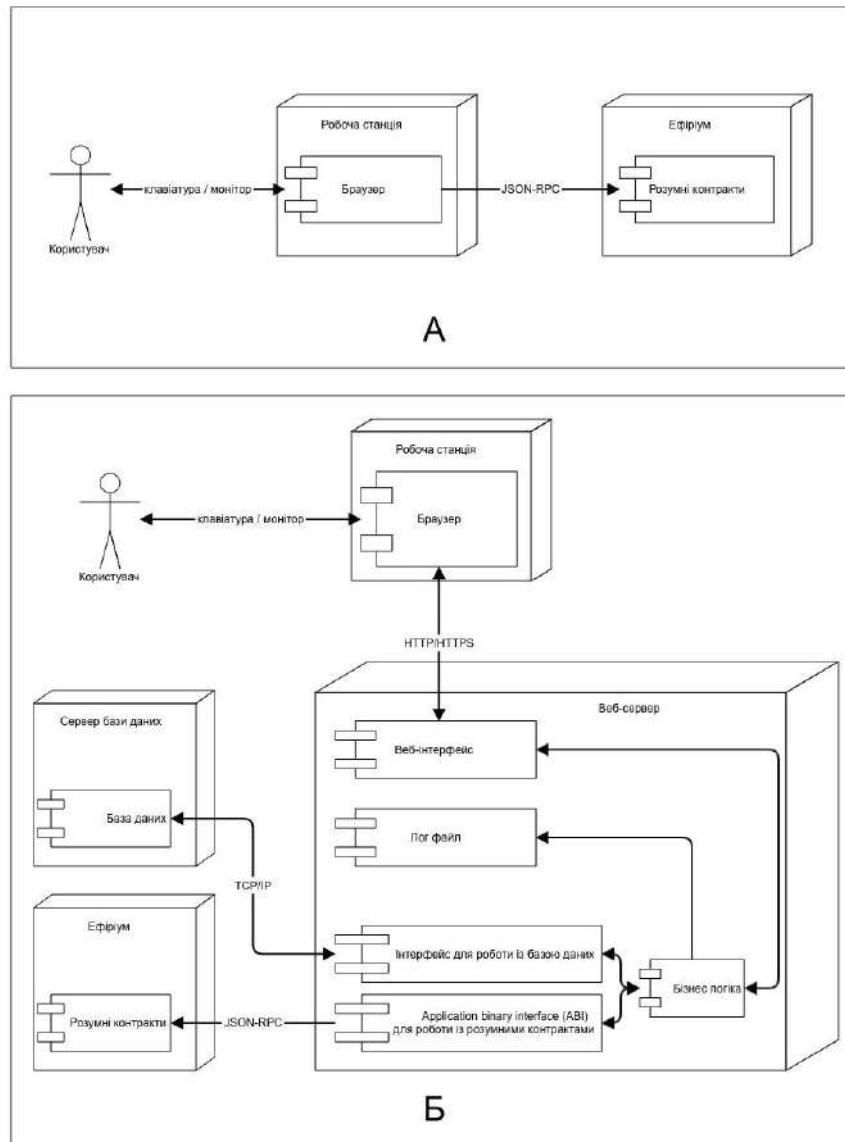


Рисунок 1. Діаграма розгортання системи регульованих лікарських засобів із використанням технології ланцюжків блоків транзакцій

Розглянемо підхід з доповненням до звичайного централізованого програмного забезпечення (рис. 1 - б). У цьому випадку вся бізнес-логіка проходить як і на центральному сервері так і у розумному контракті в блокчейні (гібридна архітектура). Проте до блокчейну записується лише основна інформація зі складського обліку – накладні на прибуток і вимогу - виникає навмисне дублювання даних для більшої надійності системи в цілому. В такому разі зали клієнтський додаток працює лише із центральним сервером та сам сервер взаємодіє з контрактом на блокчейні. Переваги підходу з гібридною архітектурою:

- малий час відгуку від додатку для клієнтів;

- можливість написання бізнес-логіки будь-якої складності;
- простота клієнтського додатку;
- прозора історія основних операцій на складі.

Недоліки підходу з гібридною архітектурою:

- невелике ускладнення системи;

Таким чином, найкращим варіантом, з архітектурної точки зору, для створення системи обліку регульованих лікарських засобів із значною мірою надійності збереження та незмінності даних про операції із ліками являється система із центральним сервером та додатковим розумним контрактом у блокчейні.

### Висновки.

У ході дослідження був розглянутий сучасний стан систем обліку регульованих лікарських засобів. Також проаналізовано сучасні блокчейн-платформи із підтримкою розумних контрактів. Найкращі результати показала платформа Ethereum із її «потужними» розумними контрактами на мові програмування Solidity. Також були вирішені основні проблеми програмного забезпечення на базу розумних контрактів. Запропоновано підходи для покращення надійності збереження та незмінності даних про операції із ліками у системах обліку регульованих лікарських засобів шляхом використання технології ланцюжків блоків транзакцій та розумних контрактів.

### Список літератури

1. Прокуратура міста Києва [Електронний ресурс] – Режим доступу до ресурсу: <https://kyiv.gp.gov.ua/ua/index.html>
2. Bitcoin: A Peer-to-Peer Electronic Cash System [Електронний ресурс] – Режим доступу до ресурсу: <https://bitcoin.org/bitcoin.pdf>
3. Coulouris, George; Jean Dollimore; Tim Kindberg (2001), Distributed Systems: Concepts and Design (3rd Edition), Addison-Wesley, p. 452, ISBN 0201-61918-0
4. Tapscott, Don; Tapscott, Alex (May 2016). The Blockchain Revolution: How the Technology Behind Bitcoin is Changing Money, Business, and the World. с. 72, 83, 101, 127. ISBN 978-0670069972.
5. Ethereum white paper [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/ethereum/wiki/wiki/White-Paper>
6. Tron white paper [Електронний ресурс] – Режим доступу до ресурсу: [https://o836fhe91.qnssl.com/tron/whitebook/TronWhitepaper\\_en.pdf](https://o836fhe91.qnssl.com/tron/whitebook/TronWhitepaper_en.pdf)
7. EOS.IO white paper [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md>
8. Waves white paper [Електронний ресурс] – Режим доступу до ресурсу: <https://blog.wavesplatform.com/waves-whitepaper-164dd6ca6a23>
9. NEM white paper [Електронний ресурс] – Режим доступу до ресурсу: [https://nem.io/wp-content/themes/nem/files/NEM\\_techRef.pdf](https://nem.io/wp-content/themes/nem/files/NEM_techRef.pdf)
10. МОЗ України від 07.08.2015 р. № 494 [Електронний ресурс] – Режим доступу до ресурсу: <http://zakon.rada.gov.ua/laws/show/z1028-15>
11. Martin Kleppmann. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems



УДК 004.93(015.7)

ОГОРОДНИКОВ Д.,  
СТЕЦЕНКО І.В.

## БЕЗПЕКА ІНТЕРНЕТ РЕЧЕЙ

В статті описані різні типи вразливостей пристроїв інтернету речей від апаратного втручання до технологій безпроводної передачі інформації, які можуть бути використані для отримання приватної інформації чи створення мереж уражених пристроїв. Підведений висновок стосовно ринку інтернет речей і стану питання захисту інформації й подальшого розвитку.

The article describes various types of vulnerabilities in Internet devices from hardware interference to wireless communication technologies that can be used to obtain private information or to create networks of affected devices. Bringing a conclusion on the Internet of things and the issue of information security and further development.

**Ключові слова.** Інтернет речей, кібератака, інформаційна безпека, стандарти захисту, захист пристроїв.

**Key words.** Internet things, cyberattack, information security, security standards, device protection.

### Вступ.

У 2018 році важко уявити життя без використання якоїсь інформаційної технології. Картки проїзду у міському транспорті, сенсорні панелі по відбиткам пальців вже не викликають такого захоплення як декілька років назад. Камери спостереження, датчики руху, біочіпи, розумній побутові прилади, розумна колонка, розумні годинники, всі ці речі спрощують наші повсякденні справи і роблять сучасне життя більш зручнішим. Багато хто не усвідомлюючи принцип роботи таких приладів, має змогу безперешкодно користуватись ними, але якщо ваш відбиток пальця не зможе розпізнати система при вході на роботу, або система сигналізації не спрацює коли це було потрібно, чи навпаки, спрацює помилково тоді вже у людей виникає паніка.

Інтернет речі, попри свою розповсюдженість, простоту і в тому часі складність реалізації, мають велику проблему зв'язану з інформаційною безпекою. Інколи, у злагодженій системі може виникати проблема, котра заважає приладу виконувати звичайну роботу. Це може бути як звичайна поломка приладу, зумовлена старістю деталей чи неправильністю експлуатації, але нас цікавить

такий чинник як зовнішнє втручання для зміни принципу роботи приладу.

**Предмет дослідження.** Дослідження можливих вразливих місць пристроїв інтернет речей. Дослідження можливих стандартів захисту пристроїв Інтернет речей. Розвиток систем і принципів захисту пристроїв інтернет речей у майбутньому.

**Аналіз публікацій.** Аналіз проблем безпеки інтернет речей – оцінка небезпек на структурних рівнях пристроїв інтернет речей[1].

Методи зовнішнього втручання у технології передачі даних інтернет речей. Огородников Д.В, описані найрозповсюдженіші методи втручання у середу передачі даних між пристроями інтернет речей.[2]

Розділення прав і обов'язків у мережі інтернету речей з використанням блокчейн технології описано у докладі Блокчейн як ДНК-протокол для Інтернет речей.

**Основні результати дослідження.** Дослідження систем і методів захисту пристроїв інтернет речей можна поділити на два типи. Теоретичну частину, описуючи можливі технології втручання і порушення роботи пристроїв інтернет речей, і практичну - це дослідження існуючих вразливих місць і тих випадків які вже слугують методами

доступу до контролю і інформації пристроїв. Також, дослідження можна робити опираючись на потреби в цей час, коли пристрої вже стали цілями для враження, або в макро-масштабі, прогнозуючи майбутні наслідки і можливі атаки у майбутньому, розробка стандартів захисту інтернет речей щоб бути до них готовими.

На теперішньому ринку переважають системи де різноманітні ролі віддаються пристроям інтернет речей, але є можливість відмовитися від них, і, наприклад, купити звичайний холодильник без підключення до інтернету. Справа полягає у тому, що, в майбутньому, вже може не бути звичайних пристроїв і тоді, при відповідних наслідках, жертвами уразливих місць речей може бути кожен.

Кожен пристрій інтернет речей не може бути захищений від будь яких вразливостей, завжди було що, щоб щось працювало добре чи стабільно, жертвують іншим, у даному випадку зручністю таких пристроїв є їхня мобільність, технологічність і простота у використанні. Для отримання таких критеріїв виробники пожертвували захистом.

Вразливість пристроїв інтернет речей поділяється на два типи, взаємодія пристроїв між собою через технології передачі даних і апаратне ураження. Якщо зараз, всі дослідження направлені на написання і доопрацювання безпеки протоколів технології передачі даних, то такі чинники, як захист інфраструктури життєзабезпечення функціонування приладів, не завжди беруть до уваги. Майже з стовідсотковим шансом можна сказати, що при встановленні розумних датчиків нехтують перевірянням станів і захищеності ліній електропередач. В таких випадках, потрібно завжди перевіряти де і кудюю проходять ці лінії, чи існує резервний кабель підключення, котрий при виході з ладу основного, зможе взяти підтримку функціонування приладу на себе. Зрозуміло, що серед сотні приладів в одній будівлі, вихід з ладу одного-чи декількох не є суттєвою проблемою, але інколи, такий збій може нести за собою інші причини. Одні з найстаріших методів підслуховування, перехоплення

трафіку є підключення по типу «man in the middle». В такому випадку, ураження системи в загалом може виникнути не через прилад Інтернету речей, а через засіб його підтримання.

На апаратному рівні дуже великою популярністю використовуються радіочастотна ідентифікація(RFID мітки). Ці мітки встановлюються на такі прилади як перепустки, ключі, аудіо і відео контроль, системи електронних платежів, збір даних як засіб отримання інформації на датчиках. Особливістю і тим самим уразливістю цієї технології є:

- безконтактне спрацювання і передача інформації, відсутня необхідність видимого контакту;
- змога записувати і зчитувати інформацію за отриманням сигналів;
- можливість спрацювання без джерела енергії, спрацювання як датчик;
- легкість у використанні і встановленні, великий шанс пошкодження;
- впровадження інформації на пристрій без попередньої перевірки;
- можливість впровадження нового пристрою у мережу з відповідними частотами;
- отримання даних про місцеположення за сигналом міток.

Ще одна технологія NFC, котра принесла дуже зручний спосіб розрахування картками і телефонами, має ті уразливі місця, що RFID мітки, окрім деяких особливостей. NFC це дуже близька безконтактна технологія обміну інформацією, обмін відбуваються на відстані не більше 10см. Попри зручність даної технології, існує велика можливість отримати троянську програму на пристрій і втратити персональну інформацію, паролі чи сам дост. Ще одні з найрозповсюдженіших методами отримання доступу чи впровадження неправдивої інформації через технологією NFC - є перехоплення сигналів антенами, зміна інформації за запитом без перевірки(запит від неочікуваного приладу), атаки з використанням ретрансляторів. За стандартом, сама технологія не гарантує безпеки передачі даними в цілому, за безпеку

обміну інформації в даний час несуть відповідальність програми, які використовують дану технологію передачі, шифрування пакетів і перевірки.

QR – технологія штрих кодів нового рівня, їх називають фізичними гіперсилками, підпис, за котрим можна легко отримати контакти людини, перейти на сайт приладу в інтернет магазині чи завантажити вірус на пристрій по посиланню. Попри зручність, без спеціального програмного забезпечення перевірки, людина ніяким чином не може визначити правдивість даної картинки-адреси, котра може бути так само і набором певних команд для зміни налаштування скануємого пристрою.

Однією з найрозповсюдженіших методів отримання доступу через механізми передачі даних є втручання у бездротову мережу. Це засвідчує про уразливості у інтерфейсі IEEE 802.11, а також внутрішніх атак. Уразливості

через аутентифікація і шифрування не завжди є ключовими цілями для проникнення. Внутрішні атаки – це прилади, чи чинники які викликають уразливість у мережі всередині системи захисту. У ролі збудника внутрішньої атаки може бути будь-що, що має доступ до бездротової чи дротової мережі.

Усі бездротові мережі в будь-якому випадку з'єднуються з дротовою мережею, відповідно, при проникненні у дротову мережу і навпаки, ураження розповсюджується по всім приладам. Ураження може проходити через програмно-апаратні методи захисту локальних мереж:

- Мережеві екрани;
- Криптографічний захист;
- Антивірусні програми;
- Біометричні технології аутентифікації;
- Віртуальні захищені канали.

### **Висновки.**

Наразі, всі технологічні країни використовують пристрої інтернет речей, але не в кожній такій країні є стандарти і юридичні тлумачення правил і технологій цих пристроїв. Ураження пристроїв з котрими кожного дня зіштовхуються користувачі мають одне єдине пояснення, відсутність стандартів з безпеки інформації. Інтернет речі слугують дуже великою зручністю, але при цьому жертвують безпекою інформації. Стрімкий ріст приладів інтернет речей пояснює тільки те, що у майбутньому вони все більше набиратимуть популярність у всіх сферах людської життєдіяльності. На даний час стає актуальним питанням при розробці і впровадженні рішень щодо запобігання отримання несанкціонованого доступу до таких приладів.

### **ДЖЕРЕЛА**

1. Аналіз проблем безпеки інтернету речей. Кисиленко В.К., студент (гр. РС-42, РТФ НТУУ «КПІ ім. Ігоря Сікорського»).
2. Науково-практична конференція 24.10.2017 «ІНТЕРНЕТ РЕЧЕЙ: Проблеми правового Регулювання та впровадження». Методи зовнішнього втручання у технології передачі даних інтернет речей. Огородников Д.В.
3. Блокчейн, як ДНК-протокол для Інтернет Речей, Роман Кравченко. ІОТ Конференція Росія 2018.
4. Khan, R. [and others], Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges, Frontiers of Information Technology (FIT), 2012 10th International Conference on. — 2012. — 257-260 с.
5. Науково-практична конференція 24.10.2017 Некіт К. Г. Інтернет речей та інформаційна безпека: деякі правові проблеми.
6. Bill Glover, Himanshu Bhatt ,RFID essentials, O'Reilly Media, Inc., 2006 ISBN 0-596-00944-5, p. 88-89.
7. Fisher, Jill A. 2006. Indoor Positioning and Digital Management: Emerging Surveillance Regimes in Hospitals. In T. Monahan (Ed), Surveillance and Security: Technological Politics and Power in Everyday Life (pp. 77–88).

## **Огляд бенчмарків, їх цілей та склад, активність з використання в ТММО і науці в цілому**

Вибір, розробка або порівняння методів машинного навчання в області інтелектуального аналізу даних може бути складним завданням, заснованим на цільовій задачі і цілі конкретного дослідження. Численні загальнодоступні реальні і змодельовані базові набори даних з'явилися з різних джерел, але їх організація і прийняття в якості стандартів були непослідовними. Таким чином, вибір і вивчення конкретних контрольних показників («бенчмарков») залишається непотрібним тягарем для фахівців з комп'ютерного навчання і вчених-дослідників. Внаслідок цього, були розроблені доступні, перевірені «бенчмарки», які можуть бути використані як спосіб ідентифікації ефективності різних методологій машинного навчання. Ця робота представляє огляд цілей та структури основних запропонованих «бенчмарків», аналіз їх використання в теорії машинного навчання.

The choice, development or comparison of machine learning methods in the field of data mining can be a complex task based on the objective and purpose of a particular study. Numerous publicly available real and simulated databases originated from a variety of sources, but their organization and acceptance as standards were inconsistent. Thus, the choice and study of specific benchmarks ("benchmark") remains an unnecessary burden for computer scientists and researchers. As a result, accessible, proven "benchmarks" have been developed that can be used as a way to identify the effectiveness of various methodologies in machine learning. This paper presents an overview of the goals and structure of the main proposed benchmarks, an analysis of their use in the theory of machine learning.

### **KEYWORDS**

Benchmarks, architecture, neural network, convolution, convolutional networks, models, deepbench, dawnbench, mlperf, pmlb

### **1. Вступ**

Бенчмарки використовуються в машинному навчанні для перевірки і оцінки використовуваних методів МО в розрізі їх здатності вивчити текстові дані в наборах «еталонних» шаблонів. Частіше за все, очікується, що використовуваний метод буде успішно працювати на вже перевірених даних і зможе надати, як мінімум, схожі або кращі результати розпізнання ніж «еталонні» методи навчання. Порівняння ефективності може бути в рамках швидкості навчання, точності та ефективності розпізнання, складності обчислень, «стресостійкості» методу.

Набори тестових даних зазвичай відносяться до одного з трьох видів. Це або добре вивчені дані з реального світу,

взяті з конкретної предметної області, або дані, що були згенеровані, штучно створені, і в цілому «виглядають» як реальні дані, або, знову ж таки, згенеровані дані, в яких можливі не реалістичні показники. Відповідно і показники ефективності навчання на різних типах датасета відрізняються. Частіше за все, один і той же метод показує кращі результати розпізнавання на даних, згенерованих, але перевірених на «реалістичність», потім слідує реальні дані, і потім дані без перевірки. Втім, можливі і включення в конкретних наборах перевірочних даних, в наслідок більшого розкиду тестових даних у 3-го типу набору тестових даних.

### **2. Постановка завдання**

Насамперед, при всій різноманітності бенчмарків в конкретних областях, але, в

цілому, МО все так само не має центрального, уніфікованого набору базових даних, для перевірки сильних і слабких сторін більшості відомих методів МО. «Бенчмаркінг» спеціалізується вирішенні якийсь вузько певної проблеми, на кшталт розпізнання символів латиниці, або просто для 2-смугових епістатичних взаємодій, Але не для вирішення просторової задачі.

На даний момент, хоча і існують такі бази даних як Kaggle або OpenML, в яких можна вільно знайти тисячі наборів даних, які можна використовувати для навчання розв'язування задачі класифікації, потрібна значна попередня обробка перед тим як вони будуть використані.

Отже при бажанні використати будь-яку методологію з можливістю обробки декількох областей, є необхідність шукати та використовувати декілька бенчмарків, від різних авторів, які використовували зовсім різні умови при тестуванні.

### 3. DeepBench

Основна мета DeepBench полягає в тестуванні операцій, важливих для глибокого навчання на різних апаратних платформах.

DeepBench використовує бібліотеки нейронних мереж для тестування базових операцій на різних апаратних засобах. Це не працює з глибокими навчальними рамками або глибокими моделями навчання, побудованими для додатків. Ми не можемо виміряти час, необхідний для тренування всієї моделі, використовуючи DeepBench. Ефективність характеристик моделей, побудованих для різних додатків, дуже відрізняється один від одного. Тому ми проводимо порівняльну оцінку основних операцій, пов'язаних із глибокою моделлю навчання. Бенчмаркінг цих операцій допоможе підвищити рівень поінформованості серед постачальників обладнання та розробників програмного забезпечення про вузькі місця в процесі глибокого навчання та висновків.

Згортання складають переважну більшість флопу в мережах, що працюють на зображеннях і відео, і утворюють важливі частини мереж, таких як моделювання мови та природної мови, що робить їх, мабуть, найважливішим шаром з точки зору продуктивності.

Згортання мають 4 або 5-мірні входи та виходи, що призводить до великої кількості можливих упорядкування цих вимірів. Для першої версії тесту ми стурбовані лише продуктивністю у форматі NCHW, тобто дані представлені в зображеннях, функціональних картах, рядках і стовпчиках.

### 4. DAWNBench


DAWNBench це відкритий бенчмарк для повноцінного машинного навчання та аналізу результатів. Замість того, щоб просто вимірювати час на ітерацію чи пропускну здатність, DAWNBench вимірює кінцевий результат навчання (наприклад, час, вартість) та висновок (наприклад, затримка, вартість) на заданому рівні точності. Це забезпечує об'єктивний засіб нормалізації відмінностей в обчислювальних рамах, апаратних засобах, алгоритмах оптимізації, параметрах гіперпараметра та інших факторах, що впливають на реальну продуктивність.

DAWNBench оцінює глибокі системи навчання за різними завданнями на основі кількох показників, використовуючи декілька наборів даних. Тест дозволяє інновації в програмному забезпеченні, алгоритмах, методах комунікації тощо. Завдяки визначенню завдання DAWNBench також дозволяє експериментувати з новими архітектурами та апаратними моделями. У початковій версії складаються записи про два завдання: класифікація зображень на CIFAR10 і ImageNet, а також питання відповіді на SQuAD, а також оцінювати за чотирма показниками: час навчання до вказаної точності перевірки, вартість (у доларах США) навчання до вказаної валідації точність з використанням загальних

хмарних екземплярів, середня латентність виконання висновку на окремий елемент (зображення або питання) та середня вартість висновку для 10 000 елементів.

### 5. PMLB

PMLB це представлений публічний "набір наборів даних", реалізований на основі 165 різних датасета, розроблених для оцінки ефективності методів класифікації. Але крім простого запозичення даних, була проведена стандартизація формату даних і запропонований зручний інтерфейс для вилучення датасета прямо з Інтернету. Більшість подібних датасета мають більш 5000 записів з виділеними 500 і більше особливостями. Приблизно половина з цих датасетів це проблеми бінарної класифікації, а друга половина відносяться до проблем класифікації мультікласів з варіаціями від 3 до 26 класів. Усі категоріальні функції та мітки з не числовими кодуваннями були замінені числовими еквівалентами (наприклад, "Низький", "Середній" та "Високий" були замінені 0, 1 і 2). Крім того, у кожному наборі даних стовпець залежної змінної позначено як "клас". Для множинних наборів даних ми видалили екземпляри будь-якого класу, у якому було менше 10 примірників для цього класу, оскільки <10 примірників занадто мало, щоб навмисно навчатися. Нарешті, усі бази даних з відсутніми даними були виключені з PMLB, тому що багато алгоритмів ML не можуть обробляти відсутні дані у своїх стандартних реалізаціях, і ми хотіли уникнути накладання певного методу для врахування даних в цьому початковому дослідженні.  $\hat{y}(x)$ :

 В даний час PMLB складається з наборів даних для контрольованої класифікації (бінарний та багато-класовий). Передбачається знаходження відображення, що пов'язує вектор функцій з мітками класу з множини, використовуючи N парних прикладів з набору тренувань.

### 6. MLPerf

Метою MLPerf є створення спільного набору тестів, що дає змогу в полі машинного навчання (ML) вимірювати продуктивність системи в кінцевому підсумку як для навчання, так і для висновків з мобільних пристроїв на послуги хмарної служби.

MLPerf підтримує такі вирішення таких проблем, як:

- Класифікація зображень
- Виявлення об'єкта
- Вимова до тексту
- Переклад
- Рекомендація
- Аналіз настроїв
- Підвищення кваліфікації

Також слід відмітити 2-х сторонній підхід:

Відділ Closed Model MLperf визначає модель, яка буде використовуватися, і обмежує значення гіперпараметрів, наприклад розмір партії та швидкість навчання, з акцентом на чесні порівняння апаратних та програмних систем. ("Еквівалент сорту" називається "Дайтона", натякаючи на запасні автомобілі в гонці на Майлі 500 км в штаті Дайтона).

У відділі Open Model MLperf конкуренти повинні вирішувати ту ж саму проблему, використовуючи один і той же набір даних, але з меншим обмеженням, при цьому акцент робиться на покращенні сучасного рівня ML. ("Еквівалент сорту" отримав назву "Indy", що називає навіть більш швидкі Формули-1 спеціальні гоночні автомобілі, розроблені для подій, таких як Indianapolis 500.) Слідуючи прецеденту DAWNbench, основна метрика MLperf для тренувань визначається як час настінного годинника, щоб тренувати модель до мінімальної якості (часто в годинах). Кожен еталон має показник якості, визначений його оригінальними авторами.

### 7. 1NWPU-RESISC45

Представлений набір даних NWPU-RESISC45 складається з 31 500

зображень дистанційного зондування, поділених на 45 сцени класів. Кожен клас включає 700 зображень розміром  $256 \times 256$  пікселів у колірному просторі червоно-синього кольору (RGB). Просторова роздільна здатність

коливається від приблизно 30 м до 0,2 м на піксель для більшості сцен класів, за винятком класів острова, озера, гори та сніговика, які мають менші просторові дозволи.



**Рис. 1. Приклад зображень с датасету NWPU-RESISC45**

Пропонований набір даних NWPU-RESISC45 має наступні три важливі характеристики:

- Великий масштаб: Воскресіння глибокого навчання мала революційний вплив на сучасні сучасні технології машинного навчання та комп'ютерного бачення. Важливим фактором його успіху є наявність великомасштабних наборів даних, що дозволяє глибоким мережам розвивати свій повний потенціал
- Різноманітні варіанти зображення: толерантність до варіацій зображення - важлива властивість будь-якої системи класифікації сцени, будь то людина чи машина. Проте більшість існуючих наборів наборів не дуже багаті за варіаціями зображення.
- Високий ступінь у класовому розмаїтті та між класовою подібністю. Зображення були отримані під всілякими умовами і додано кілька більш дрібнозернистих сцени класів з високим семантичним перекриттям, таких як кругові сільськогосподарські угіддя та прямокутні сільськогосподарські угіддя, торговельна зона та промислова зона.

## 8. SD-VBS

The San Diego Vision Benchmark Suite (SD-VBS) - це набір різноманітних програм зору, витягнутих з області зору. Він покликаний допомогти архітекторам, авторам-компіляторам, а системним дизайнерам вивчити побудову майбутніх систем, які відрізняються від перспективних застосувань. Крім того, коди зору, як правило, мають достатню кількість паралелізму, що робить їх хорошим кандидатом для формування майбутніх багатоядерних та паралельних архітектур.

Для покращення розуміння навантаження на бачення для багатопроцесорних систем, або для спеціалізованих платформ, або для систем компіляції та виконання, було розроблено стандартний набір, який називається SD-VBS, який включає в себе дев'ять важливих програм зору з різноманітного набору зоряних доменів. Самі дев'ять додатків, у свою чергу, складаються з колекції понад 28 нетривіальних обчислювальних інтенсивних ядер.

Для кожного з цих програм надається як MATLAB, так і C версія. MATLAB, як

правило, є оригінальною мовою вихідних орієнтирів, що кодується дослідниками зору. Версії C надаються, щоб зробити дослідникам архітектури легше працювати з орієнтирами. Вони були закодовані в "чистому" підмножині C, що дозволяє уникнути непотрібного використання покажчиків, спадщини або машин-специфічної оптимізації, а також функцій мови C, які роблять аналіз та перетворення складними. Мета полягає в тому,

щоб полегшити аналіз, трансформацію, розпаралелювання і моделювання тестів дослідниками компілятора та архітектури.

Кожен еталон забезпечує входи трьох різних розмірів, що дає змогу архітекторам контролювати час моделювання, а також зрозуміти, як додаток масштабується з більш складним введенням.

## 9. Висновок

В результаті аналізу популярних бенчмарків та датасетів для них, для подальшої роботи був обраний датасет MNIST, як датасет, що є основним примірником для більшості датасетів для класифікації зображень. MNIST представляє собою датасет, що містить біля 70000 уніфікованих зображень рукописних символів, та є одним з найпопулярніших датасетів для тестування різноманітних методологій.

**Табл.1.Опис SD-VBS бенчмарків**

Бенчмарк	Опис	Характеристики	Область використання
Disparity Map	Обчислює інформацію про глибину за допомогою щільного стерео	Дані, що надаються зовні	Robot vision for Adaptive Cruise Control, Stereo Vision
Feature Tracking	Витяг руху з послідовності зображень	Дані, що надаються зовні	Robot vision for Tracking
Image Segmentation	Розбиття зображення на концептуальні регіони	Обчислювальні дані	Medical imaging, computational photography
SIFT	Витяг інваріантних функцій з спотворених зображень	Обчислювальні дані	Object recognition
Robot Localization	Визначення місцезнаходження на основі навколишнього середовища	Обчислювальні дані	Robotics
SVM	Контрольований метод навчання для класифікації	Обчислювальні дані	Machine learning
Face Detection	Визначення обличчя в зображенні	Обчислювальні дані	Video Surveillance, Image Database Management
Image Stitch	Смужка, що перекриває зображення, використовуючи вирівнювання та відповідність на основі функцій	Зовнішні та обчислювальні ресурси	Computational photography



Табл.2. Датасети, що використовуються для тестів розпізнавання зображень

Датасет	Зображення /клас	Кількість класів	Загальна Кількість зображень	Просторова роздільна здатність	Розмір зображень	Рік
UC Merced Land-Use ( <a href="http://weegeevision.ucmerced.edu/datasets/landuse.html">http://weegeevision.ucmerced.edu/datasets/landuse.html</a> )	100	21	2,100	0,3	256×256	2010
WHU-RS19( <a href="http://www.xinhuafluid.com/people/yangwen/WHU-RS19.html">http://www.xinhuafluid.com/people/yangwen/WHU-RS19.html</a> )	~50	19	1,005	0,5	600×600	2012
SIRI-WHU( <a href="http://www.lmars.whu.edu.cn/prof_web/zhongyanfei/e-code.html">http://www.lmars.whu.edu.cn/prof_web/zhongyanfei/e-code.html</a> )	200	12	2,400	2	200×200	2016
RSSCN7( <a href="https://www.dropbox.com/s/j80iv1a0mvhonsa/RSSCN7.zip?dl=0">https://www.dropbox.com/s/j80iv1a0mvhonsa/RSSCN7.zip?dl=0</a> )	400	7	2,800	-	400×400	2015
RSC11( <a href="https://pds.nasa.gov/ds-view/pds/viewProfile.jsp?dsid=ULY-J-SCE-1-ROCC-V1.0">https://pds.nasa.gov/ds-view/pds/viewProfile.jsp?dsid=ULY-J-SCE-1-ROCC-V1.0</a> )	~100	11	1,232	0,2	512×512	2016
Brazilian Coffee Scene( <a href="https://github.com/jeradhoy/DeepSatelliteData">https://github.com/jeradhoy/DeepSatelliteData</a> )	1438	2	2,876	-	64×64	2015
NWPU-RESISC45 ( <a href="http://www.escience.cn/people/JunweiHan/NWPU-RESISC45.html">http://www.escience.cn/people/JunweiHan/NWPU-RESISC45.html</a> )	700	45	31,500	~30 to 0.2	256×256	2016
MNIST( <a href="http://yann.lecun.com/exdb/mnist/">http://yann.lecun.com/exdb/mnist/</a> )	~35000	2	70,000	-	28×28	1999
ImageNet( <a href="http://www.image-net.org/">http://www.image-net.org/</a> )	~500	21481	14197122	-	256×256	2010
COCO( <a href="http://cocodataset.org/">http://cocodataset.org/</a> )	~2000	171	330000	-	256×256	2014

### Список літератури

1. Randal S. Olson «PMLB: a large benchmark suite for machine learning evaluation and comparison» Dec 11, 2017
2. Roger Chan «Setting Benchmark in machine learning» May 16, 2018
3. John Russel «Will New Machine Learning Benchmark Help Propel AI Forward» May 2, 2018
4. David Patterson «MLPerf: SPEC for ML» May 2, 2018
5. Dan Ciresan «Multi-column Deep Neural Networks for Image Classification» February 2012
6. Gong Cheng «Remote sensing image scene classification benchmark and state of the art»
7. Qin Zou «Deep Learning Based Feature Selection for Remote Sensing Scene Classification» November 2015

ТКАЧЕНКО М.С.,  
ЛИЦУК К.І.

## ДОСЛІДЖЕННЯ АРХІТЕКТУРИ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ РОЗПІЗНАВАННЯ 68 КЛЮЧОВИХ ТОЧОК ОБЛИЧЧЯ

В даній статті розглянута задача розпізнавання обличчя по фотографії з використанням ключових точок. Запропонована архітектура нейронної мережі з блоками, які можна налаштовувати, для вирішення задачі знаходження 68 ключових точок обличчя. Досліджена ефективність запропонованої мережі для розпізнавання облич.

### КЛЮЧОВІ СЛОВА

Архітектура, нейронна мережа, розпізнавання обличчя, розпізнавання напрямку погляду людини, 68 ключових точок обличчя, згортка, згорткові мережі, архітектура hourglass, архітектура unet, налаштовуваність.

In this paper, the problem of face recognition in a photo with the use of facial key points is considered. Architecture of the neural network with blocks that can be configured to solve the problem of finding 68 key points of the face was proposed. The efficiency of the offered network for facial recognition is investigated.

### KEYWORDS

Architecture, neural network, face recognition, eye(gaze) tracking, 68 face landmarks, convolution, convolutional networks, architecture hourglass, architecture unet, configurability.

### 1. Вступ

Обличчя є найвиразнішою характеристикою людини. Тому не дивно, що аналіз обличчя використовується при вирішенні дуже різноманітного кола задач, таких як розпізнавання обличчя, розпізнавання емоцій, розпізнавання міміки, розпізнавання напрямку погляду людини (eye tracking), тощо.

На сьогодні існує велика кількість математичних методів, котрі використовують для розпізнавання облич. Серед них часто використовуваними є

наступні: метод головних компонент, котрий використовує перетворення Карунена-Лоева, метод порівняння з еталоном, нейронні мережі, приховані марковські моделі [7]. Більшість алгоритмів мають наступний недолік: для розпізнавання необхідне фронтальне зображення обличчя майже без відхилень. Для усунення цього недоліку використовують крім самої фотографії обличчя, часто використовуються ще і його ключові точки. Необхідність використання для розпізнавання

обличчя ключових точок обґрунтовується тим, що на фотографіях людина не завжди може дивитись прямо на камеру, інколи обличчя частково повернуте в сторону. В цьому випадку виділяють основні точки, котрі необхідно використовувати для розпізнавання. Їх кількість та розміщення можуть варіюватися в залежності від обраної задачі та наявних баз даних. Найчастіше використовують 68 точок.

Існує і багато інших варіантів ключових точок. Як було сказано вище, вибір кількості і їх розміщення залежить від конкретної задачі, наявних баз даних і наявних ресурсів. Але найчастіше використовуваними є саме 68 і 5 точок, тому що для них існує реалізація у бібліотеках `dlib` та `opencv`[5].

Але у `dlib` та `opencv` є недоліки – вони погано знаходять точки, якщо на фотографії повернута голова або погане освітлення.

## 2. ОГЛЯД ЛІТЕРАТУРИ

Статтею-натхненням стала архітектура `hourglass`[1]. Ключовою особливістю архітектури `hourglass` є її налаштовуваність, тобто мережа може складатися з будь-якої послідовної кількості блоків `hourglass`. На вхід мережі подається зображення, на виході та між блоками вона видає `heatmap` ймовірних точок. `Heatmap` («карта тепла») – це розподіл ймовірностей, тобто у місцях де знаходяться точки вона найбільша. У статті[1] ця мережа

використовується для виявлення положення тіла людини у просторі(пози). Але у даній роботі ця архітектура використовується для знаходження 68 ключових точок.

Сам блок `hourglass` складається з послідовного збільшення кількості фільтрів («донизу»), з подальшим зменшенням кількості фільтрів («доверху»). Кількість фільтрів на вході співпадає з кількістю фільтрів на виході.

Інша архітектура `unet`[5], яка створювалася для того, щоб ефективно працювати з аугментованими даними. Сама архітектура `unet` подібна до одного блоку `hourglass`. Спочатку послідовне «зниження» зі збільшенням фільтрів для виділення контексту та ключових характеристик, потім «підйом» зі зменшенням фільтрів. У самій статті ця архітектура використовувалася для сегментації біомедицинських знімків. Під час дослідження різних архітектур `unet` застосовувалася не для сегментації як в оригінальній статті, а знову ж для знаходження `heatmap` – знаходження 68 точок. Але використання `unet` не дало бажаних результатів.

І останній цікавий знайдений підхід, про який буде йти мова – це використання оптичного потоку(`optical flow`) і `Lucas Kanade loss` для навчання без розмітки(`unsupervised`)[6].

Принцип роботи `supervision-by-registration(SBR)`[6]

заключається в наступному: беруться розмічені зображення та нерозмічені відео. З розмічених зображень тренується мережа, яка знаходить точки. Через цю мережу пофреймово пропускається відео. Також пофреймово з точок попереднього кадру за допомогою оптичного потоку та алгоритму Лукаса Канаде робляться точки для наступного. І порівнюється положення цих точок – різниця між ними є частиною функції кошовності та впливає на зворотнє розповсюдження (backpropagation).

### 3. ЗАПРОПОНОВАНЕ РІШЕННЯ

В результаті проведеного дослідження літератури було вирішено створити нейронну мережу з частинами, які можна налаштувати.

Налаштуванню підлягають наступні частини:

- звичайні (резидуальні) блоки;
- reduction блоки, тобто блоки зменшення розмірності;
- кількість послідовних блоків hourglass;
- глибина кожного блоку hourglass;
- кількість фільтрів для кожного підблоку блоку hourglass;
- два режими об'єднання минулих блоків;
- кількість вихідних точок.

Тепер детальніше про кожну з частин, які можна налаштувати.

Звичайні (резидуальні) блоки можуть бути трьох типів:

- ті, які використані в статті hourglass[1];
- блок bottleneck з статті [2];
- блок block35 з архітектури Resnet[3].

У системі передбачено два способи зменшення розмірності:

- max pooling;
- блок inceptionA з архітектури Inception.

Для кожного підблоку можливо налаштувати глибину рівнів та кількість фільтрів на кожному рівні. Це важливо і зручно в тому числі і для того, щоб зменшити розмір нейронної мережі, що дуже впливає на швидкодію системи, яка її використовує, бо зараз все більше і більше систем, які вирішують задачі машинного навчання використовують у браузерях або на мобільних телефонах.

Також налаштуванню підлягають режими конкатенації та сумачії на підйомі у блоці. Якщо при налаштуванні для різних рівнів використовуються різні розміри фільтрів, то тоді необхідно використовувати конкатенацію, інакше розмірності не зійдуться. Якщо ж всі розміри фільтрів для всіх рівнів однакові, то можна використовувати суму.

Ще одним ключовим моментом є можливість налаштувати кількість необхідних точок на виході. Таким чином ця система підходить не тільки для задачі зі знаходженням 68 точок, а і для

оригінальної задачі зі знаходженням пози людини у просторі, та будь-якої задачі, у якій необхідно знаходити точки на зображення. Наприклад, ця система використовувалася для знаходження трьох точок ока, які у свою чергу потрібні для знаходження зіниці.

#### 4. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Було досліджено 5 комбінацій різних налаштовуваних блоків.

1. Повне повторення архітектури hourglass (4 послідовних блоки з 4 рівнями, на кожному рівні однакова кількість фільтрів - 128, резидуальний блок – hourglass, на підйомі режим сумачії, для зменшення розмірності – max\_pooling). Далі рожевий графік.

2. 4 послідовних блоки з 4 рівнями, на кожному рівні однакова кількість фільтрів - 128, резидуальний блок – vnesk, на підйомі режим сумачії, для зменшення розмірності –

max\_pooling. Далі оранжевий графік.

3. 6 послідовних блоки з 4 рівнями, на кожному рівні різна кількість фільтрів - [8,16,32,64], резидуальний блок – vnesk, на підйомі режим конкатенації, для зменшення розмірності – inception. Далі сірий графік.

4. 4 послідовних блоки з 4 рівнями, на кожному рівні різна кількість фільтрів - [16, 32, 64, 128], резидуальний блок – hourglass, на підйомі режим конкатенації, для зменшення розмірності – max\_pooling. Далі зелений графік.

5. 4 послідовних блоки з 4 рівнями, на кожному рівні однакова кількість фільтрів - 64, резидуальний блок – block35, на підйомі режим сумачії, для зменшення розмірності – max\_pooling. Далі голубий графік.

Результати навчання у вигляді графіку значення функції кошовності від ітерації наведені на Рисунку 1.

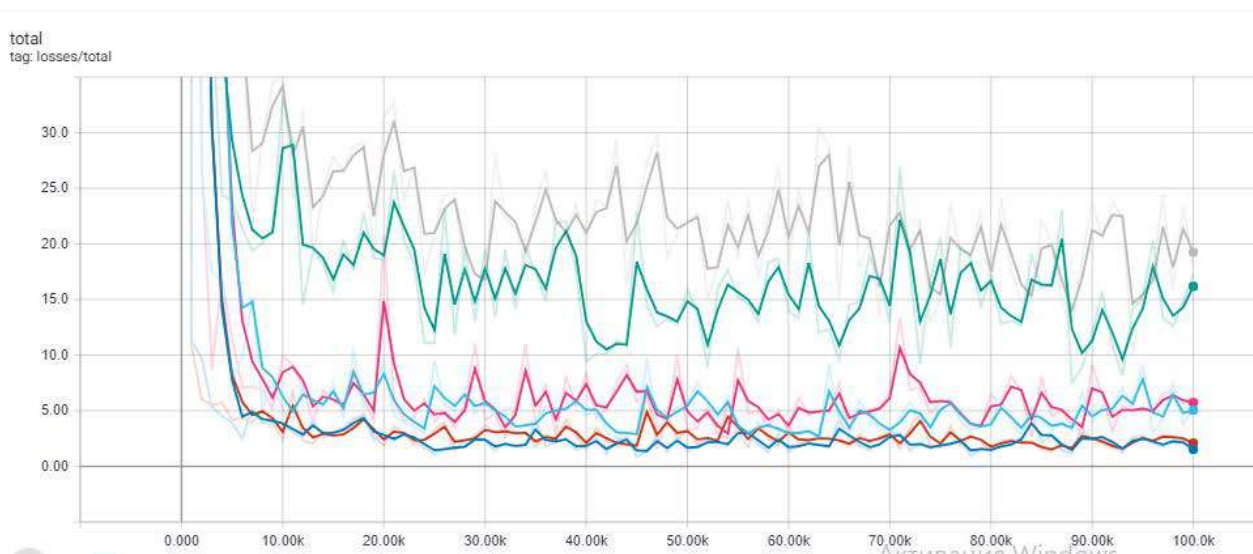


Рисунок 1 - Результати навчання у вигляді графіку

Як видно з графіку найменший лос дала архітектура під номером 2(червоний, нижній графік). Її фактичні результати показані на Рисунках 2, 3 і 4.

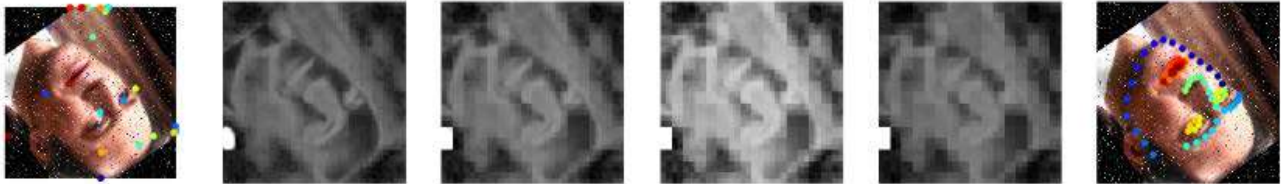


Рисунок 2 – Результат роботи нейронної мережі на 1 ітерації

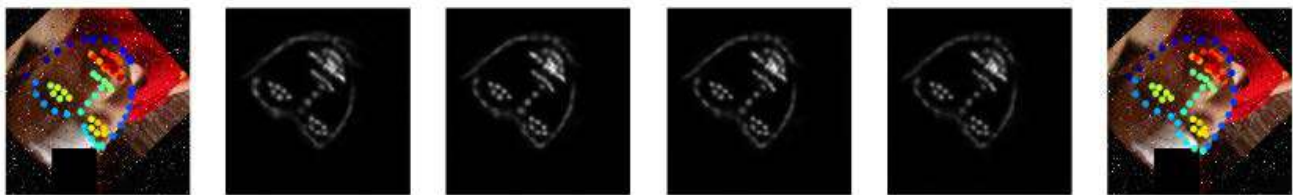


Рисунок 3 – Результат роботи нейронної мережі на 46900 ітерації

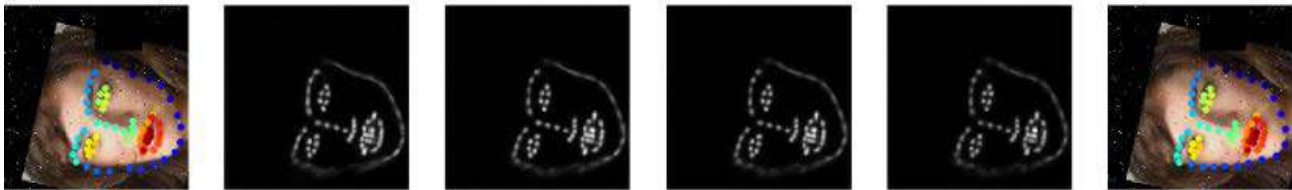


Рисунок 4 – Результат роботи нейронної мережі на 92600 ітерації

## 5. Висновки

В роботі запропонована архітектура нейронної мережі для розпізнавання 68 ключових точок обличчя. Найкращою за функцією оцінки виявилася архітектура з 4 послідовними блоками hourglass з 4 рівнями, де на кожному рівні однакова кількість фільтрів - 128, в якості резидуального блоку – `bnesc`, на підйомі режим використовується сумації, для зменшення розмірності – `max_pooling`. Фактичні результати, як було показано на малюнках, вона також дала хороші. Мінуси таких архітектур заключаються в тому, що для них потрібні великі обчислювальні ресурси. Це являє собою серйозну проблему для застосування таких мереж у веб-додатках. Тому майбутні роботи будуть направлені на пошук більш «легких» архітектур.

## СПИСОК ЛІТЕРАТУРИ

1. Stacked Hourglass Networks for Human Pose Estimation [Електронний ресурс] / A. Newell, K. Yang, J. Deng – Режим доступу до ресурсу: <https://arxiv.org/pdf/1603.06937.pdf>.
2. Deep Residual Learning for Image Recognition [Електронний ресурс] / H.Kaiming, Z. Xiangyu, R. Shaoqing, S. Jian – Режим доступу до ресурсу: <https://arxiv.org/pdf/1512.03385.pdf>.
3. Going deeper with convolutions [Електронний ресурс] / [C. Szegedy, W. Liu, Y. Ji та ін.] – Режим доступу до ресурсу: <https://arxiv.org/pdf/1409.4842.pdf>.
4. Rosebrock A. Facial landmarks with dlib, OpenCV, and Python [Електронний ресурс] / Adrian Rosebrock – Режим доступу до ресурсу: <https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/>.
5. Ronneberger O. U-Net: Convolutional Networks for Biomedical Image Segmentation [Електронний ресурс] / O. Ronneberger, P. Fischer, T. Brox – Режим доступу до ресурсу: <https://arxiv.org/pdf/1505.04597.pdf>.
6. Supervision-by-Registration: An Unsupervised Approach to Improve the Precision of Facial Landmark Detectors [Електронний ресурс] / [X. Dong, S. Yu, X. Weng та ін.] – Режим доступу до ресурсу: <https://arxiv.org/pdf/1807.00966.pdf>.
7. Mankar V. A Review Paper on Face Recognition Techniques [Електронний ресурс] / V. Mankar, S. Bhele – Режим доступу до ресурсу: [https://www.researchgate.net/publication/270583005\\_A\\_Review\\_Paper\\_on\\_Face\\_Recognition\\_Techniques](https://www.researchgate.net/publication/270583005_A_Review_Paper_on_Face_Recognition_Techniques).

*Кахерський О.І.  
Гавриленко О.В.*

## **ДОСЛІДЖЕННЯ АЛГОРИТМУ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ АВТОМАТИЧНОЇ КЛАСИФІКАЦІЇ ОБЛИЧЬ**

В даній статті розглянута задача автоматичної класифікації облич на зображеннях. Запропонована модифікація методу (автори Ladislav Lenc, Pavel Král) класифікації облич, що робить його більш гнучким саме для автоматичної класифікації. Виконано дослідження ефективності алгоритму на базі даних ORL.

### **КЛЮЧОВІ СЛОВА**

Автоматична класифікація облич, визначення локальних ознак, визначення ключових точок обличчя, SIFT, SURF, метод k середніх.

The task of automated face recognition has been investigated in this article. A modification of the Ladislav Lenc, Pavel Král face recognition method, that makes it more flexible for this class of tasks, has been proposed. Efficiency research has been fulfilled based on ORL database.

### **KEYWORDS**

Automated face recognition, local feature detection, face keypoints detection, SIFT, SURF, K-Means.

### **1. Вступ**

Задача розпізнавання облич зустрічається в широкому колі комерційного програмного забезпечення (наприклад, автентифікація користувача системи по обличчю або зіницям), а також систем безпеки (наприклад, в аеропортах чи правоохоронних органах). Саме тому за останні роки було представлено багато підходів для систем розпізнавання. Загалом ці методи діляться на дві категорії: виконання розпізнавання з урахуванням всього обличчя або ж порівняння окремих його частин. До останньої групи належать методи, що базуються на алгоритмах SIFT та SURF, запропоновані для пошуку ключових точок зображення та обчислення їх основі дескрипторів, які є незалежними від зміни кутів нахилу обличчя, освітлення та масштабу, з метою їх подальшого порівняння. Слід також зауважити, що підходи також можна розділити за можливістю використання тестової вибірки для подальшої класифікації.

В даній статті запропонована модифікація існуючого підходу, який може бути використаний в системах автоматичного розпізнавання облич, адже він може віднаходити співпадіння на нерозмічених даних.

У розділі 2 наводиться опис існуючих рішень, на яких базується модифікація, у розділі 3 – самої модифікації, в останньому – деталі та результати експерименту.

### **2. Опис існуючих рішень**

#### **2.1. Опис методів для знаходження ключових точок та обчислення їх дескрипторів**

SIFT - алгоритм із області комп'ютерного зору, який виявляє і описує локальні ознаки зображення. Він використовується, коли необхідно виконати розпізнавання об'єкту на зображеннях, які за своєю природою можуть мати різний масштаб, освітлення та кути нахилу.

Першим етапом цього алгоритму є знаходження екстремумів в зображенні, що фільтрується різницею гаусового фільтру. Вхідне зображення поступово масштабується зі зменшенням розмірів та фільтр застосовується на всіх рівнях, забезпечуючи таким чином інваріантність масштабу. Виконується порівняння кожного пікселю зображення на поточному рівні, а також на вищому та нижчому. Якщо піксель є меншим або більшим за всі інші, він вважається потенційною ключовою точкою.

Наступним кроком є подальше фільтрування знайдених точок. Місця з



низькою контрастністю та вздовж країв відхиляються. Для точок, що залишилися, назначається напрямок, що базується на обчисленнях градієнтів в околицях точки. Отримані значення є пропорційними величиною градієнту. Обчислення виконуються на регіонах розміром 16x16 пікселів. Величина та напрямок градієнтів обчислюється в кожній точці регіону. Для кожного регіону розміром 4x4 обчислюється гістограми напрямків. Нарешті, створюється вектор, що складається з 128 значень (16x8)[1].

SURF є ще одним ефективним методом знаходження ключових точок та створення дескрипторів на їх основі. Метод також є інваріантним до поворотів зображення. Для пришвидшення знаходження ключових точок використовується інтегральне зображення. Детектор базується на матрицях Гессе. “Коробкові” фільтри (англ. Box filter) використовуються для апроксимації похідних другого порядку функції Гауса. Разом з інтегральними зображеннями цей підхід значно пришвидшує обчислення. На відміну від SIFT, жодних пірамід зображень не створюється. Замість цього, масштабуються фільтри та застосовуються на вхідному зображенні.

Для забезпечення інваріантності до поворотів, кожній ключовій точці назначається напрямок. Обчислення базуються на круговому регіоні навколо ключової точки та результуючий дескриптор складається з 64 значень [2].

Слід зауважити, що при використанні SIFT/SURF для вирішення задачі розпізнавання обличчя, можна стикнутися з проблемою неправильно співставлень ключових точок. Більшість алгоритмів, що базуються на цих методах, використовують різні схеми розбиття обличчя на частини з метою співставлення точок в їх рамках. Це дозволяє частково вирішити цю проблему, а також зменшити час роботи процесу порівняння (через порівняння точок, які відносяться до одної частини обличчя)[3].

## 2.2 Критерій подібності Керенексі

В [4] був представлений підхід для порівняння двох зображень на основі

вейвлетів Габора та успішно адаптований та інтегрований з SIFT в [5].

Керенексі запропонував під час порівняння використовувати зважену суму двох компонентів порівняння: загального значення співпадіння двох облич та кількості успішно співставлених точок.

Локальність порівняння, тобто порівняння точок спільних частин обличчя, забезпечується за допомогою введення порогового значення, яке фільтрує точки з різних регіонів.

Нехай  $T$  – тестове зображення,  $G$  – зображення з бази даних. Для кожної ключової точки  $t \in T$  визначається множина релевантних ключових точок  $g \in G$  за формулою (1):

$$\sqrt{(x_t - x_g)^2 + (y_t - y_g)^2} \leq \theta, \quad (1)$$

де  $x$  та  $y$  – координати ключових точок.

Якщо жодної релевантної точки не знайдено, точка  $t$  виключається з процедури порівняння.

Як зазначено в [6], неможливо обчислити оптимальне порогове значення аналітично, адже воно залежить від розмірів та якості зображення. Наприклад, для бази даних СТК це значення варіюється в межах [10,40].

Загальне співпадіння двох облич  $OS$  обчислюється як середнє значення співпадіннь усіх відповідних точок:

$$OS_{T,G} = \text{mean}\{S(t, g), t \in T, g \in G\}, \quad (2)$$

де  $S$  – критерій подібності косинусів.

Після цього визначається зображення з бази даних з найбільш подібною ключовою точкою до кожної ключової точки тестового зображення. Далі виконується обчислення наступного виразу:

$$G_i = \frac{C_i}{N_i}, \text{ де} \quad (3)$$

$C_i$  – кількість ключових точок  $i$ -го зображення з бази даних, які є найближчими до ключових точок тестового зображення,  $N_i$  – загальна кількість ключових точок в  $i$ -му зображенні з бази даних.

Зважена сума двох критеріїв подібності обчислюється за наступною формулою:

$$FS_{t,i} = \alpha OS_{t,i} + \beta G_i, \quad (4)$$

де  $t$  – тестове зображення;

$i$  – індекс зображення з бази даних.

Індекс зображення з БД, зважена сума якого є максимальною серед всіх зображень в БД, є результатом розпізнавання тестового обличчя. [5]

### 3. Опис алгоритму

Не зважаючи на високі показники розпізнавання [5], є два пункти, які роблять даний підхід не зовсім гнучким для вирішення певних задач, а саме:

- використання порогового значення, яке визначається лише емпірично;
- не нормалізоване значення критерію (4) та коефіцієнтів.

Для обчислення порогового значення необхідно мати тестовий набір даних, що є не зовсім можливо в деяких задачах класу автоматичного розпізнавання облич, а також значно ускладнюється процес підготовки зображення до аналізу. У свою чергу, поточна область значень критерію (4) значно ускладнює вирішення задачі розпізнавання з можливістю використання варіанту “не знайдено співпадінь”, адже порогове значення тепер також необхідно визначати емпірично і відкритим постає питання чи можна взагалі адаптувати метод до таких умов.

Одним із методів заміни порогового значення є використання методу кластеризації  $k$ -середніх. Кількість кластерів можна встановити в п'ять (два ока, ніс, ліва та права нижні частини обличчя)[7].

Альтернативою, яка використовується в даній модифікації, є виявлення координат ключових точок обличчя (очей, носу, центру губ). Цього можна досягти за допомогою використання методів детектування (наприклад, алгоритму Віюлі-Джонса) на тестових базах даних, які є у вільному доступі. Після успішного виявлення координат (центрів кластерів),

співвіднести кожен ключову точку з одним з кластерів, Евклідова відстань до якого буде найменшою.

Для нормалізації критерію (4) пропонується замість зваженої суми критеріїв обчислювати їх середнє арифметичне (або середнє зважене арифметичне, якщо необхідно вводити вагові коефіцієнти). У випадку відсутності необхідності вводити вагові коефіцієнти, вираз матиме наступний вигляд:

$$FS_{i,i} = \text{mean}(OS_{i,i}, G_i) \quad (5)$$

Така зміна виразу (4) вводить більш зрозумілу метрику подібності облич на зображеннях – процентну, тому можна встановити більш зрозуміле порогове значення і якщо не знайдено жодного значення, яке його перевищує, то вважати, що співпадінь в базі даних не знайдено.

### 4. Результати експерименту

В якості датасету була обрана база даних ORL, яка складається з зображень 40 людей (по 10 на кожну). Зображення містять лише обличчя та мають однотонний фон, в яких змінюється розміри, кути нахилу голови, умови освітлення. Розміри: 92x112 пікселів.

Одне зображення кожної людини використовується як базове, а інші 9 – тестові.

В якості детектору частин обличчя (очей, носу, губ) використовується Google Cloud Vision API.

Результати порівняння з методами, наведеними в [5] наведені в таблиці 1.

**Таблиця 1. Порівняння результатів класифікації на датасеті ORL**

Метод порівняння	Процент співпадіння
Aly	61.25
Lenc-Kral	78.75
Керенексі	80.56
Запропонований метод	81.94

### Висновки

В роботі наведено опис існуючих рішень задачі розпізнавання облич, проаналізовані недоліки адаптації критерію подібності Керенексі, описаного в [5]. На основі цього запропонована модифікація методу (автори Ladislav Lenc, Pavel Král) класифікації облич, що робить його більш гнучким саме для автоматичної класифікації, а також адаптує його під задачі, в яких є допустимим відсутність результату класифікації. Також був проведений експеримент з класифікації на датасеті ORL і отриманий результат є кращим, ніж наведено в [5].

### Список використаної літератури

1. Distinctive Image Features from Scale-Invariant Keypoints [Електронний ресурс] / David G. Lowe – Режим доступу до ресурсу: <https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf>
2. SURF: Speeded Up Robust Features [Електронний ресурс] / Herbert Bay, Tinne Tuytelaars, Luc Van Gool – Режим доступу до ресурсу: <http://www.vision.ee.ethz.ch/~surf/eccv06.pdf>
3. Adaptation of SIFT Features for Robust Face Recognition [Електронний ресурс] / Janez Križaj, Vitomir Štruc, Nikola Pavesic – Режим доступу до ресурсу: [https://www.researchgate.net/publication/221472612\\_Adaptation\\_of\\_SIFT\\_Features\\_for\\_Robust\\_Face\\_Recognition](https://www.researchgate.net/publication/221472612_Adaptation_of_SIFT_Features_for_Robust_Face_Recognition)
4. Face recognition using Gabor waveleth transform [Електронний ресурс] / Burcu Kerenekci – Режим доступу до ресурсу: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.100.6259&rep=rep1&type=pdf>
5. Novel Matching Methods for Automatic Face Recognition using SIFT [Електронний ресурс] / Ladislav Lenc, Pavel Král – Режим доступу до ресурсу: [https://home.zcu.cz/~pkral/papers/kral\\_aiai12.pdf](https://home.zcu.cz/~pkral/papers/kral_aiai12.pdf)
6. Automatic face recognition system based on the SIFT features [Електронний ресурс] / Ladislav Lenc, Pavel Král – Режим доступу до ресурсу: [https://home.zcu.cz/~pkral/papers/cee\\_kral15.pdf](https://home.zcu.cz/~pkral/papers/cee_kral15.pdf)
7. Person-Specific SIFT Features for Face Recognition [Електронний ресурс] / Jun Luo, Yong Ma, Erina Takikawa, Shihong Lao, Masato Kawade, Bao-Liang Lu – Режим доступу до ресурсу: <https://ieeexplore.ieee.org/document/4217478>

## СПОСОБ КОНСТРУИРОВАНИЯ ТРАФИКА В SDN

В работе предложен способ конструирования трафика в SDN, который позволяет упростить процесс управления сетью, за счет разделения уровня управления от уровня передачи данных, значительно повысить использование ресурсов сети и получить равномерно загруженную сеть. В результате работы алгоритма, наблюдаем положительную тенденцию по уменьшению количества потерянных пакетов.

The paper proposes a method for constructing traffic in SDN, which allows us to simplify the network management process, by separating the control level from the data transmission level, to significantly increase the use of network resources and obtain a uniformly loaded network. As a result of the algorithm, we observe a positive trend to reduce the number of lost packets.

**Введение**

Телекоммуникационные сети становятся все более динамичными, а системы управления - сложными и интеллектуальными. SDN (Software Defined Networks) [1] – компьютерная сеть, в которой уровень управления представлен

программным контроллером и является отделенным от уровня передачи данных (рис. 1). Технология SDN обеспечивает высокую гибкость управления и существенно упрощает виртуализацию сетевых ресурсов.

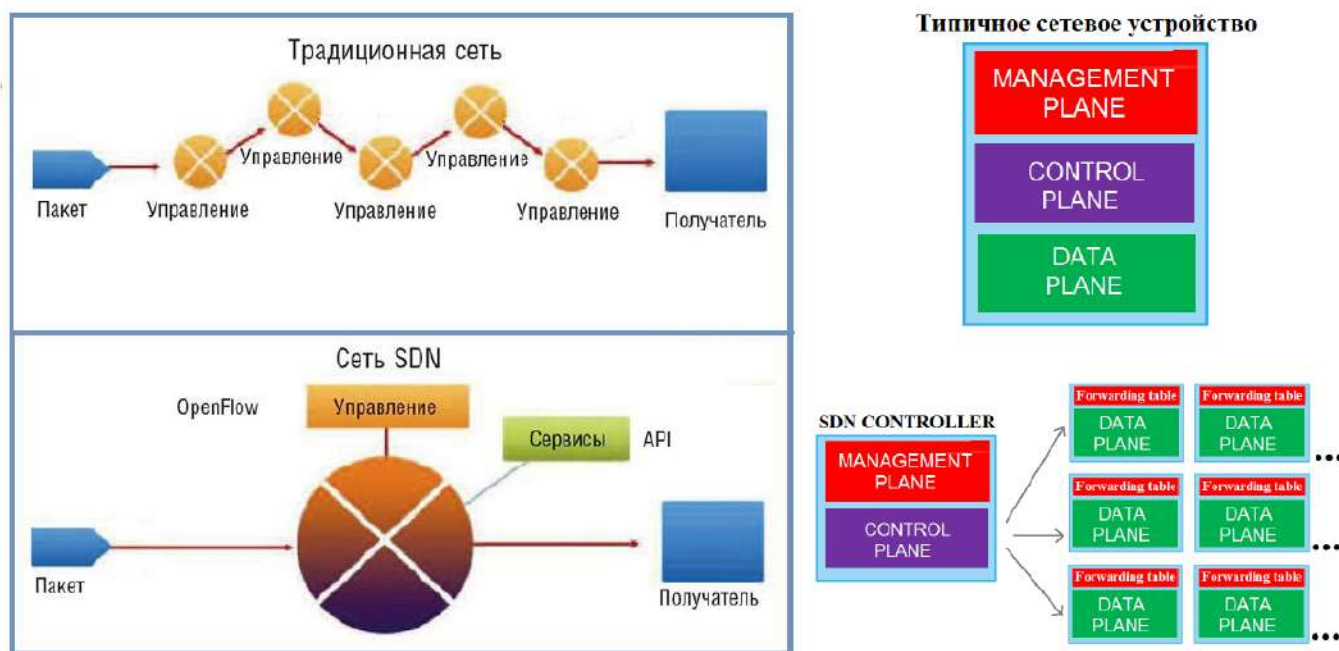


Рис. 1. Традиционная сеть & программно-конфигурируемая сеть

За последнее время, а точнее начиная с 2011-2012 годов, наблюдается лавинообразный рост трафика (рис. 2). Данных становится всё больше, их структура становится всё сложнее. К примеру, видео-трафик растет по экспоненте: с 2010 по 2015 его объем увеличился в 26 раз. А

постоянное увеличение мобильных устройств и популярности социальных сетей привело к тому, что мобильный трафик за этот период вырос в 4000 раз (2005-2015) [2].

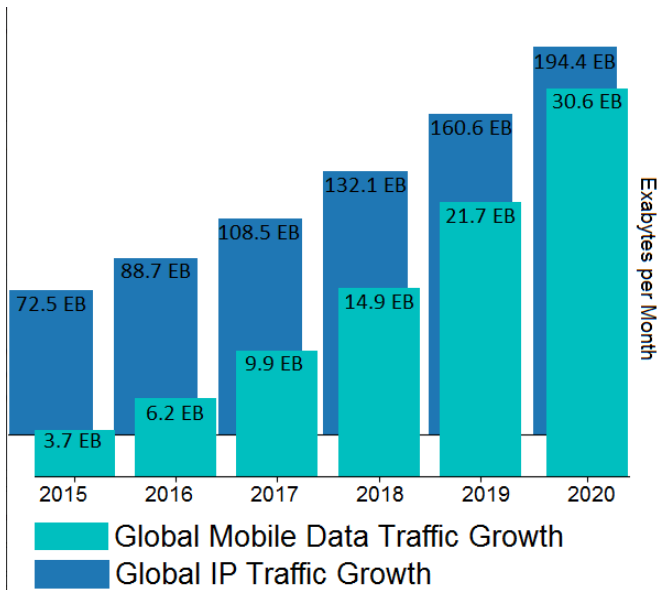


Рис. 2. Рост трафика

Под маршрутизацией в компьютерных сетях понимается процесс определения (выбора) пути следования информации от источника к адресату, удовлетворяющего заданным параметрам *QoS* (*Quality of Service*) и обеспечивающего сбалансированную загрузку компьютерной сети в целом [3]. Маршрутизация является одной из основных функций сетевого уровня и является достаточно сложной задачей, не имеющей однозначного решения для сетей с различной топологией, величиной и характером потока данных.

В общем случае маршрутизация состоит из двух подзадач:

- определение путей и распространение маршрутной информации между ними;
- конструирования трафика.

Основной целью маршрутизации является обеспечение наилучшего пути следования информации с точки зрения ее минимально возможной задержки и максимальной пропускной способности сети [4].

В качестве единицы передаваемой информации на сетевом уровне выступает пакет [3]. Последовательность передаваемых пакетов образует поток информации. Величину передаваемого потока информации за период времени принято называть сетевым трафиком [5].

Понятие *QoS* включает в себя соглашение или гарантии сети о предоставлении определенным типам трафика набора измеримых и требуемых этому трафику

параметров *QoS*, которые обеспечат необходимую пропускную способность, допустимые задержки и процент потерянных пакетов.

Трафик в современных компьютерных сетях можно условно разделить на две группы [6]:

- Эластичный трафик;
- Не эластичный трафик.

Алгоритмы управления трафиком и методы передачи данных остаются неизменными, а значит, вопрос качества обслуживания трафика в соответствии с требованиями пользователей и эффективности использования сетевых ресурсов не теряют актуальности.

### Постановка задачи

Конструирование трафика это один из главных методов оптимизации работы/производительности сети, который динамично анализирует, предсказывает и регулирует поведение данных, передаваемых в сети.

Для традиционных сетей разработано множество техник конструирования трафика, они приспособлены к архитектуре классических сетей и достаточно хорошо в них работают, хотя у них и есть некоторые недостатки такие как: использование однопутевой маршрутизации, пути выбираются без учета текущей загрузки ресурсов сети.

Такие методы конструирования трафика являются недостаточно эффективными для SDN, так как они не учитывают их особенности (к примеру, централизованное управление). Поэтому сейчас активно занимаются разработками техник по конструированию трафика в SDN сетях: одни из предложенных решений Hedera Mahout MiceTrap [7]. Одним из популярных решений, которое используется сейчас является схема распределения трафика ECMP [8], которая поровну делит трафик между несколькими путями. Основным недостатком техник, которые разрабатываются и тех, что уже используются, является отсутствие учета загруженности сети и типа трафика, это нужно учитывать ведь к разному трафику разные требования качества обслуживания (Табл. 1).

Табл. 1. Методы конструирования трафика

	Методы конструирования трафика	Основные недостатки
<b>Традиционные сети</b>	<ul style="list-style-type: none"> <li>- MATE: MPLS Adaptive Traffic Engineering</li> <li>- DEFT: Distributed Exponentially-weighted Flow Splitting</li> <li>- DATE: Distributed Adaptive Traffic Engineering</li> <li>- PEFT: Penalizing Exponential Flow splitting</li> </ul>	<ul style="list-style-type: none"> <li>- однопутевая маршрутизация</li> <li>- не учитывается загруженность сети</li> </ul>
<b>Сети SDN</b>	<ul style="list-style-type: none"> <li>- HEDERA</li> <li>- MAHOUT</li> <li>- MiceTrap</li> <li>- ECMP: Equal-Cost Multi-Path</li> </ul>	<ul style="list-style-type: none"> <li>- тип трафика не учитывается</li> <li>- не учитывается загруженность сети</li> </ul>

### Решение поставленной задачи

Проанализировав существующие способы конструирования трафика, в работе предлагается способ конструирования трафика в SDN с использованием многопутевой маршрутизации, что позволит достичь равномерной загрузки путей, контроля перегрузки и оперативно реагировать на изменения в топологии сети. В алгоритме также учитывается тип трафика: эластичный или не эластичный, что позволяет обеспечивать необходимый уровень качества обслуживания.

Рассмотрим подробно этапы алгоритма.

Представим сеть SDN в виде графа  $G=(V, E)$ . Нам известно: пара узлов, между которыми осуществляется передача данных, существующие маршруты между этими узлами, данные, которые нужно передать.

- 1 На первом шаге контроллер собирает информацию о загруженности путей, надежности и длине.
- 2 Рассчитывается обобщенная метрика для каждого пути по формуле (1), которая учитывает все эти параметры.

$$[(K_1/V_e)+(K_2*D_c)] *r. \quad (1)$$

где  $K_1$   $K_2$  – константы, которые по сути являются весовыми коэффициентами пропускной способности и задержки;

$V_e$  – эффективная пропускная способность которая определяется как произведение полосы пропускания на загруженность канала;

$D_c$  - время задержки,  $r$  надёжность, то есть процент информации успешно передана следующему узлу;

- 3 Далее выбирается лучший путь по метрике для следующего пакета, учитывая тип трафика этого пакета. И пакет отправляется по этому пути.
- 4 Метрики пересчитываются, и следующий пакет отправляется по выбранному для него пути. Аналогично и все последующие пакеты.
- 5 Если же возникает ситуация, когда меняется топология сети, к примеру, вышел из строя узел в одном из маршрутов, то контроллер просчитывает варианты обхода проблемного участка, с помощью теории принятия решений находит лучший из них и перенаправляет трафик на него, и пакеты продолжают передаваться дальше.

Также был разработан программный продукт, с помощью которого было проведено ряд экспериментов и получены следующие результаты (рис 3). Как видно с графика, количество потерянных пакетов значительно меньше, если учитывать тип трафика вместе с загруженностью сети и пропускной способностью канала.

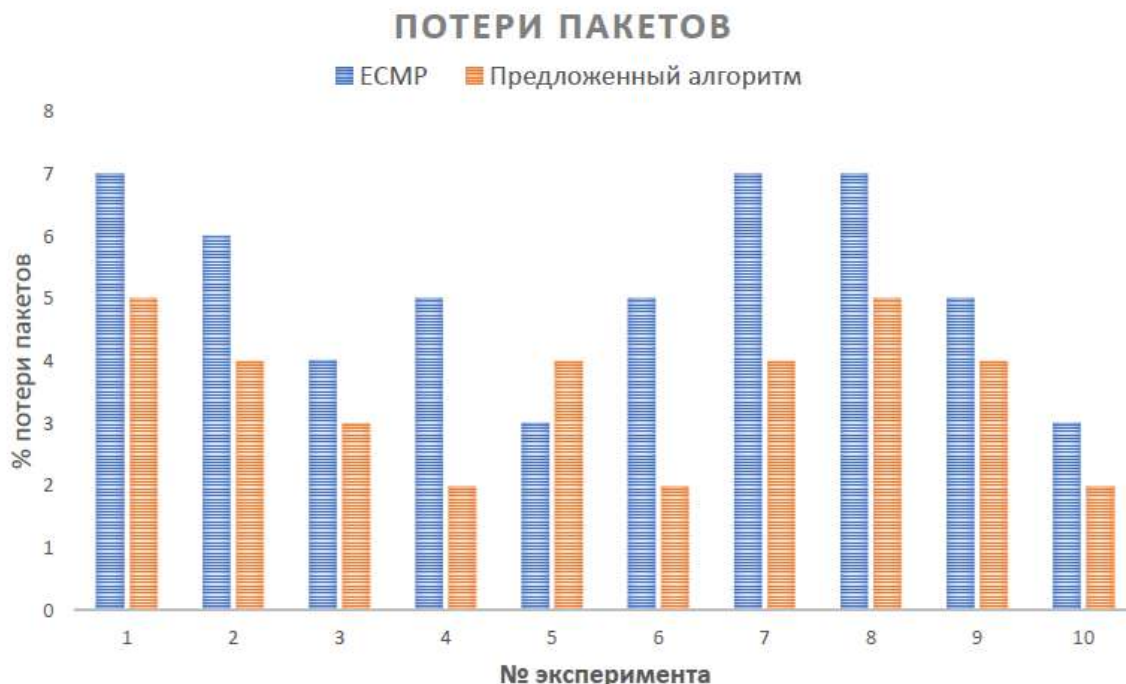


Рис. 3 Сравнительный анализ работы алгоритмов

### Выводы

Существующие методы управления трафиком не обеспечивают дифференцировки отдельных потоков пользователей. Система управления SDN не гарантирует выполнение требований к качеству обслуживания отдельных потоков, поскольку оперирует неточной информацией о состоянии сетевых ресурсов и не способна оптимально выбрать маршрут передачи данных.

В связи с этим в работе предложен алгоритм конструирования трафика в SDN на основе многопутевой маршрутизации, что обеспечивает лучшее качество обслуживания в сравнении с ECMP, потеря пакетов в целом, и в частности пакетов неэластичного трафика меньше, загрузка сети осуществляется более равномерно.

### Список использованной литературы

1. Смелянский Р.Л. Программно-конфигурируемые сети – основа Интернета нового поколения // Сборник материалов 2-й Всероссийской научно-технической конференции «Суперкомпьютерные технологии (СКТ-2012)». – Ростов н/Д: Южный фед. у-т, 2012. – С. 321–329.
2. Cisco Systems, Inc. All rights reserved. – 2009.
3. Таненбаум Э. С. Компьютерные сети. 4-е изд. / Таненбаум Э. С. – СПб. : Питер, 2003. – 992 с. – (Серия «Классика computer science»).
4. Руководство Cisco по междоменной многоадресатной маршрутизации / [Адамс Б., Ченг Э., Фокс Т. и др.] : под ред. Алмерос К.С., Кроукфорд Д. – М. : Вильямс, 2004. – 320 с. – (Серия “Cisco Press”).
5. Кульгин М. В. Технологии корпоративных сетей / Кульгин М. В. – СПб : Питер, 2000. – 704 с. – (Серия “Энциклопедия”).
6. Кульгин М. В. Компьютерные сети. Практика построения / Кульгин М. В. – [2-е изд.]. – СПб : Питер, 2003. – 464 с. – (Серия “Для профессионалов”).
7. Mohammad Al-Fares, Sivasankar Radhakrishnan, Nelson Huang, Amin Vahdat, Barath Raghavan, « Hedera: Dynamic Flow Scheduling for Data Center Networks», NSDI'10 Proceedings of the 7th USENIX conference on Networked systems design and implementation, 2010
8. Andrew R. Curtis, Wonho Kim, Praveen Yalagandula, « Mahout: Low-Overhead Datacenter Traffic Management using End-Host-Based Elephant Detection», IEEE INFOCOM, 2011.

УДК 004.021

*ПЕТРОВСЬКИЙ А.Д.,  
ЦЮЦЮРА М.І.*

## **ДОСЛІДЖЕННЯ НАЙЕФЕКТИВНІШОГО АЛГОРИТМУ ДЛЯ СТИСНЕННЯ ЕЛЕКТРОННИХ ЗОБРАЖЕНЬ ДОКУМЕНТІВ**

В даній статті досліджено алгоритм для стиснення електронних зображень документів. Описано шлях зберігання зображень в електронному вигляді. Алгоритм стиснення розглянуто послідовно по кожному пункту та з приведенням прикладів. Також приведено результат стиснення електронного документу за даним алгоритмом і описано кількість внутрішньої пам'яті яка необхідна для зберігання стисненого документу.

### **КЛЮЧОВІ СЛОВА**

Алгоритм, стиснення, електронний документ, піксель, цифрове зображення, модель ARGB, біт, байт, формат, оптимізація, оптимізація пікселів, кодування зображення, документ.

In this paper an algorithm for compressing electronic documents of documents was investigating. The way of storing images in electronic form is described. The compression algorithm is considered sequentially for each point and with examples. Also, the result of compression of the electronic document according to this algorithm is given and the amount of internal memory necessary for storing the compressed document is described.

### **KEYWORDS**

Algorithm, compression, electronic document, pixel, digital image, ARGB model, bit, byte, format, optimization, pixel optimization, image encoding, document.

### **1. Вступ**

Стиснення зображень – це зменшення розміру цифрових документів, за допомогою алгоритмів стиснення файлів, які застосовуються до зображення. Це робиться для того щоб зменшити місце яке вони займають при зберіганні а також для зменшення часу, який потрібен для передачі зображень по мережі. Не слід плутати алгоритми для стиснення файлів з форматом. Це зовсім різні речі. Алгоритм – це набір інструкцій, який в скінченне число кроків приводить до перетворення з початкового коду цифрового зображення в код меншого об'єму із ліквідацією збитковості. Звісно існує досить багато алгоритмів для стиснення цифрових зображень, але деякі

з них не завжди перетворюють початковий код у менший кінцевий. Це залежить від багатьох факторів таких як тип зображення, його розмір та якість. Тому було досліджено найефективніший алгоритм для стиснення цифрових зображень документів.

### **2. Огляд рішень**

Існує два види стиснення документів – із втратами та без. Алгоритми стиснення із втратами більш продуктивні, але документи, які вони стискають, можуть втратити якість зображення. Алгоритми без втрат навпаки, зберігають початкову якість зображень, але не такі продуктивні.



Розглянемо найпростіший спосіб зберігання електронного зображення.

Якщо приблизити будь-яке цифрове зображення, ми зможемо побачити що воно складається із клітинок, які мають свій колір.

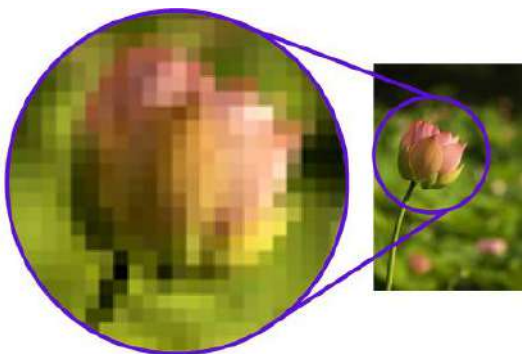


Рис.1 – Вигляд цифрового зображення

Ці клітинки – це пікселі, які мають свій колір. Чим більша густина пікселів, тим краще зображення. Колір кожного пікселя в основному формується за допомогою моделі ARGB. ARGB – розшифровується як Alpha, Red, Green, Blue. Це основні кольори для формування кольору пікселя. Кожен колір зберігається в одному байту інформації. Тобто для зберігання в пам'яті одного пікселю використовується чотири байти.

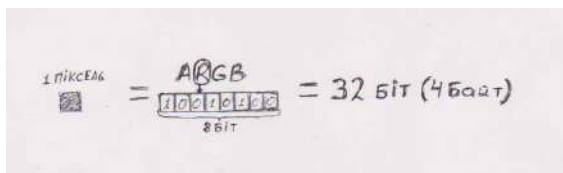


Рис.2 - Зберігання одного пікселю.

### 3. Запропоноване рішення

Так як в основному документи мають чорно-білий формат зображення, можна зберігати його в чорно-білому електронному вигляді. Щоб перевести один кольоровий піксель у відтінок сірого необхідно знайти середнє арифметичне кожного з кольорів R,G,B та присвоїти отримане значення кожному з них. Прозорість A можна опустити.

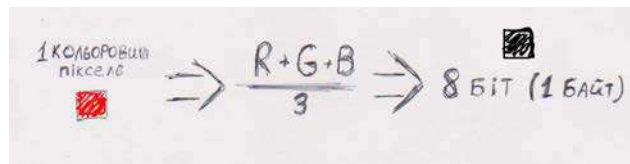


Рис. 3 – переведення кольорового пікселя у відтінок сірого.

Таким чином, щоб зберегти один піксель відтінка сірого в пам'яті займається всього 1 байт. Це значно зменшує розмір файлу.

Звичайно не усі документи є чорно-білими. В деяких з них є кольорова печать або підпис. Для оптимального зберігання таких документів, алгоритм виділяє кольорову зону та залишає її без змін.

Після переводу зображення в чорно-білий формат, здійснюється оптимізація зображення. Алгоритм оптимізації досить простий – спочатку ми беремо відтінок кольору пікселю (від 0 до 255) а також сусідніх пікселів і зводимо їх до ближнього значення. Наприклад, у електронному зображенні документу білий фон, але відтінок пікселів може трохи відрізнитись. Для ока це не помітно, але код відтінку у пікселів різний. Тому ми приводимо ці пікселі до єдиного коду.

### 4. Результати дослідження

На рисунку 4 зображено послідовність перетворення електронного зображення.



Рис. 4 – послідовність алгоритму оптимізації зображення.

Як ми можемо побачити, початкове зображення та його фінальна версія яка пройшла етап виділення кольорової області, переведення у відтінок сірого і оптимізації пікселів зовсім не відрізняються для людського зору, але після стиснення, його розмір значно зменшується.

Останнім кроком даного алгоритму є кодування зображення. Щоб не записувати код кожного пікселю який займає місце в 1 байт, ми знаходимо послідовності однакових бітів і записуємо у форматі «кількість + колір». Таким чином якщо є послідовність 30 пікселів однакового кольору, нам необхідно лише 2 байти замість 30. Даний алгоритм не використовує запис пікселів у послідовність за допомогою проходження масиву пікселів зигзагом як це відбувається у форматі JPEG який ми можемо побачити на рисунку 5.

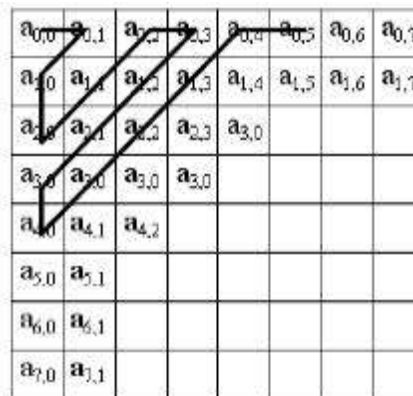


Рис. 5 – Проходження пікселів зображення в алгоритмі JPEG.

Це створено для того, щоб якомога більше пікселів одного кольору були один біля одного. Але цей обхід зигзагом втрачає свою ефективність при обходженні зображення документу, тому що в більшості документів саме горизонтальні лінії пікселів мають схожий відтінок пікселів. Тому було обрано горизонтальний обхід.

## 5. Висновки

Отже ми дослідили найефективніший алгоритм для стиснення електронних зображень документів. Розглянули кожен пункт алгоритму, починаючи переводом зображення у чорно-білий формат, закінчуючи оптимізацією пікселів. В результаті ми привели приклад документу який був стиснено даним алгоритмом. Побачили що для людського ока різниця в документах майже не помітна, але в результаті документ став займати менше внутрішньої пам'яті що дозволить швидше передавати його по мережі.

## СПИСОК ЛІТЕРАТУРИ

1. Дж. Міано. Формати і алгоритми стиснення зображень в дії.—М.: Вид. триумф, 2003.— 336с.
2. Д. Селомон. Стиснення даних, зображень и звуку. — М.: Техносфера, 2004. — 368 с.
3. Д. С. Ватолін. Алгоритми сжатия изображений. Методическое пособие. — Издательский отдел факультета Вычислительной Математики и Кибернетики МГУ имени М. В. Ломоносова, 1999. — 76 с.
4. Ватолін Д., Ракушняк А., Смирнов М., Юкин В. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. — М.: ДИАЛОГ-МИФИ, 2003. — 384 с.

УДК 681.3.07

СКИДАН Д. О.  
ЖДАНОВА О. Г.

## КОМПЛЕКСНИЙ АСИМЕТРИЧНИЙ АЛГОРИТМ ШИФРУВАННЯ З ДИНАМІЧНИМ КЛЮЧЕМ

В статті розглядається комплексний асиметричний алгоритм шифрування з динамічним ключем, в основі якого лежить використання декількох ключів різного типу: асинхронні, синхронні та динамічні. Для динамічного ключа розроблений алгоритм з механізмом зворотного зв'язку. Проаналізовані властивості синхронних та асинхронних алгоритмів, обґрунтовано вибір алгоритмів для використання в комплексі з розробленим алгоритмом, що використовує динамічний ключ. Створений гібридний алгоритм дозволяє забезпечити комплексний захист персональної інформації.

АЛГОРИТМ ШИФРУВАННЯ, ДИНАМІЧНИЙ КЛЮЧ, СИМЕТРИЧНИЙ АЛГОРИТМ ШИФРУВАННЯ, АСИМЕТРИЧНИЙ АЛГОРИТМ ШИФРУВАННЯ, ГІБРИДНИЙ АЛГОРИТМ ШИФРУВАННЯ.

The complex asymmetric encryption algorithm with a dynamic key is considered in the article, on the basis of using several keys with different types, such as asynchronous, synchronous and dynamic. For dynamic key, an algorithm with a feedback mechanism is developed. Properties of synchronous and asynchronous algorithms are analyzed and justified a choice of algorithm for using in complex with the developed algorithm, which uses a dynamic key. The developed hybrid algorithm provides comprehensive protection of personal information.

ENCRYPTION ALGORITHM, DYNAMIC KEY, SYMMETRIC ENCRYPTION ALGORITHM, ASYMMETRIC ENCRYPTION ALGORITHM, HYBRID ENCRYPTION ALGORITHM.

### 1. Вступ

З кожним днем зростає кількість крадіжок та інцидентів, пов'язаних з даними. Кібератаки продовжують розвиватися, хакерами знаходяться нові витончені методи враження даних інтернет-користувачів. Щоб протидіяти таким спробам захист інформації став невіддільною мірою у сучасному світі кібербезпеки. Шифрування - це один зі способів захисту дискретної інформації, що передається в Інтернеті.

На сьогодні створено багато складних алгоритмів для шифрування чутливої до ураження інформації (тобто переводу її в незрозумілий формат). Оскільки найбільш вразливою частиною системи є передача даних, то однією з основних задач є знаходження комплексного підходу до захисту каналів передачі.

Зашифровані дані можуть бути розшифровані лише за допомогою належних функцій, відомих як "криптографічні ключі". В основному криптографічний ключ - це пароль, який використовується для шифрування та розшифрування інформації. Є два типи криптографічних ключів, відомі як симетричні (секретні) та асиметричні, вони відповідають

алгоритмам шифрування, а саме симетричному та асиметричному. Асиметричний алгоритм більш захищений шляхом створення пари ключів, але і він має вразливі місця. Один зі шляхів усунення цих вразливих місць є використання динамічного ключа, тобто ключа, який змінюється на кожній ітерації.

Виходячи з цього виникає ціла низка задач, які необхідно вирішити. По-перше, треба проаналізувати існуючі синхронні та асинхронні алгоритми та обрати ті, які за певними характеристиками найкраще підходять до роботи в комплексі. По-друге, розробити алгоритм динамічної зміни ключа. По-третє, об'єднати алгоритми в один гібридний алгоритм для комплексного захисту каналу.

### 2. Порівняльний аналіз асиметричних алгоритмів

Основними асиметричними алгоритмами виступають RSA, Diffie-Hellman, DSA, ECC.

RSA є одним з методів шифрування відкритих ключів. Це був перший алгоритм, розроблений у криптографії з відкритим ключем, і одне з перших великих досягнень у шифруванні відкритих ключів. Він містить в

собі генерацію ключів, шифрування та дешифрування. В процесі генеруються два ключі секретний та публічний для обміну та шифрування.

Алгоритм Deffie-Hellman дозволяє обом сторонам, які попередньо не знають один про одного, взаємно встановлювати загальний секретний ключ в небезпечному каналі. Обмін ключами Діффі-Хеллмана оснований на симетричній криптографії, оскільки загальний секретний ключ і ключ сеансу використовуються для шифрування та дешифрування. Алгоритм використовується багатьма протоколами, такими як SSL, Secure Shell та IPSec [1].

Цифрові підписи є одним з кращих інструментів для забезпечення безпеки. Цифровий підпис - це електронна версія письмового підпису. Це криптографічний алгоритм відкритого ключа, який забезпечує автентифікацію та авторизацію. Цифровий

підпис реалізується за допомогою алгоритмів публічного та приватного ключів та хеш-функцій. Він використовується на стороні приймача інформації для підтвердження повідомлення та ідентифікації відправника.

Алгоритм ECC в основному залежить від алгебраїчної структури еліптичних кривих. ECC має три етапи експлуатації: ключове узгодження, шифрування та алгоритми цифрового підпису. ECC вважається кращим для створення більш швидких, і менших ефективних ключів. ECC пропонує еквівалентну кількість безпеки для набагато менших розмірів ключа, а отже, зменшує витрати на обробку та комунікацію. ECC вважається найбільш підходящим для сенсорних мереж, що забезпечує гарний компроміс між ключовими розмірами та безпекою [2].

В таблиці 1 наведені основні характеристики аналізованих алгоритмів:

**Табл. 1 Порівняльна таблиця асиметричних алгоритмів**

<i>Алгоритм</i>	<i>Розмір ключа</i>	<i>Переваги</i>	<i>Недоліки</i>	<i>Можливі атаки</i>	<i>Контрзаходи для атак</i>
1	2	3	4	5	6
RSA	1024, 2048, 3072, 4096	Менший обчислювальний час	Малий показник шифрування та мале повідомлення. Той самий ключ для шифрування та підписання. Використання спільного модуля для різних користувачів	Атака Хастада, Франкліна-Рейтера, атака по частковій експоненті	Оптимальне асиметричне шифроване доповнення
Deffie-Hellman	1024, 3072	Вирішує складний дискретний логарифм. Створення і розповсюдження ключа, не інформації	Дорога експоненціальна операція. Відсутність автентифікації.	Атака Головний по центру	Використання алгоритму автентифікації з алгоритмом D-H
DSA	від 512 до 1024 (включно)	Автентифікація Цілісність даних Невідхильність	Ентропія, секретність та унікальність випадкового значення є критичними.	Атака відновлення ключа	-

1	2	3	4	5	6
ECC	160, 224,56	Менший розмір ключа. Зниження рівня пам'яті. Зменшення часу передачі. В 15 разів швидше, ніж RSA. Менша споживана потужність	Збільшується розмір зашифрованого тексту. Залежить від дуже складних рівнянь, що підвищують складність алгоритму	Атаки стороннього каналу Backdoors. Квантові обчислювальні атаки	

Навантаження алгоритмів експериментами не має сенсу, оскільки алгоритми досить різні і мають різну сферу застосування. Алгоритм Diffie-Hellman побудований на симетричній базі, тому в гібридному використанні він не доцільний. Алгоритм DSA має застосовуватись більше для шифрування файлів і цифрового підпису. Можна розглядати алгоритми ECC та RSA. По швидкодії алгоритм ECC має більше переваг, але розмір ключа занижкий. Тому для шифрування будемо використовувати RSA алгоритм.

### 3. Порівняльний аналіз симетричних алгоритмів

В процесі аналізу будуть розглядаються алгоритми: DES, 3DES, AES, Blowfish, DES

DES не виконує вимогу RFC, використовуючи лише 56-бітні клавіші, що дає 256 можливих ключів. Цей відносно невеликий розмір ключа робить його вразливим до атак і багато разів було зламано, наприклад, за допомогою кластерного комп'ютера FPGA SciEngines FPGA, використовуючи 128 Spartan-3 5000 FPGA. Використовуючи пристрій грубої сили з швидкістю вгадування в тисячі мільярдів кнопок на секунду 1012

клавіш / сек, для пошуку ключа потрібно менш як день. Тому DES не вважається захищеним алгоритмом і його було видалено з TLS 1.2.

Алгоритм 3DES використовує два або три ключі DES, які є 56-бітними та забезпечують безпеку від 56 біт до 168 біт залежно від параметра ключа.

Ключ змінної довжини для Blowfish може коливатися від 32 до 448 біт, за замовчуванням він має 128 біт, використовуючи 64-бітний блоковий шифр. Найменший захист, який може бути наданий, - це 232 можливих ключів до найвищої безпеки 2448 можливих ключів з періодом грубого застосування від кількох хвилин до 10115 років [3].

AES має розмір ключів 128 біт, 192-бітних і 256-бітових, він здатний виробляти від 2128 до 2256 можливих ключів. Використовуючи найменшу безпеку AES, тобто 128-бітовий розмір ключа, знадобиться 1019 років, щоб знайти потрібний ключ, використовуючи швидкість 1012 ключів / сек. AES ще не був зламаний.

Над алгоритмами були проведені тести для оцінки їх за основними параметрами. Результати порівняльного аналізу представлені в таблиці 2 [4].

**Табл. 2 Порівняльна таблиця симетричних алгоритмів**

Параметр	DES	3DES	AES	Blowfish
1	2	3	4	5
Розмір ключа	56	56, 112 або 168	128, 196 або 256	32-448
Тип	Блочний шифр	Блочний шифр	Блочний шифр	Блочний шифр
Розміри блоків	64	64	128	64
Число раундів	16	48	10,12 або 14	16
Рік випуску	1975	1998	1998	1993

1	2	3	4	5
Brute Force Атака	Менш як день	Менш як день до $\approx 10^{34}$ років	$10^{19}$ років до $\approx 10^{58}$ років	Пару хвилин до $\approx 10^{115}$ років
Більш ефективні атаки	Диференційований криптоаналіз	Meet in the middle атака	Паралельний канал	Слабий ключ
Зламаний	Так	Ні	Ні	Ні
Статус безпеки	Небезпечний	Частково	Безпечний	Безпечний
Швидкість шифрування	Швидкий	Повільний	Найшвидший	Повільний
Ініціалізація ключа	Середня	Повільна	Швидка	Найповільніша
Простір складності	Чудовий	Добрий	Добрий	Поганий

Виходячи з даних таблиці 2, можна зробити висновок, що доцільніше використовувати алгоритм AES, оскільки він має найкращу швидкість при досить великому розмірі ключа. Також простір складності є добрим при найшвидшій ініціалізації ключа. Варто відзначити, що AES добре документований, що дозволяє подальшу оптимізацію алгоритму. AES був обраний як стандарт шифрування. Сьогодні існує багато систем, що мають апаратне прискорення для AES, це є великою перевагою, бо може суттєво зменшити час виконання алгоритму [2].

#### 4. Розробка алгоритму динамічної зміни базового ключа

Для рішення задачі впровадження динамічного ключа було вирішено розробити алгоритм з механізмом зворотного зв'язку. Кожен блок відкритого тексту або частини

(крім першого) побітово складається по модулю 2 (операція XOR) з попереднім результатом дешифрування. Таким чином дані будуть зашифровані по різному на кожній ітерації, а для дешифрування потрібно мати відкритий текст з попередньої ітерації.

Процес шифрування проілюстровано на рисунку 1 та може бути описано наступним чином:

$$C_0 = V \quad (1)$$

$$C_i = E_k(P_i \oplus P_{i-1}) \quad (2)$$

де  $i$  - номер блоку,  $V$  - вектор ініціалізації,  $C_i$  і  $P_i$  - блоки зашифрованого і відкритого текстів відповідно, а  $E_k$  - функція блочного шифрування.

Процес розшифрування продемонстрований на рисунку 2, в його основу покладено така операція:

$$P_i = C_{i-1} \oplus D_k(C_i) \quad (3)$$

де  $D_k$ , функція блочного розшифрування.

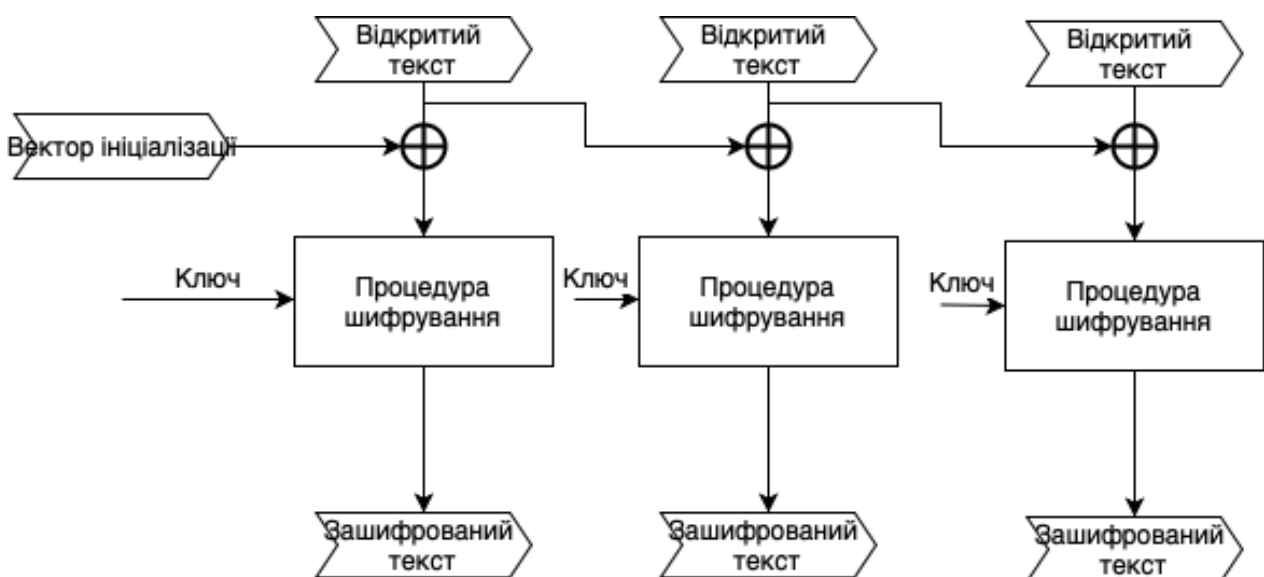
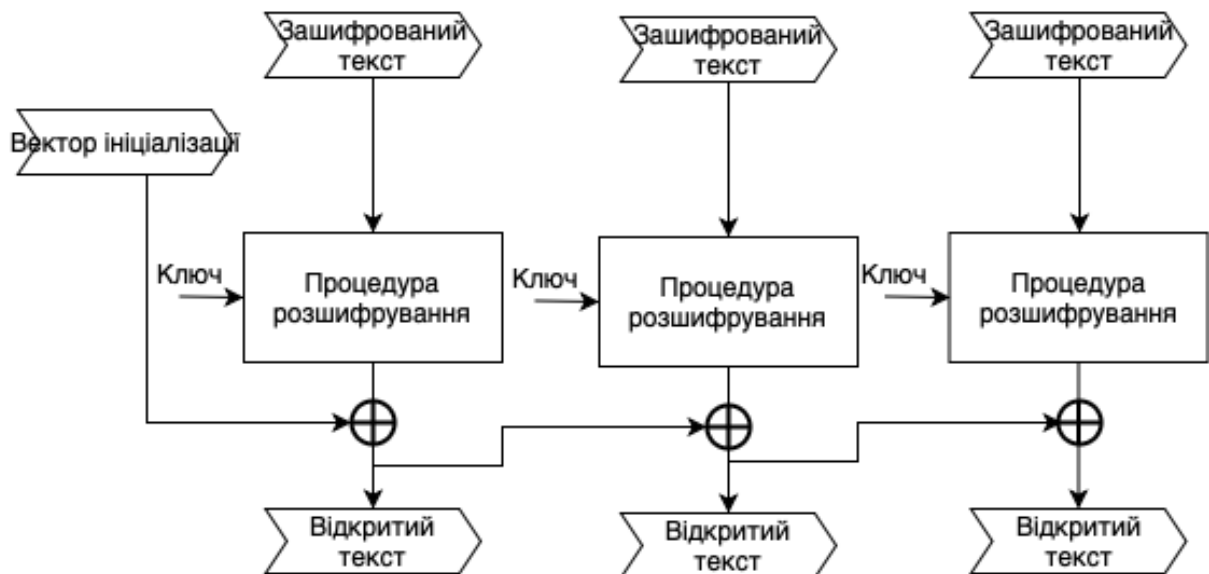


Рисунок 1. Алгоритм шифрування



*Рисунок 2. Алгоритм дешифрування*

Таким чином створюється деякий динамічний ключ, який передається в середині історії блоків. Швидкість операції XOR є настільки малою, що нею можна нехтувати. Для підвищення захисту операцію XOR можна замінити більш складною функцією, але це тягне за собою і збільшення часу операції.

#### **5. Розробка комплексного алгоритму з динамічним ключем**

Комплексний алгоритм базується на гібридному поєднанні симетричного, асиметричного алгоритмі та алгоритму з механізмом зворотного зв'язку. В якості симетричного алгоритму використовується алгоритм AES, в якості асиметричного – RSA.

##### **Схема алгоритму шифрування:**

1. Відкритий текст побітово складається по

модулю 2 з попереднім відкритим текстом (в першій ітерації – з вектором ініціалізації).

2. Генерується новий ключ для шифрування AES і ним шифрується складений текст.

3. Ключ вирівнюється до 256 біт і кладеться на початок зашифрованих даних.

4. Ключ разом з шифрованим текстом шифрується публічним ключем RSA.

##### **Схема алгоритму дешифрування:**

1. Зашифровані дані розшифровуються закритим ключем RSA.

2. З початку розшифрованих даних вилучається 256 біт – ключ для AES.

3. Данні без перших 256 біт розшифровуються AES алгоритмом.

4. Розшифровані дані складаються по модулю 2 з попереднім відкритим текстом (для першої ітерації з вектором ініціалізації)

#### **Висновок**

В цій статі вирішена задача розробки комплексного асиметричного алгоритму з динамічним ключем. Були проаналізовані існуючі симетричні та асиметричні алгоритми та обґрунтовано вибір тих, що якнайкраще підходять до використання в комплексі. Вирішення поставленої задачі було досягнене шляхом розробки гібридного алгоритму, що базується на обраних симетричних та асиметричних підходах, та використанні динамічного ключа. Задача введення динамічного ключа була вирішена шляхом створення алгоритму зі змінним ключем в залежності від даних. Була наведена схема та описана робота повного комплексного алгоритму. Використання цього алгоритму дозволить підвищити складність зламу та захищеність каналу, при передачі даних.

#### **Перелік посилань**

1. Johannes A. "Introduction to Cryptography", 2nd ed. / Johannes A. – Springer, 2016. – P. 71-137
2. Nadeem "A Performance Comparison of Data Encryption Algorithms" / Nadeem – 2015. – P. 7-87
3. Bingmann T. "Speedtest and Comparison of Cryptography Libs" [Електронний ресурс] / Bingmann T. – 2008. – Режим доступу: <https://panthema.net/2008/0714-cryptography-speedtest-comparison/>
4. Rogaway P. "Evaluation of Some Blockcipher Modes of Operation" / Rogaway P. – 2017. – P. 15-53

УДК 004.023

*ДМИТРИШИН С. А.***ПІДХІД ДО УПРАВЛІННЯ ЛОГАМИ КОМП'ЮТЕРНОЇ ІНФРАСТРУКТУРИ**

Швидка поява нових загроз стимулює до постійного пошуку нових рішень, однак скорочення бюджету ІТ компаній ставить під загрозу цільність систем управління логами. Зважаючи на проблеми, які виникли наразі в сфері управління логами, в статті будуть проаналізовані наступні теми: збір відповідних даних, доступне використання логів і регулярний аналіз логів.

**ІНФРАСТРУКТУРА, ЛОГИ, ЗАГРОЗИ, АВТОМАТИЗОВАНІ СИСТЕМИ, БЕЗПЕКА, АНАЛІЗ ЛОГІВ**

The rapid emergence of new threats prompts continued search for new solutions, but budget cuts in IT companies jeopardize the integrity of logging management systems. Due to the problems that have arisen in the field of logging management, the article will analyze the following topics: collecting relevant data, using logs and regularly analyzing logs.

**INFRASTRUCTURE, LOG, THREATS, AUTOMATED SYSTEMS, SECURITY, LOG ANALYSIS**

**Вступ**

Більшості організаціями на сьогодні загрожує велика кількість небезпек. Ці загрози змінюються і збільшуються експоненціально. Спеціалісти в області безпеки стикаються зі значними штрафами від державних та галузевих правил, якщо системи не відповідають мінімальним вимогам безпеки. Тим не менше, незважаючи на ці нові проблеми, інвестування в безпеку скорочується, адже керівники ІТ компаній надають перевагу інвестуванню в операційну діяльність компаній, яка приносить прибуток.

На жаль, скорочення ІТ-бюджетів є доволі великою проблемою, оскільки еволюція загроз вимагає розширення систем безпеки, а не їх скорочення. Наприклад, налаштування систем, зокрема управління логами в минулому було доволі простим процесом, але зважаючи на сучасні мережі і нормативні вимоги потрібні все складніші налаштування систем. Управління логами тепер виходить за

рамки простого збору даних; адже включає в себе нормування, аналіз, звітність та зберігання критичної інформації про можливі загрози чи внутрішні помилки систем.

**Типи логів**

Для того, щоб відповідати нормам безпеки, потрібно збирати всі логи (журнали роботи), а не лише ті, які мають причетність до безпеки. Наприклад, логи операційної системи та логи додатків часто містять інформацію, пов'язану з безпекою. Те ж саме можна сказати і про події, які спочатку можуть видаватися не пов'язаними з безпекою. Організації повинні враховувати потенційну цінність кожного логу.

Існують наступні типи логів:

- антивірусне програмне забезпечення;

Антивірусні програми включають в себе: програми, які виявляють загрози у вигляді шкідливого програмного забезпечення, вірусів, перехоплювання і



передачу даних. Логи антивірусних програм включають в себе інформацію про виявлення загроз, про очищення або переміщення на карантин, про останні рази, коли системи сканували на наявність загроз.

- застосунки;

Інформація, зареєстрована різними програмами, може суттєво відрізнитись і може включати зміни в обліковому записі, спроби автентифікації, використання привілеїв, відомості про використання, активність клієнта та сервера, зміни в конфігурації, серйозні збої системи та багато іншого. Однак логи можуть бути в специфічному для цих застосунків форматі.

- сервери автентифікації;

Сервери каталогів та сервери єдиного входу зазвичай записують кожну спробу автентифікації, яка відображає вихідний ідентифікатор користувача, систему призначення або додаток, інформацію про дату та час, а також відомості про успіх/нездатність.

- файерволи;

Деякі брандмауери є орієнтованими на периметр і загальними за своїм характером, тоді як інші - дуже специфічні для застосування. Брандмауерами можуть не тільки блокувати діяльність на основі політики, але також можуть перевіряти вміст і забезпечувати стан та цілісність дозволених з'єднань. Брандмауери мають численні можливості, і їх логи можуть бути дуже докладними та інформативними.

- системи попередження загроз та протидії загрозам

Ці системи записують докладну інформацію про підозрілу поведінку та виявлені атаки, а також про дії, спрямовані на припинення шкідливої активності. Деякі системи захисту від

вторгнень, такі як системи цілісності файлів, запускаються періодично, а не безперервно, тому вони генерують логи партіями, а не постійно.

- сервери контролю доступу в мережу

Контроль доступу до мережі може працювати як для внутрішніх, так і для зовнішніх хостів, що підключаються до внутрішньої мережі. Під час підключення визначається стан безпеки хостів, а хости, які не дотримуються визначеної політики, виділяються на окремий сегмент VLAN (Virtual Local Area Network). Сервери NAC містять велику кількість корисної інформації про успішні / дозволені та невдалі мережеві з'єднання.

- мережеві пристрої (маршрутизатори, свічі);

Маршрутизатори можуть бути налаштовані на блокування певних типів трафіку. Мережеві пристрої можуть бути налаштовані записувати дуже детальні логи, але зазвичай записують їх на мінімальному рівні. Ці логи можуть містити детальну інформацію про мережеву комунікаційну діяльність.

- операційні системи;

Є багато різноманітних операційних систем на серверах, робочих станціях та різноманітних мережевих пристроях. Адміністратор хосту зазвичай контролює ведення логів. Ці логи зазвичай містять інформацію про запуск і зупинку сервісів, спроби автентифікації, звернення до файлів, зміни політики безпеки, зміни облікових записів, дозволи та привілеї, а також використання привілеїв.

Журнали операційної системи корисні для виявлення підозрілих дій, пов'язаних із певним хостом.

- програмне забезпечення для забезпечення доступу;

Віртуальні приватні мережі (VPN) - це найпопулярніший тип захищених рішень віддаленого доступу, і вони включають як успішні, так і невдалі спроби з'єднання. Вони записують такі дані, як дата та час, коли кожен користувач під'єднується та від'єднується, а також типи та кількість даних, надісланих та отриманих під час підключеного сеансу.

-веб проксі;

Веб-проксі сервери є проміжними хостами, через які доступні веб-сайти, і вони можуть бути використані для обмеження доступу до Інтернету, а також додати шар захисту між користувачем та зовнішніми веб-сайтами. Журнали веб-проксі записують активність користувачів та URL-адреси, до яких звертаються вказані користувачі.

Кожен тип логів містить різноманітну інформацію, і ця інформація буде в різних форматах. Залежно від обставин, різні джерела журналів можуть бути більш-менш цінними. Слід також зазначити, що якщо адміністративні привілеї не належним чином обслуговуються і журнали захищені, то журнали можуть бути змінені або змінені. Важливо розуміти та обмежувати ці привілеї та доступ до записаних даних.

### **Підхід до управління логами**

Багато відповідних мандатів, таких як PCI DSS 3.1, вимагають не лише того, щоб компанії збирали всі логи, але і щоб вони регулярно переглядалися, були доступними для пошуку і зберігалися в оригінальній, незмінній, необробленій формі для певних термінів виконання мандату.

Логи також можуть бути надзвичайно корисними для визначення порушень політики, шахрайської діяльності та операційних завдань незабаром після їх виникнення. Вони надзвичайно корисні

при проведенні перевірок, криміналістичному аналізі, внутрішньому розслідуванні, встановленні базових показників та виявленні операційних тенденцій та довготермінових проблем. Проте нескінченна різноманітність форматів даних логів робить неможливим використання даних без нормалізації даних.

Розумно припустити, що різноманітність джерел даних логів та обсяг даних завжди зростатимуть. Інфраструктура в постійному стані змін, з новими системами, додатками, користувачами та пристроями кожного дня року.

Це створює різні специфічні завдання для управління журналами. Ці виклики можна розділити на три напрямки: збір, аналіз, а також зберігання.

Збирання логів пов'язане з наступними проблемами:

- Дані логів змінюються.
- Розподілені джерела логів. Джерела даних можуть бути розташовані в межах внутрішньої локальної інфраструктури, розташовані в центрі обробки даних, розміщені в керованому хостинг-провайдері.
- Джерела даних логів постійно змінюються.
- Дані журналу можуть містити конфіденційну інформацію, наприклад витяги з електронної пошти, імен користувача та паролі.

Аналіз та перевірка логів представляє собою дві важливі проблеми: регулярний огляд даних та різні формати логів.

Кожен запис журналу або повідомлення містить певні певні фрагменти інформації, такі як IP-адреса хосту або ім'я користувача.

Кожне джерело логів записує ті відомості, які вважаються важливими; отже, може бути надзвичайно важко пов'язати різні джерела логів, оскільки вони можуть містити або не містити загальних значень.

Навіть коли два джерела записують однакові значення, вони можуть бути записані неоднаковим способом. І навіть коли вони записують однакові значення, вони можуть представляти їх по-різному.

Одним із підходів до вирішення цих проблем є створення сценаріїв PERL для пошуку та створення лише тих логів, які відповідають запиту.

Дані логів повинні розглядатися як будь-які інші організаційні дані, за умови дотримання політики безпеки. Оскільки в них часто містяться конфіденційні дані (наприклад, дані клієнта), порушення конфіденційності є серйозною проблемою.

Доступ до журнальних даних повинен суворо контролюватися, і за жодних обставин дані журналу не можуть бути змінені.

Крім того, зберігання логів з різних джерел централізовано створює величезну проблему керування сховищем. Для придбання та розгортання необхідного сховища необхідна велика кількість ресурсів.

### Висновки

Оскільки проблеми з керуванням логами зросли, популярними стали автоматизовані рішення для керування логами. Вони забезпечують збирання логів в межах всієї організації, аналізують дані, надають інструменти для звітності, забезпечують надійний та регулярний огляд логів.

### Список літератури

1. Kent K., Souppaya M. Guide to Computer Security Log Management // Recommendations of the National Institute of Standards and Technology [Electronic resource] - Access: <https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-92.pdf>
2. Chuvakin A., The Complete Guide to Log and Event Management [Electronic resource] - Access: [https://www.novell.com/docrep/documents/9x1wixnqhd/Log\\_Event\\_Mgmt\\_WP\\_DrAntonChuvakin\\_March2010\\_Single\\_en.pdf](https://www.novell.com/docrep/documents/9x1wixnqhd/Log_Event_Mgmt_WP_DrAntonChuvakin_March2010_Single_en.pdf)
3. Guide for Security Event Logging [Electronic resource] - Access: <http://www.usmd.edu/usm/adminfinance/itcc/secevent.pdf>
4. System Logging and Log Analysis [Electronic resource] - Access: [http://ranum.com/security/computer\\_security/archives/logging-notes.pdf](http://ranum.com/security/computer_security/archives/logging-notes.pdf)
5. Data Analysis and Reporting using Different Log Management Tools [Electronic resource] - Access: <https://ijcsmc.com/docs/papers/July2015/V4I7201553.pdf>

## **РОЗПІЗНАВАННЯ АВТОМОБІЛЬНИХ НОМЕРІВ У РЕЖИМІ РЕАЛЬНОГО ЧАСУ НА ОСНОВІ YOLO-ДЕТЕКТУВАННЯ**

У даній статті розглянута технологія розпізнавання автомобільних номерів у режимі реального часу в умовах руху. Автоматичне розпізнавання автомобільних номерів - доволі розповсюджена тема досліджень через важливість практичного застосування. Однак, варто зазначити, що більшість наявних рішень у сфері розпізнавання автомобільних номерів не дають якісних результатів за вдалих випробувань за життєвих ситуацій через велику кількість обмежень. Представлена технологія розпізнавання автомобільних номерів має за основу технологію знаходження необхідних областей YOLO та конволюційні нейронні мережі (CNN).

In this article the technology of recognition of automobile numbers in real time under conditions of movement is considered. Automatic recognition of automobile numbers is a rather widespread topic of research because of the importance of practical implantation. However, it should be noted that most of the available solutions in the area of recognition of automobile numbers do not provide qualitative results for testing in life situations due to the large number of restrictions. The technology of recognition of automobile numbers is based on the technology of finding the necessary areas with YOLO technology and convolutional neural networks(CNN).

Ключові слова: YOLO, CNN, UFPR-ALPR, ГЛИБИННЕ НАВЧАННЯ, АВТОМАТИЧНЕ РОЗПІЗНАВАННЯ АВТОМОБІЛЬНИХ НОМЕРІВ

### **1. Вступ**

Автоматичне розпізнавання автомобільних номерів - популярна тема досліджень через велику кількість сфер застосування: автоматичний збір мита, дорожня поліція, контроль доступу до приватних секторів, моніторинг дорожнього руху.

Розпізнавання автомобільних номерів можна поділити на наступні три етапи:

- детектування автомобільного номеру;
- сегментація символів;
- розпізнавання.

Етап детектування має бути найбільш надійним та майже стовідсотково точним, оскільки у випадку помилок на даному етапі, наступні дії не матимуть сенсу. Існує низка методів, котрі починають етап детектування з пошуку саме машини, а вже потім – автомобільного номеру. Все це робиться з ціллю скорити час опрацювання зображення та усунути вірогідність помилкових спрацювань.

### **2. Огляд досліджень у сфері розпізнавання автомобільних номерів**

Варто зауважити, що, незважаючи на кількість досліджень на тему розпізнавання автомобільних номерів, наукових праць та алгоритмів, у реальних життєвих ситуаціях вони не завжди дають позитивні результати. Всі рішення сильно залежать від таких обмежень, як специфіка камери, кути огляду, задній план, якісне освітлення, пошук у

певній частині зображення, конкретно визначені типи засобів пересування (неможливо детектувати номерний знак на таких транспортних засобах пересування, як мотоцикли, вантажівки та автобуси).

Багато задач комп'ютерного бачення досягли значних результатів завдяки наявності великомасштабних анотованих наборів даних (наприклад, ImageNet) та апаратного забезпечення, здатного обробляти великі масиви даних. У даному випадку найкраще підходить технологія Deep Learning (DL). Однак, не дивлячись на значні досягнення алгоритмів DL у розпізнаванні автомобільних номерів, все ще є попит на набори даних автоматичного розпізнавання автомобільних номерів з анотаціями транспортних засобів та номерних плит [1]. Для технології DL обсяг даних для тренування є визначеним. Проте, якщо задіяти більше даних, можна досягти значно кращих результатів: з використанням більш надійної архітектури мережі з більшою кількістю параметрів та рівнів. Такий підхід називається UFPR-ALPR, він орієнтований на більш широкий набір даних та зданий обробляти більшу кількість сценаріїв.

### **3. Алгоритм YOLO**

Алгоритм YOLO використовує 19 конволюційних шарів та 5 шарів об'єднання. Однак, варто зазначити, що існує декілька варіацій алгоритму YOLO. Наприклад, модель Fast-YOLO, що є більш легкою та

швидкою, використовує меншу кількість слоїв (9 з 19) та меншу кількість фільтрів у цих шарах. Недоліком являється менша надійність, порівняно із YOLOv2.

У даній статті пропонується система автоматичного розпізнавання автомобільних номерів у режимі реального часу, що використовує конволюційні нейронні мережі (CNN) з YOLO детектуванням.

Через обробку відеоінформації необхідно врахувати тимчасову надмірність, що може бути викликана окремою обробкою кожного кадру відео з подальшим об'єднанням результатів.

#### 4. Огляд використаних технологій

У даному розділі мова йде про наукові досягнення, котрі були використані у запропонованій системі у контексті автоматичного розпізнавання автомобільних номерів у режимі реального часу.

Багато авторів співвідносять детектування номерного знаку з виявленням об'єктів у CNN. Наприклад, у роботі [1] використовували єдину CNN, впорядковану у вигляді каскаду, щоб знаходити машини (вигляд ззаду та попереду) та номерні знаки на них. Вони досягли достатнього рівня точності та безвідмовності. Автор роботи [2] кастомізував CNN власне під детектування номерного знаку та демонстрував більш ефективні результати. А от автор роботи [3] додав метод опорних векторів.

Крім того, в порівнянні з YOLO, YOLOv2 не має повністю підключених шарів у своїй мережевій архітектурі. Експериментальні результати цього підходу показали, що YOLOv2 може виконувати пошук об'єктів на частоті 67 FPS на GPU Nvidia Titan-X, одночасно досягаючи високої ефективності виявлення об'єктів у режимі реального часу [4].

#### 5. Опис роботи алгоритму

Модель YOLO [5], що накладає на зображення сітку, розділяючи його на осередки (рисунок 1).

Розглянемо даний алгоритм більш детально.

На вхід подається зображення певної розмірності. Скажімо, це  $448 \times 448 \times 3$  (трьохканальне). Спочатку дане зображення пропускається через частину модифікованої архітектури GoogLeNet. На виході отримуємо деякий філімакс просторової розмірності  $14 \times 14 \times 1024$ . Далі застосовується конволюція (C.R.) [6], після якої отримується тензор тієї ж розмірності. Повторюємо процедуру

конволюції другий раз, отримуємо такий самий результат.



Рис. 1 – Робота алгоритму Yolo

Після третього етапу конволюції просторова розмірність зменшується у два рази за рахунок страйду у конволюції [7]. Повторюємо процедуру конволюції у четвертий раз.

Наступний етап – FC.R., котрий повторюється двічі. На виході із другого етапу FC.R отримується вектор розмірності  $1470 \times 1$  (складається з 1470 елементів), котрі потім трансформуються у тензор  $7 \times 7 \times 30$ .

Після того, як отримано тензор розмірністю  $7 \times 7 \times 30$ , та на ньому задіємо процедуру детектування, однак спочатку необхідно його обробити.

Процедура обробки тензору полягає в наступному. На початкове зображення (розмірністю  $448 \times 448 \times 3$ ) накладається сітка розмірністю  $7 \times 7$ , що відповідає розмірності останнього тензору. Якщо взяти одну комірку, їй відповідатимуть координати на тензорі розмірністю  $7 \times 7 \times 30$  (рисунок 2).

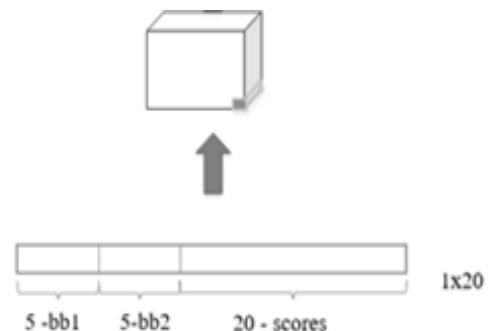


Рис.2 – Відображення обраної комірки на сітці зображення у проекції отриманого тензора

На тензорі обрана комірка має координати та певний вектор розмірністю  $30 \times 1$ .

Перші п'ять чисел відповідають за перший обрамляючий прямокутник (надалі -  $bbox$ ) для обраної комірки.

П'ять чисел діляться на наступні компоненти:

- $x$  – координати центру по осі  $X$  для першого  $bbox$ ;
- $y$  – координати центру по осі  $X$  для першого  $bbox$ ;
- $w$  – ширину першого  $bbox$ ;
- $h$  – висота першого  $bbox$ ;
- $c$  – вірогідність того, що  $bbox$  обрамляє необхідний об'єкт.

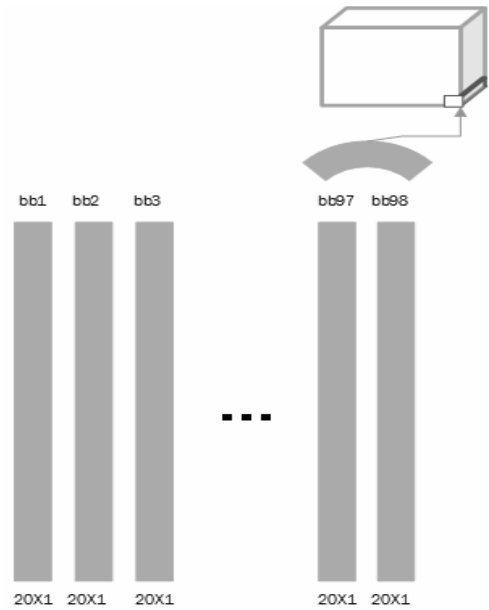
Аналогічно для другого  $bbox$ -у.

Необхідно зазначити, що для кожної комірки формується по два  $bbox$ -и, центри яких мали б співпадати із центром комірки.

Останні двадцять чисел – відповідають оцінкам для кожного із класів, де двадцять – це кількість класів, котрі модель може детектувати ( $\epsilon$  заданою із самого початку).

Оцінки по кожному з класів відображають вірогідність того, що центр об'єкта знаходиться всередині обраної комірки. Від початку дані оцінки не прив'язані до  $bbox$ -ів, однак їх можна пов'язати операцією множення елементу  $c$  на вектор  $1 \times 20$  оцінок кожного класу. В результаті отримаємо вектор  $20 \times 1$ , котрий містить оцінки для кожного класу для конкретного  $bbox$ -а. Дана

операція повторюється для кожної комірки сітки  $7 \times 7$ , таким чином отримана матриця оцінок розмірністю  $20 \times 98$  (рисунок 3).

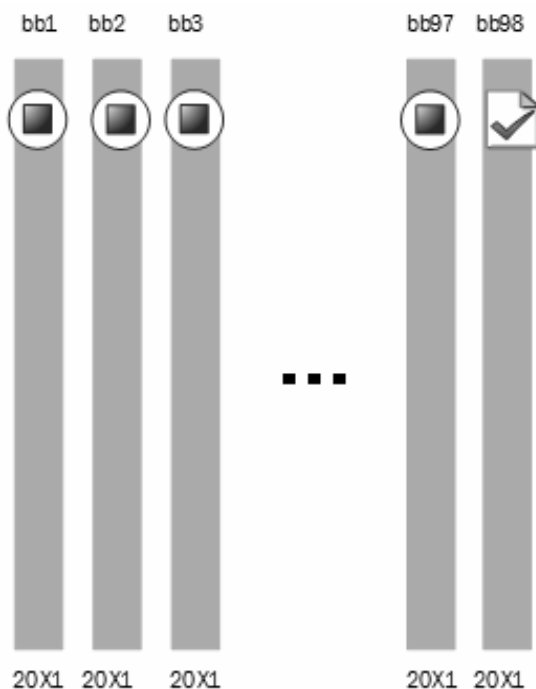


**Рис. 3 – Процедура отримання векторів оцінок для кожної комірки**

Матрицю  $20 \times 98$  необхідно обробити, задавши певний поріг проходження оцінки. Якщо оцінка нижча за поріг, вона прирівнюється до нуля. Процедура повторюється для усіх векторів. Схема вибору контуру для нанесення на зображення відображена на рисунку 4. На рисунку 5 відображена результуюча матриця оцінок.



**Рис. 4 – Схема вибору контуру**



**Рис. 5 – Результуюча матриця оцінок**

Отже, в результаті було отримано матрицю, з якою обираються комірки із максимальними оцінками, які і накладаються на зображення. Вихідним даними етапу знаходження являється область зображення, якій відповідає номерний знак автомобіля.

Дана область потім обробляється за допомогою алгоритму Tesseract [8] На виході маємо номерний знак у строковому вигляді: «CB2558AK»

## 6. Висновки

Використання алгоритму YOLO зробило можливим розпізнавання автомобільних номерів у режимі реального часу з пристрою, що знаходиться у постійному русі. Головним

чином, це пов'язано із алгоритмом накладення сітки, що дозволяє, тільки поглянувши на зображення, знаходити всі необхідні області. Ця особливість алгоритму і відображена у його назві - “You look only once”. Існує декілька версій конфігурацій і наборів даних для машинного навчання, однак найкраще обрати свій варіант, яким є найбільш «компактна» версія продукту - Tiny YOLO. Важливим плюсом Tiny YOLO являється те, що вона не втрачає швидкості, навіть працюючи на смартфонах та мікрокомп'ютерах Raspberry Pi.

## Список літератури

1. Montazzolli S. Real-time brazilian license plate detection and recognition using deep convolutional neural networks / S. Montazzolli, C. R. Jung. // SIBGRAPI Conference on Graphics, Patterns and Images. – 2017. – №30. – С. 55–62.
2. Hsu G. Robust license plate detection in the wild / G. Hsu, A. Ambikapathi, S. Chung. // IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). – 2017. – №14. – С. 1–6.
3. Rafique M. A. Vehicle license plate detection using region-based convolutional neural networks / M. A. Rafique, W. Pedrycz, M. Jeon., 2017.
4. Redmon J. A Robust Real-Time Automatic License Plate Recognition Based on the YOL Detector / J. Redmon, A. Farhadi. // University of Washington. . – 2018. – С. 1–10.
5. Redmon J. YOLOv3: An Incremental Improvement / J. Redmon, A. Farhadi. // University of Washington. . – 2016. – С. 1–6.
6. Redmon J. Real-Time Grasp Detection Using Convolutional Neural Networks / J. Redmon, A. Angelova. // University of Washington. – С. 1–7.
7. Rother C. Computer Vision I - Image Processing / Carsten Rother. // Computer Vision Lab. – 2016. – С. 41–46.
8. Smith R. An Overview of the Tesseract OCR Engine / Ray Smith. // IEEE. – 2007. – С. 629–633.

## МЕТОДИ СТИСНЕННЯ МОДЕЛЕЙ В ГЛИБИННОМУ НАВЧАННІ НА ОСНОВІ МЕТОДУ УЧИТЕЛЯ-УЧНЯ

Високі результати використання підходу глибинного навчання моделей призвели до створення мереж з більшою глибиною що, в свою чергу дозволило вирішувати більш комплексні проблеми. З іншої ж сторони це також призводить до збільшення глибини моделей, що підвищує складність та збільшує час виконання. Для вирішення цієї задачі можна використовувати стиснення об'ємів моделей. В статті наведений метод використання шумового регуляризатора за використання підходу дистиляції знань.

The high results of using the approach of deep learning of models led to the creation of models with greater depth which, in turn, allowed to solve more complex problems. On the other hand, it also leads to an increase in the depth of the models, which increases complexity and reduces the execution time. To solve this delivery, it is fashionable to use compression of model volumes. The article presents the method of using a noise regulator for using the distillation of knowledge approach

**Ключові слова:** модель, дистиляція знань, гаусів шум, глибинне навчання, нейромережі, нейрон.

### 1. Вступ

Ще в недалекому минулому потужностей обчислювальних машин не вистачало для обробки великих датасетів (datasets), які б ґрунтувалися на даних з зовнішнього світу (зображення, відео, величезні масиви даних на зразок статистики), або ж це займало неприйнятно багато часу і ресурсів. Використання глибоких мереж (deep networks) на мобільних чи не дуже потужних пристроях, на яких ресурсів не так багато, можна забезпечити за допомогою методів стиснення глибоких моделей (deep model compression methods), що надасть можливість використовувати їх набагато менше.

Видів методів стиснення (deep model compression methods) за останні роки з'явилося досить багато. Приблизною класифікацією цих методів можна вважати наданий нижче список видів подібних методів

- методи обміну параметрами (parameter sharing methods)
- методи обрізання мережі (network pruning methods)
- «темні знання» ('dark knowledge')
- методи навчання учителя-ученика (teacher-student methods)
- методи декомпозиції матриць (matrix decomposition methods)

Ці методи в основному зосередженні на зменшенні складності глибоких моделей. Після

стиснення одним із цих методів модель все одно доведеться «розпаковувати».

Одним з маловивчених методів, котрий може вирішити цю проблему є метод навчання учителя-студента (teacher-student method) [4] для стиснення глибоких моделей. Це достатньо простий підхід, де неглибока мережа (студент) навчається у глибокій мережі (вчителя).

Проблема неглибокої моделі у низькій точності в обмін на економію ресурсів та зменшення складності обробки. За таких умов досягти точності, як у глибокій моделі-вчителя, потребує набагато більше часу, що не є добре. Тоді як глибока мережа досить швидко але з використання більшої ресурсів досягає високої точності. Описані нижче методи використовуються для підвищення точності мережі-ученика без збільшення її глибини, а відповідно і кількості ресурсів.

### 2. Методи стиснення моделі

Методи стиснення моделей були створенні відносно недавно і їх можна розділити на 4 типи що будуть описані нижче.

**Метод обміну параметрами** (parameter sharing methods) передбачає використання простої (не затратної) [5] хеш-функції для групування ваг (параметрів) в певні хеш-групи (hash buckets) для розділення параметрів де кожна хеш-група відповідає одному параметру. В методі використовується  $k$ -вимірний



кластеризація [7] для повного квантування (розбивка діапазону їх значень на скінчену кількість інтервалів) параметрів пов'язаних між собою шарів моделі. Цей метод дозволяє, судячи з проведених експериментів, підвищити ступінь стиснення моделі в десятки разів (в одному з дослідів модель була стиснута в 24 рази), при втраті лише 1% точності. В даному методі регуляризація використовується замість прямого квантування параметрів лише зв'язаних шарів.

**Обрізання мережі** (network pruning methods) - полягає у відкиданні параметрів, вага яких нижче певного порогового значення заданого до навчання мережі. Цей метод можна розширити з використанням кодування Гаффмана для ще більшого скорочення кількості параметрів. Метод, в основному, спрямований на скорочення кількості обчислень та ігнорування фільтрів, які мають найменший вплив на точність. Також в рамках цього методу мережу можна дослідити надлишкові нейрони.

**«Темні знання»** («dark knowledge»). Ключовою ідеєю цієї групи методів (до яких входить метод навчання учителя-студента (teacher-student method)) в навчанні мережею, яка має більшу глибину та складність та прості моделі котрі виконують схожі задачі. Метод навчання учителя-студента (teacher-student method) був запропонований С. Vucilu, котрий помічав штучні(тестові) дані без вказівки на вчителя, від якого вони були надані. Ці мітки потім використовувалися в тренуванні меншої моделі (студента). Однією з реалізацій цього методу передбачається імітація логіт змінних (logit values) моделі вчителя. Також в цьому методі можна використовувати проміжні результати прихованих шарів моделі як цільові значення для моделі студента. Потім цей метод було узагальнено введенням певної температурної змінної в softmax функцію. Це призвело до пом'якшення результуючих даних та збільшення значення змінної температури котра надає досить важливу інформація. Ця інформація, яка виражається у відносній оцінці (в ймовірності) вихідних класів, називається «темні знання».

**Метод декомпозиції матриць** (matrix decomposition methods) [6] полягає у

використанні низькорівневої декомпозиції для стиснення ваг в різних шарах мережі. Метод перетворює щільні матриці ваг у повністю з'єднані шари до вигляду декомпозиції Тензора (Tensor decomposition), що значно зменшує кількість вхідних параметрів.

Стиснення глибоких моделей (compression of deep models) зарекомендувало себе у вирішенні трьох проблем: використання пам'яті, зменшення часу, витраченого на тренування моделі, та зменшення складності виконання. Як методи обміну параметрами (parameter sharing methods) так і метод декомпозиції матриць (matrix decomposition methods), зосередженні лише на зменшенні використання пам'яті глибокими моделями, але ці методи не зменшують час виконання (або час тренування), чи складність. Метод вчителя-студента (teacher-student method), навпаки, зосереджений на зменшенні складності та часу.

### 3. Навчання методом вчителя-ученика (teacher-student method)

В рамках методу вчителя-студента (teacher-student method) в глибинному навчанні (deep learning), вчитель - це попередньо підготовлена та навчена глибинна модель, яка використовується для навчання іншої (зазвичай неглибокої) моделі, яку називають учеником. Подібний метод має як плюси, так і обмеження, які були описані вище в розділі «Методи стиснення моделі». У використанні методу вчителя-студента є такі переваги:

- «темні знання», які присутні в результатах роботи моделі-вчителя, працюють як певні регуляризатори для моделі студента, так як вони забезпечують більш «м'який» набір знань, за яким легше відсіяти корисну інформацію;
- конвергенція (сходимість) зазвичай швидша ніж при використанні булевих міток завдяки м'яким цілям що пришвидшують тренування;
- відносно невелика кількість даних для тренування моделі-учня.

Наведені вище переваги дозволяють використати метод шумового (noise-based) регуляризатора для моделі вчителя. Далі буде описана система з одного вчителя та студента в системі вчитель-студент, яка є базою для наступних експериментів.

#### 4. Навчання моделі студента з використанням логістичної регресії

У роботі [1] запропонований метод навчання моделі студента з логарифмічною ймовірністю по змінній  $z$ , яка називається логіт функцією (logits) і є результатом шару перед викликом функції м'якого максимуму (softmax). Мережа виконує навчання за допомогою регресії з використанням логіт функції (logits) з навчальними даними поданими у вигляді  $\{(x^{(1)}, z^{(1)}), \dots, (x^{(i)}, z^{(i)}), \dots, (x^{(n)}, z^{(n)})\}$ .

Функція втрат (L2) буде мати наступний вигляд:

$$L(x, z, \theta) = \frac{1}{2T} + \sum_i \|g(x^{(i)}; \theta) - z^{(i)}\|_2^2, \quad (1)$$

де:

- $T$  розмір піднабору даних для однієї ітерації (mini-batch),
- $x^{(i)}$  це  $i$ -тий навчальний рядок в піднаборі (mini-batch),
- $z^{(i)}$  це відповідний вивід логіт функції (logits) попередньо навченого вчителя для  $x^{(i)}$ ,
- $\theta$  – набір параметрів моделі студента,
- $g(x^{(i)}; \theta)$  вивід учнівської моделі логіт функції (logits) для  $x^{(i)}$ .

Далі дана ідея буде використовуватися для додання шумів (noise-based) в знання вчителя.

#### 5. Навчання моделі студента з використанням логіт функції

Продуктивність неглибоких моделей в рамках методу вчителя-студента (teacher-student method) була значно покращена за допомогою методів запропонованих Хінтоном (G. Hinton) [12]. Але як себе поведе модель якщо її навчає декілька вчителів. Аналогічно реальному світові де студент може покращити швидкість і якість навчання одного предмету отримуючи знання про нього від декількох викладачів (з альтернативними думками, повторенням вже пройденого і т.д.), можна припустити, що схожим чином поведе себе і модель студента в рамках методу вчитель-ученик (teacher-student method). Але замість використання декількох вчителів (що збільшить кількість вхідних даних, а відповідно і час навчання) використовується методологія симуляції ефекту навчання в декількох вчителів шляхом введення «шумів» та «заплутувань» в початкові знання моделі вчителя. Результатом заплутувань є не лише імітація навчання від

декількох вчителів, а також призводить до шумів в шарі втрат (loss layer), що створює ефект регуляризатора. Таким чином, цей новий зашумлений вчитель є регуляризатором і допомагає студентам краще навчатися і отримувати результати ближчі до того якими вони є у вчителя. Метод має назву «Логіт збурення» («Logit Perturbation»).

Нехай  $\varepsilon$  вектор гаусівського шуму (Gaussian noise) з середнім  $\mu = 0$  і середнім відхиленням  $\sigma$ . Значення  $\varepsilon$  дорівнює числу класів/логітів у мережі вчителя. Якщо  $z^{(i)}$  результат вихідного шару моделі вчителя для  $x^{(i)}$  тоді  $z^{(i)}$  змінюється так:

$$z^{(i)} = (\mathbf{1} + \varepsilon), \quad (2)$$

де  $\mathbf{1}$  - одиничний вектор та  $i \in R^n$ ,  $n$  - кількість класів. Функція втрат буде виглядати наступним чином:

$$L(x, z, \theta) = \frac{1}{2T} \sum_i \|g(x^{(i)}; \theta) - z^{(i)}\|_2^2 \quad (3)$$

Значення  $\sigma$  визначає визначає збурення – більше  $\sigma$  – більше збурення оригінальних значень логіт функції вчителя  $z^{(i)}$ . Але це збурення не потрібно використовувати для всіх наборів. Замість цього потрібно вибрати набори з певної міні-вибірки з деякою ймовірністю  $\alpha$ , а тоді значення логіту обраних наборів збурюються за допомогою рівняння (2). Отримавши модель студента з початковими вагами  $\theta_0$ , ми знайдемо кінцеві параметри  $\theta$ , використовуючи метод стохастичного градієнта (Stochastic gradient descent (SGD)), де в  $(t + 1)$  ітерації  $\theta$  змінюється наступним чином:

$$\theta_{(t+1)} = \theta_t - \gamma_t \cdot \sum_{(x,y) \in D_t} \nabla_t [L(x, z, \theta)] \quad (4)$$

Тут  $T = |D_t|$  - розмір міні-вибірки, взятий випадково з навчальної вибірки  $D$ ,  $\gamma_t$  - швидкість навчання,  $L(x, z, \theta)$  – рівняння (3),  $\nabla_t [L(x, z, \theta)]$  обчислюється з використанням методу зворотного поширення помилки (gradient backpropagation).

Таким чином, деякі набори відбираються з ймовірністю  $\alpha$  з міні-вибірки. Цільовими значеннями логіт функції збурюються за допомогою рівняння (2). Функція втрат студентської мережі визначається рівнянням (3).

Давно доведено, що «зашумлені» дані допомагають регуляризувати модель. додавання регуляризації в L2 функцію втрат еквівалентно додаванню гаусівського шуму (Gaussian noise) у

вхідні дані. Регуляризована функція втрат виглядає як:

$$L(x', \theta, z) = L(x', \theta, z) + R(\theta), \quad (5)$$

де  $x'$  є гаусівським шумом (Gaussian noise), L2 еквівалентна шумним вхідним даним  $L(x, \theta, z)$ , а  $R(\theta)$  – регуляризатор L2 [3]. У вибраному методі ми збудуємо цільові вихідні дані  $z$  (шумні дані), замість вхідних даних  $x$ . Тепер покажемо, що збурення цільових вихідних даних  $z$ , значення логіт функції вчителя, еквівалентно додаванню шумного (noise-based) регуляризатора, до функції втрат. З рівняння (2) отримуємо:

$$z^{(i)} = (\mathbf{1} + \varepsilon) * z^{(i)} = z^{(i)} + \varepsilon z^{(i)}$$

Отже, можемо переписати рівняння втрат L2 (3):

$$\begin{aligned} L(x, \theta, z') &= \|z^{(i)} - g(x^{(i)}, \theta) - \varepsilon \cdot z^{(i)}\|_2^2 \\ &= \|z^{(i)} - g(x^{(i)}, \theta)\|_2^2 + \|\varepsilon \cdot z^{(i)}\|_2^2 \\ &= \|z^{(i)} - g(x^{(i)}, \theta)\|_2^2 * \|\varepsilon \cdot z^{(i)}\|_2^2 = \\ &L(x, \theta, z) + E_R, \end{aligned}$$

де  $E_R = \|\varepsilon \cdot z^{(i)}\|_2^2 + 2\|z^{(i)} - g(x^{(i)}, \theta)\|_2 * \|\varepsilon \cdot z^{(i)}\|_2$  - новий регуляризатор, який виведений на основі шуму  $\varepsilon$ . Таким чином, збурення логітів у мережі вчителя еквівалентно додаванню зашумленого (noise-based) регуляризатора до функції втрат.

## 6. Результати експериментів

Оцінка вибраного методу була проведена на декількох наборах даних, зокрема на MNIST [8]. Метод стохастичного градієнта (Stochastic gradient descent (SGD)) використовується для навчання всіх мереж, з розміром піднабору даних для однієї ітерації (mini-batch) рівному 64. Конвергенція (сходимість) була досягнута шляхом тестування на наборі підтверджених даних. В цьому розділі результати експериментів порівнюються з результатами і продуктивністю метода [1]. Результати на всіх наборах даних виглядають досить перспективно, найкращого результату вдалося досягти на наборі даних CIFAR-10.

### Використання MNIST

MNIST [8] – популярний датасет для тренування моделей розпізнавання рукописного вводу з 10 класами (0-9). Навчальний набір містить 50000 зображень та підтверджених з них 10000. Всі зображення в розмірі  $28 \times 28$  та кольорами які є градаціями сірого кольору. Тренувальні данні попередньо не

обробляються, дозволяючи виконувати власну попередню обробку на цих наборах даних.

*Модель вчителя.* У якості мережі вчителя використана модифікована мережа LeNet, котра має 2 згорткові шари і повнозв'язаний шар з 10-ма класифікаторами. (кофігурація виглядатиме так [C5 (S1P0) @ 20-MP2 (S2)])

*Мережа студент.* У якості мережі студента використана невелика мережа з двома повнозв'язними шарами з 800 нейронами в кожному шарі. Архітектура може бути закодована як: FC800-FC800-FC10

*Результати.* Модель вчитель отримала 68 помилок в тестових наборах (з 10000 тестових зразків, коеф. помилки = 0,0068). Студентська мережа допустила 97 помилок (коефіцієнт помилки = 0,0097) при базовому методі вчитель-студент (метод логіт-регресії). Так як різниця в продуктивності між вчительською та студентською мережею невисока то ймовірність відбору вибірки  $\alpha = 0,15$ , тобто приблизно 15% піднаборів кожного набору параметрів відбираються для збурення. Збурення відбувалося при різних рівнях гаусівського шуму ( $\mu = 0$ , різними  $\sigma$ ), як показано в таблиці 1. Цей шум доданий безпосередньо до ненормалізованих логітів у всіх наших експериментах у цьому дослідженні. Ми бачимо, що відбувається послідовне покращення роботи студента при застосуванні обурення до логіт.

Як видно з таблиці 1, використання гаусівського шуму для збурення ненормалізованих логітів, зменшує кількість помилок, а значить поліпшує роботу мережі студента.

**Табл.1 Результати експериментів на MNIST**

Рівень шуму ( $\sigma=std$ )	Коефіцієнт помилок	Покращення %
0.10	0.0096	1.0%
0.20	0.0093	4.1%
0.30	0.0094	3.1%
0.40	0.0087	10.3%
0.50	0.0087	10.3%
0.60	0.0090	7.2%
0.70	0.0090	7.2%
0.80	0.0086	11.3%
0.90	0.0086	11.3%
1.00	0.0087	10.3%

## 7. Висновки

Для навчання моделі використані підходи дистиляції знань та застосування шумового регуляризатора для навчання моделі студента від моделі учителя. Це призвело до пришвидшення навчання моделі студента (так як навчання, як таке, вже було проведено раніше), а за допомогою регуляризатора зменшена кількість помилок, які допускає мережа студента.

## Список літератури

1. J. Ba and R. Caruana. Do deep nets really need to be deep? In Advances in neural information processing systems, 2014.
2. G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. arXiv preprint, 2015.
3. C. M. Bishop. Training with noise is equivalent to tikhonov regularization. Neural computation, 1995.
4. C. Bucilu, R. Caruana, and A. Niculescu-Mizil. Model compression. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2006.
5. W. Chen, J. T. Wilson, S. Tyree, K. Q. Weinberger, and Y. Chen. Compressing neural networks with the hashing trick. CoRR, 2015.
6. M. Denil, B. Shakibi, L. Dinh, N. de Freitas, et al. Predicting parameters in deep learning. In Advances in Neural Information Processing Systems, 2013.
7. Y. Gong, L. Liu, M. Yang, and L. Bourdev. Compressing deep convolutional networks using vector quantization.
8. <http://www.mnis.fr/> – open source solutions for network training

УДК 334.021

ТРОЦЬОК А.Р.  
ПОПЕНКО В.Д.

## АНАЛІЗ СТРУКТУРИ СІВОЗМІН ЗА РІЗНИМИ ФОРМАМИ ЗЕМЕЛЬНОЇ ВЛАСНОСТІ В УКРАЇНІ

У статті переглянутий підхід до обґрунтування необхідності впровадження ринку землі на основі повноцінної власності на неї на противагу пануючим сьогодні відносинам оренди. Ускладнена модель побудови сівозмін врахуванням знецінювання грошових активів, яке з часом відбувається. Враховано прогноз урожайності за рахунок деякого технічного прогресу в агросфері України. Та власне побудовані самі сівозміни для обох типів власності на період від 4-х до 7-ми років.

The article revises the approach to substantiating the necessity of introducing a land market on the basis of land full ownership, in contrast to the current lease relations. The model of crop rotation is more complicated by taking into account depreciation of monetary assets, which occurs in future. Crop yield forecast is taken into account due to some technical progress in the agro-industrial area of Ukraine. And actually, we built the crop rotation for both types of property for the period from 4 to 7 years.

### 1. Вступ

Продовжуючи удосконалювати модель сформовану у нашій попередній статті на суміжну тему [1], було вирішено переформулювати задачу в більш строгому вигляді.

### 2. Задача оптимізації сівозміни для землевласника

Вважаємо, що власник земельної ділянки оптимізує один цикл сівозміни, але враховує, що в останньому році сівозміни має бути посіяна культура, прийнятна як попередник для першої в циклі культури. Дохід від земельної ділянки за період  $T$  можна розрахувати за формулою:

$$\sum_{t=2}^T \sum_{i=1}^N \sum_{j=1}^N x_{t-1i} p_{ij} x_{tj} + \sum_{i=1}^N \sum_{j=1}^N x_{Ti} p_{ij} x_{1j},$$

де  $t$  – номер року в періоді сівозміни,  $T$  – період сівозміни в роках,  $N$  – кількість культур,  $p_{ij}$  – прибуток з 1 га культури  $j$ , що вирощується після культури  $i$ . Змінна

$$x_{ti} = \begin{cases} 1, \text{ якщо культура } i \text{ вирощується в} \\ \text{рік } t \\ 0, \text{ навпаки} \end{cases}$$

Цільова функція:

$$\max \left( \sum_{t=2}^T \sum_{i=1}^N \sum_{j=1}^N x_{t-1i} p_{ij} x_{tj} + \sum_{i=1}^N \sum_{j=1}^N x_{Ti} p_{ij} x_{1j} \right),$$

Обов'язковими є обмеження на те що в один рік вирощується лише одна культура:

$$\forall t \in [1, \dots, T], \sum_{i=1}^N x_{ti} = 1$$

### 3. Задача оптимізації сівозміни для орендаря

Вважаємо, що орендар земельної ділянки оптимізує один цикл сівозміни, але не враховує, що в останньому році сівозміни має бути посіяна культура, прийнятна як попередник для першої в циклі культури. Прибуток за період  $S$  можна розрахувати за формулою:

$$\sum_{t=1}^S \sum_{i=1}^N \sum_{j=1}^N x_{t-1i} p_{ij} x_{tj},$$

де  $t$  – номер року в періоді сівозміни,  $S$  – період оренди в роках,  $N$  – кількість культур,  $p_{ij}$  – прибуток з 1 га культури  $j$ , що вирощується після культури  $i$ . Змінна

$$x_{ti} = \begin{cases} 1, \text{ якщо культура } i \text{ вирощується в} \\ \text{рік } t \\ 0, \text{ навпаки} \end{cases}$$

Так як ми вважаємо, що на початок першого року ми маємо поле в ідеальному стані, то:

$$\forall i \in [1, \dots, N], x_{0i} = 1$$

Цільова функція:

$$\max \left( \sum_{t=2}^S \sum_{i=1}^N \sum_{j=1}^N x_{t-1i} p_{ij} x_{tj} \right),$$

Обмеження:

$$\forall t \in [1, \dots, S], \sum_{i=1}^N x_{ti} = 1$$

#### 4. Ціна агрокультур

У фінансовій практиці виникає потреба оцінити майбутні грошові потоки, пов'язані з володінням певним активом. Аналогічно потрібно визначити ціну агрокультур на майбутній період. Цю потребу можна реалізувати шляхом обчислення теперішньої вартості майбутніх надходжень коштів. Для цього використовується метод дисконтування, тобто приведення грошової суми майбутнього періоду до теперішнього. Ставка дисконту — це відсоткова ставка, яка застосовується до майбутніх доходів і враховує ризик та невизначеність, пов'язані з фактором часу [2]. На ставку дисконту впливає період приведення. Чим пізніше надійдуть кошти, тим вищий дисконт застосовується до них. Чим нижчий рівень ризику, тим нижча ставка дисконту і навпаки. Слід враховувати і такий фактор, як відсоткові ставки на ринку. За їх зростання зростають і дисконтні ставки.

Щоб дізнатися теперішню вартість суми грошового потоку, який буде отриманий у рік  $t$ , потрібно домножити його на коефіцієнт дисконтування, що визначається за формулою:

$$K_t = \frac{1}{\left(1 + \frac{d}{100}\right)^{t-1}}$$

де  $d$  — дисконтна ставка, %;  $t$  — рік, на який розраховується коефіцієнт дисконтування [3].

Врахування вартості грошового потоку у часі є головною концепцією в теорії фінансів, оскільки вартість грошей сьогодні вища, ніж вартість грошей, які можуть бути отримані в майбутньому.

До основних факторів, які зумовлюють різницю у вартості грошей, належать: інфляція, ризик, втрачена вигода, схильність до ліквідності.

У фінансованому та інвестиційному аналізі використовуються метод нарощування, або компаундування, для визначення майбутньої вартості потоку готівки за формулою складних відсотків і метод дисконтування для визначення теперішньої вартості грошових потоків, які будуть одержані в майбутньому.

Розрахунок майбутньої вартості грошового потоку використовується для визначення суми депозиту залежно від строку та способу нарахування відсотків, суми боргу за кредитними угодами, при розрахунках відсоткових ставок за цінними паперами.

Визначення теперішньої вартості майбутніх грошових надходжень методом дисконтування дає змогу розрахувати ефективність інвестиційного проекту або ціну, яку можна було б заплатити сьогодні за цінні папери з визначеним доходом у майбутньому.

Порахуємо прибуток землевласника враховуючи цей коефіцієнт:

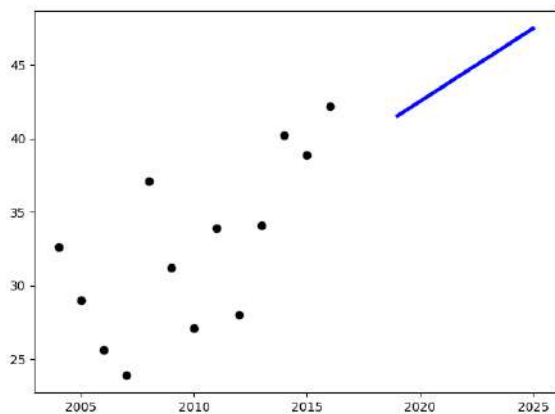
$$\sum_{t=2}^T K_t \sum_{i=1}^N \sum_{j=1}^N x_{t-1i} p_{ij} x_{tj} + K_1 \sum_{i=1}^N \sum_{j=1}^N x_{T1i} p_{ij} x_{1j}$$

Аналогічним чином зміниться прибуток орендаря.

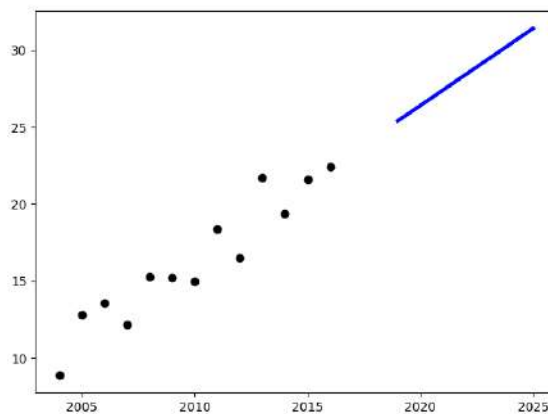
#### 5. Прогноз урожайності

Перед тим як провести основні розрахунки сівозмін для двох форм власності, потрібно зрозуміти перспективи врожаю тих чи інших культур на майбутній період. Це потрібно для визначення значення  $p_{ij}$ . Це значення залежить від прибутку із одного центнеру врожаю, яке приймемо як незмінне в часі, а також від врожайності (кількості зібраних центнерів з одного гектару), який якраз залежатиме від часу. І тому ми можемо в моделі замінити  $p_{ij}$  на  $p_{ij}^t$ .

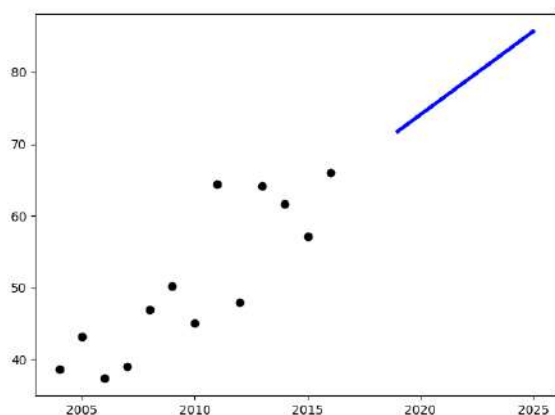
Для цього було використано інформацію про врожайність культур за період 2004-2016 роки в Україні, та побудована модель лінійної регресії. За допомогою цієї моделі було спрогнозовано урожайність на період 2019-2025 років, та щоб її можна було використати в розрахунках ми будемо вважати, що врожайність за 2019 рік — це врожайність на період  $t = 1$ , 2020 — це  $t = 2$  і т. д. На наступних графіках показана врожайність у центнерах, яка збирається з полів України у певному році. Точками зображено статистичні дані із сайту «УкрСтат», а відрізок — це частина регресійної прямої на вищезгаданому періоді прогнозованих даних.



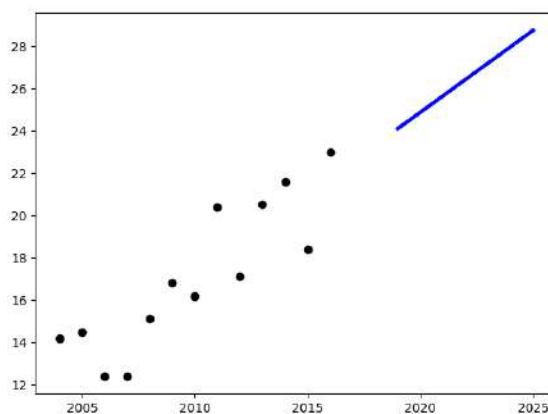
*Рис. 1. Врожайність озимої пшениці*



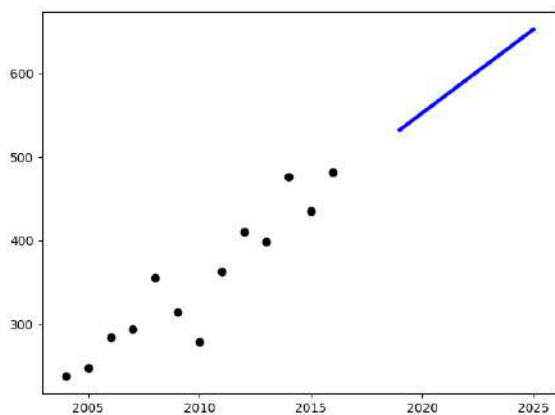
*Рис. 2. Врожайність соняшнику*



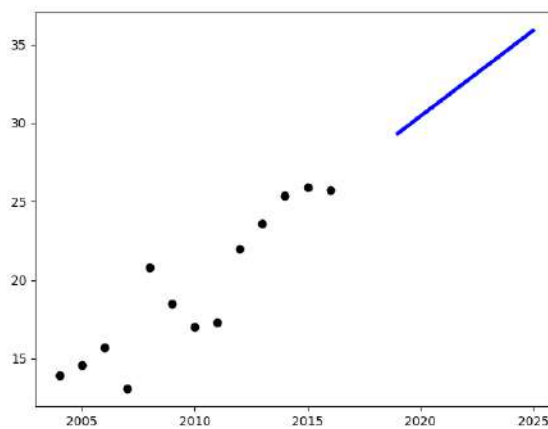
*Рис. 3. Врожайність кукурудзи*



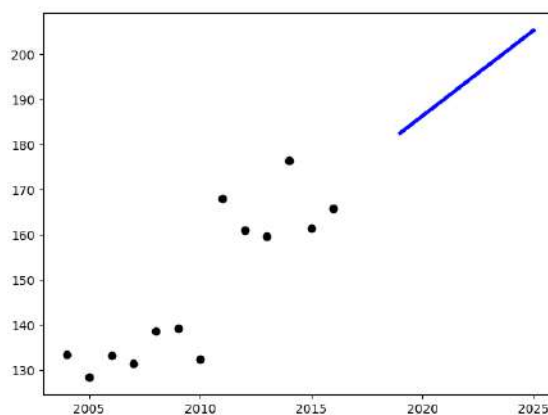
*Рис. 4. Врожайність сої*



*Рис. 5. Врожайність буряку цукрового*



*Рис. 6. Врожайність ріпаку*



*Рис. 7. Врожайність картоплі*

Для врожайності озимої пшениці можна помітити що вона не так як інші культури відображає лінію регресії. Це можна пояснити тим, що існують ще додаткові зовнішні чинники впливу, такі як погодні умови, тобто рівень опадів та температура.

Протягом історії можна побачити як розвивається сільське господарство.

Так у 60-70-х рр. XX ст. в міжнародний лексикон увійшло нове поняття – «зелена революція», що ставиться в першу чергу до країн. Це комплексне, багатокomпонентне поняття, яке в узагальненому плані можна трактувати як використання досягнень генетики, селекції та фізіології рослин для виведення таких сортів культур, при обробі яких в умовах відповідної агротехніки відкривається шлях до більш повної утилізації продуктів фотосинтезу [4].

Тому, як можна побачити із графіків, урожайність зростає з кожним роком по всіх культурах, незважаючи на те, що різкий скачок

врожайності у вигляді «революції» був ще 60 років тому. Розвиток технологій сільського господарства, до якого тягнеться Україна знаходиться у більш розвинених країнах, таких як США, країни ЄС та ін.. Для прикладу розглянемо статистичні дані із врожайності Німеччини, які наведені у табл. 1. Ця таблиця була сформована із джерел [5] та [6]. Це необхідно для того, щоб переконатися в тому, що прогнозовані значення не можуть перевищити показники обраної еталонної держави. В даній таблиці відображена врожайність в центнерах на гектар продукції в певний рік. В ті роки, у яких інформація відсутня відповідна клітинка таблиці позначена як «ND».

**Табл. 1. Врожайність основних агрокультур Німеччини**

	2018	2017	2016	2015	2014	2013	2012	2011	2010
Озима пшениця	678	769	769	815	868	803	740	706	725
Кукурудза	753	1053	965	888	1076	891	1055	1072	909
Буряк цукровий	ND	8375	7623	7217	7986	6388	6885	7430	6435
Соняшник	160	219	214	192	230	210	238	198	189
Соя	259	344	274	ND	ND	ND	ND	ND	ND
Ріпак	299	327	345	390	448	395	369	291	390
Картопля	3496	4679	4442	4381	4742	3983	4475	4576	3988

## 6. Результати досліджень

Для проведення досліджень було розроблено програму для розрахунку сівозмін. Проведемо розрахунки для задачі орендаря на період оренди від чотирьох до семи років. А також переберемо цикли сівозміни розміром 4-7 років для задачі землевласника. Сівозміни будувалися, використовуючи лише агрокультури притаманні певному регіону України.

Отже результати цієї програми на агрокультурах притаманних регіону Степ наведені на рис. 8 та 9. Для кожного періоду виводиться 3 сівозміни, після кожної сівозміни також показаний прибуток за даний період. Перша сівозміна – це є справжня циклічна сівозміна власника землі. Друга – це порядок змін культур орендаря землі. І нарешті третя сівозміна – це сівозміна наступного

орендаря землі, який цю землю отримав у користування після орендаря сівозміни 2. Ще один орендар нам потрібен для того щоб показати, що стратегія власника у довгостроковій перспективі приносить кращий результат. І дійсно як можна побачити із прибутку власника і прибутку 2-го орендаря, у періоді 5, 6 і 7 років він більший.

Також відмічаємо те, що у орендаря останньою культурою у їхньому періоді володарювання землею є соняшник, який після себе залишає землю у виснаженому стані.

У ніякій сівозміні не використовується озима пшениця яка вважається досить популярною. Це можна пояснити тим, що дана модель не враховує такий факт як те, що увесь цикл роботи із даною культурою є повністю механізований, що полегшує завдання аграріям.



Регіон: Степ

=====

Притаманні культури:

Озима пшениця: 64.46 грн/ц

Кукурудза: 88.23 грн/ц

Буряк цукровий: 16.36 грн/ц

Соняшник: 291.49 грн/ц

Соя: 194.04 грн/ц

Період: 4 р.

Власник:

Буряк цукровий, Соняшник, Кукурудза, Соняшник

Прибуток: 19793.55 грн/га

Орендар:

Буряк цукровий, Соняшник, Кукурудза, Соняшник

Прибуток: 24146.95 грн/га

Наступний Орендар:

Кукурудза, Соняшник, Кукурудза, Соняшник

Прибуток: 20188.67 грн/га

Період: 5 р.

Власник:

Буряк цукровий, Соняшник, Кукурудза, Соняшник, Кукурудза

Прибуток: 24831.95 грн/га

Орендар:

Буряк цукровий, Соняшник, Кукурудза, Буряк цукровий, Соняшник

Прибуток: 28087.89 грн/га

Наступний Орендар:

Кукурудза, Буряк цукровий, Соняшник, Кукурудза, Соняшник

Прибуток: 24200.91 грн/га

**Рис. 8. Сівозміни регіону Лісостеп періоду 4 і 5 років**

Період: 6 р.

Власник:

Буряк цукровий, Соняшник, Кукурудза, Буряк цукровий, Соняшник, Кукурудза

Прибуток: 28427.10 грн/га

Орендар:

Буряк цукровий, Соняшник, Кукурудза, Соняшник, Кукурудза, Соняшник

Прибуток: 31051.42 грн/га

Наступний Орендар:

Кукурудза, Буряк цукровий, Соняшник, Кукурудза, Буряк цукровий, Соняшник

Прибуток: 27677.38 грн/га

Період: 7 р.

Власник:

Буряк цукровий, Соняшник, Кукурудза, Соняшник, Кукурудза, Соняшник, Кукурудза

Прибуток: 31084.91 грн/га

Орендар:

Буряк цукровий, Соняшник, Кукурудза, Буряк цукровий, Соняшник, Кукурудза, Соняшник

Прибуток: 34172.97 грн/га

Наступний Орендар:

Кукурудза, Буряк цукровий, Соняшник, Кукурудза, Соняшник, Кукурудза, Соняшник

Прибуток: 30286.00 грн/га

**Рис. 9. Сівозміни регіону Лісостеп періоду 6 і 7 років**

## 7. Перелік посилань

1. Троцюк А.Р., Попенко В.Д. Дослідження стратегій поведінки землекористувачів у сільськогосподарському виробництві залежно від форм власності на землю // Інформатика та обчислювальна техніка – ІОТ-2018. – С. 79-81.
2. Розрахунок теперішньої вартості грошового потоку методом дисконтування . [Електронний ресурс] // Режим доступу: <http://library.if.ua/book/52/3782.html>
3. Оцінка ефективності проекту за методом чистої поточної (теперішньої) вартості. [Електронний ресурс] // Режим доступу: <http://studentbooks.com.ua/content/view/750/39/1/1/>
4. «Зелена революція» у сільському господарстві країн, що розвиваються. [Електронний ресурс] // Режим доступу: <http://moyaosvita.com.ua/geografija/zelena-revoljuciya-u-silskomu-gospodarstvi-kra%D1%97n-shho-rozvivayutsya/>
5. Vocabulary: Structure of production. [Електронний ресурс] // Режим доступу: <http://dd.eionet.europa.eu/vocabulary/eurostat/strucpro>
6. Eurostat. [Електронний ресурс] // Режим доступу: <https://ec.europa.eu/eurostat/web/main/home>

УДК 519.68; 681.513.7; 612.8.001.57; 007.51/.52

*ХАНЬКО ГАННА ВАДИМІВНА  
ГАВРИЛЕНКО ОЛЕНА ВАЛЕРІЇВНА,  
ОЛІЙНИК ЮРІЙ ОЛЕКСАНДРОВИЧ,*

## **АНАЛІЗ ПІДХОДІВ ДО ВИЯВЛЕННЯ ЕЛЕМЕНТІВ АГІТАЦІЇ В ТЕКСТОВИХ ДАНИХ**

Вплив на громадську думку став однією з найсильніших видів зброї в сучасних інформаційних війнах. Предметом статті є огляд загальних підходів до класифікації текстів для ідентифікації пропаганди на основі методів глибокого навчання. Було зібрано 10 тисяч сучасних текстів, створено три класифікатори: на основі згорткової, рекурентної та ієрархічно-рекурентної нейронної мережі. У цій статті зосереджена увага на політичному аспекті пропаганди. Результати дослідження показали, що сучасні алгоритми нейронних мереж можуть успішно виявляти шаблони агітації в текстових даних.

**КЛЮЧОВІ СЛОВА:** ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ, ПРОПАГАНДА, ГЛИБОКЕ НАВЧАННЯ, НЕЙРОННІ МЕРЕЖІ

Influencing the public attitude towards certain topics had become one of the strongest weapons in today's information warfare. The subject of the article is an overview of general approaches to the classification of texts for the propaganda identification based on deep learning methods. A corpus of 10,000 modern online texts was assembled and three classifiers were created: based on convolutional, recurrent and hierarchical-recurrent neural network. For the sake of this paper, we aim to focus on the political aspect. Our research has shown that modern neural network algorithms can provide end-to-end applications for identifying agitation patterns in text data.

**KEYWORDS:** DATA MINING, PROPAGANDA, DEEP LEARNING, NEURAL NETWORKS

### **1. Вступ**

Інструменти пропаганди з давніх часів використовуються для впливу на думку соціуму. Дійсно, пропаганда настільки потужна, що всі схильні до цього. Основним каналом, по якому соціум приймає пропаганду, є засоби масової комунікації. Статистика переконливості засобів масової інформації вражаюча. Поширення інформації - це індустрія з оборотами більш ніж в 400 мільярдів доларів на рік, з яких 206 мільярдів витрачаються на масове інформування, тобто на інформацію, вироблену і розподілену в ідентичній формі для споживачів по всьому світу. Кожна людина мислить вербально і тому в тій чи іншій мірі піддається навіюванню, чим вміло користуються фахівці з продажу, політики, медійники, шахраї, організатори сект, спецслужби і терористи.

Завдяки методам машинного та глибокого навчання, можливо виявити певні патерни в текстових даних, притаманні агітаційним ресурсам. Такі технології дозволять користувачам перевіряти контент на присутність спеціальних лінгвістичних конструкцій, слів та зворотів, що сприяють некритичному сприйняттю інформації [1].

### **2. Постановка задачі**

Задача класифікації текстових документів може бути сформульована як задача апроксимації невідомої функції  $\Phi: D \times C \rightarrow \{0,1\}$  (яким чином документи повинні бути класифіковані) через функцію  $K: D \times C \rightarrow \{0,1\}$ , яка є класифікатор, де  $C = \{c_1, c_2, \dots, c_{|C|}\}$   $C = \{c_1, \dots, c_{|C|}\}$  – множина можливих категорій, а  $D = \{d_1, d_2, \dots, d_{|D|}\}$  – множина документів (1).

$$\Phi(d_j, c_i) = \begin{cases} 1, & d_j \in c_i, \\ 0, & d_j \notin c_i. \end{cases} \quad (1)$$

Документ  $d_j$  називається позитивним прикладом категорії  $c_i$ , якщо  $\Phi(d_j, c_i) = 1$ , та негативним в іншому випадку. Якщо кожному документу може відповідати лише одна категорія, то це задача однозначної класифікації, а якщо декілька – багатозначної класифікації.

Знаходження класифікатора для множини категорій розглядається як пошук бінарних класифікаторів.

### 3. Опис даних

Агентство інтернет-досліджень (IRA) - це відома російська "troll" ферма, стратегічна мета якої - вносити розбрат серед американського населення та у політичній системі США. 16 лютого 2018 року спеціальний радник Роберт С. Мюллер III звинуватив тринадцять російських громадян та трьох російських організацій у втручанні в політичні та виборчі процеси США. Під час слухання члени Комітету відзначили широку активність IRA в Facebook: 3,393 купленої реклами, понад 11,4 мільйона американських користувачів піддаються впливу цих рекламних оголошень; 470 створених співробітниками IRA сторінок Facebook, аудиторія яких більш ніж 126 мільйонів американців. Рекламні пости Facebook, які використовуються у дослідженні, були ретельно розглянуті Комітетом меншин.

Згідно з даними, наданими Комітетом компанією Twitter, аналіз відповідних активностей IRA у період з 1 вересня по 15 листопада 2016 року показує: більш ніж 36 000 російських, пов'язаних з IRA ботів твітали про вибори США. Аудиторія впливу ботів - 288 мільйонів американців, кількість твітів, прив'язаних до акаунтів IRA - 130, 000.

Дані мають наступний вигляд:

- `_unit_id`: унікальний ідентифікатор запису
- `bias`: нейтральний або пропагандистський

- `message`: тематика запису
- `attack`: критика політичних осіб
- `constituency`: критика політичного виборчого округу
- `information`: інформаційне повідомлення про новини уряду та США
- `media`: повідомлення про взаємодію зі ЗМІ
- `mobilization`: повідомлення для мобілізування прихильників
- `other`: інші види повідомлень
- `personal`: особисте повідомлення, найчастіше з симпатією, підтримкою або співчуттям, або з іншими особистими думками
- `policy`: a message about political policy
- `support`: повідомлення, що закликає до політичної підтримки
- `embed`: HTML код повідомлення
- `source`: соціальна мережа, де повідомлення було опубліковане: twitter або facebook.
- `text`: текст повідомлення.

### 4. Виявлення ознак

Виявлення ознак - один з найважливіших кроків у навчання алгоритму машинного навчання.

Word2Vec - це модель, представлена компанією Google, яка навчена на великих корпусах текстових даних та представляє слова у вигляді вектору у 300-вимірному просторі. Її особливістю є те, що близькі за семантикою слова знаходяться поруч один з одним. Word2Vec є "золотим стандартом" для обробки текстових даних з використанням глибокого навчання. Модель навчається за допомогою двох алгоритмів - моделі "Continuous Bag-of-Words"(CBOW) та моделі Skip-Gram [2]. Алгоритмічно ці моделі дуже схожі, за винятком того, що CBOW прогнозує цільові слова, використовуючи слова з контексту, тоді як skip-gram використовує зворотній алгоритм і прогнозує контекст, використовуючи цільове слово. Google надає можливість використовувати натреновані Word2Vec у 100-вимірному та 300-вимірному просторі, також можна

підлаштувати модель під конкретну задачу, дотренувавши її на власних корпусах текстів.

При вирішенні поставленої задачі модель word2vec у 300-вимірному просторі було дотреновано, використовуючи записи користувачів Facebook та Twitter на політичну тематику. Розмір корпусу - 3 Гб текстових даних. На рисунку 1 наведено сформований 300-вимірний векторний простір слів з корпусу IRA.

Дослідницька група Стенфордського університету запропонувала інший метод побудови семантичних векторних просторів,

тренування моделі виконується на агрегований глобальній статистиці вживання слів у корпусі документів, а отримані уявлення демонструють цікаві лінійні підструктури отриманого векторного простору. Така модель називається Global Vectors for Word Representation (GloVe) [2]. GloVe також можна дотреновувати на власних корпусах текстів, що і було зроблено при дослідженні. Порівняльний аналіз оцінок ефективності класифікатора, використовуючи ознаки Word2Vec та GloVe, наведено у розділі Аналіз результатів.

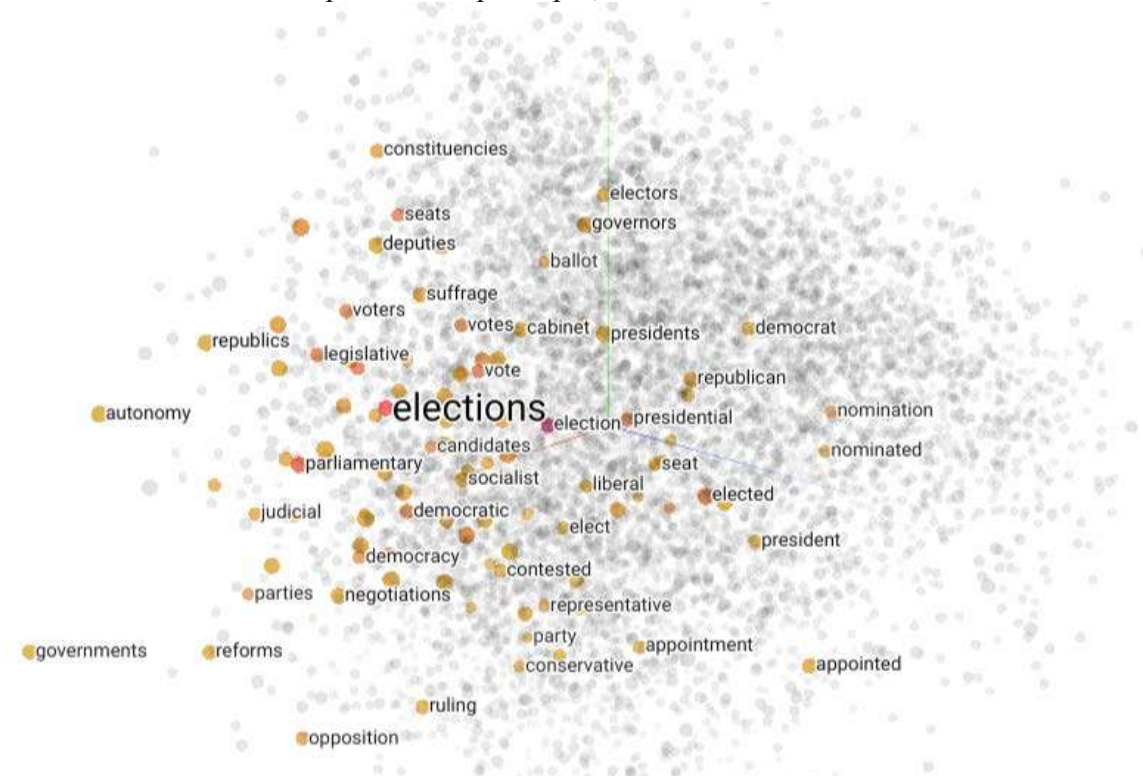


Рис. 1. Векторний простір слів корпусу IRA

## 5. Алгоритми Deep Learning

Для задачі класифікації текстових даних ми використовували дві архітектури нейронних мереж: рекурентні мережі LSTM (Long Short-Term Memory) та мультिकанальну згорткову мережу - CNN (Convolutional Neural Network).

LSTM мережа є окремим випадком рекурентних нейронних мереж. Інформація у стандартних рекурентних мережах швидко трансформується та зникає, що не дозволяє реалізовувати довгострокову пам'ять. В її архітектурі

використовуються додаткові “вентилі”: вентиль залишкової пам'яті, вхідний вентиль та вихідний вентиль. Вентиль залишкової пам'яті реалізує механізм забування, який полягає в тому, щоб визначати яка інформація вже не потрібна, а яка буде надалі використовуватися. Вхідний вентиль визначає, яка вхідна інформація може бути корисною та додає її у довгострокову пам'ять. Вихідний вентиль визначає елементи інформації, які найбільш імовірно будуть необхідні у найближчий період часу [3].

Для рішення задачі використана ієрархічна двонаправлена LSTM (H-LSTM). Слова у векторному вигляді подаються на вхід у двонаправлену LSTM на рівні речення. Кінцеві стани прямої та зворотної LSTM об'єднуються разом і надсилаються у дворівневий рівень LSTM. На кожному етапі вихідні дані прямого та

зворотного LSTM об'єднуються і надходять у кінцевий шар, який розраховує розподіл імовірності за пропагандистським забарвленням тексту. Архітектура глибокої мережі наведена на рисунку 2.

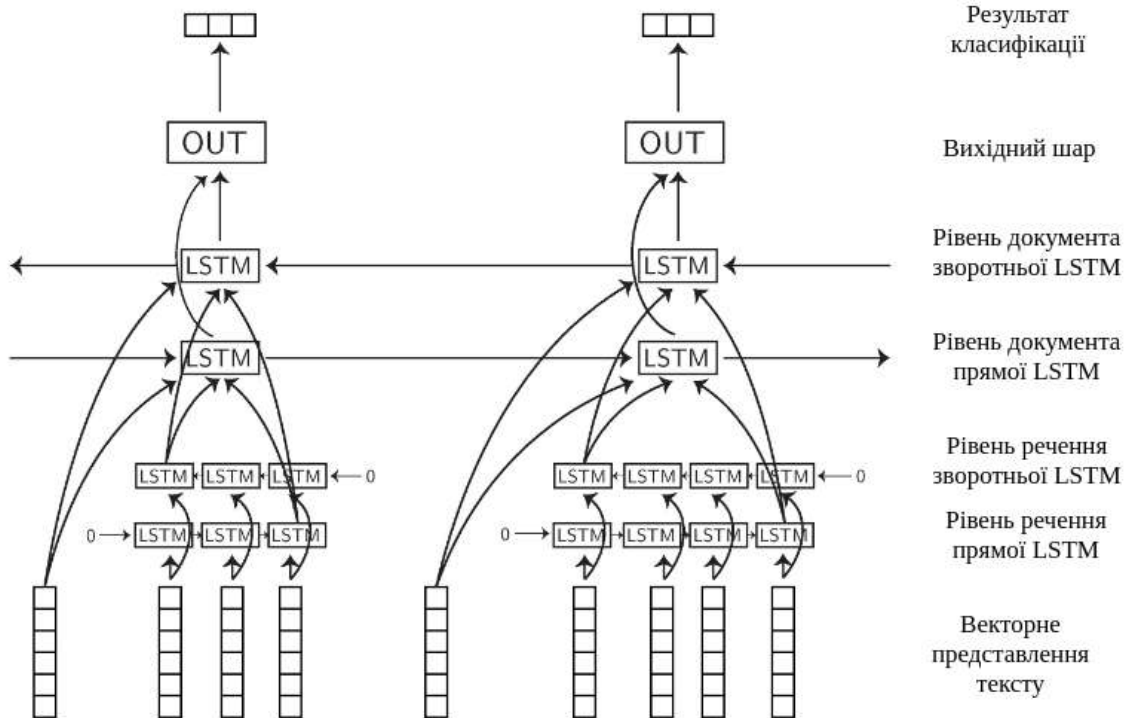


Рис. 2. Архітектура ієрархічної рекурентної мережі

Розглянемо також використану мультिकанальну згорткову нейронну мережу. Згорткові нейронні мережі найчастіше використовуються в задачах комп'ютерного зору, але дослідження показують, що згорткові нейронні мережі успішно використовуються також для обробки природної мови [4].

Кожна згортка виявляє певний спеціальний шаблон. Змінюючи розмір ядер і об'єднуючи їхні виходи, можна виявляти патерни будь-яких розмірів [5]. Шар згортки перемножує значення фільтра на вихідні значення векторних представлень слів, після чого всі результати підсумовуються. Патерни можуть бути виразами, наприклад, "я ненавиджу", " дуже добре ", і тому згорткові мережі можуть ідентифікувати їх у реченні незалежно від позиції[6].

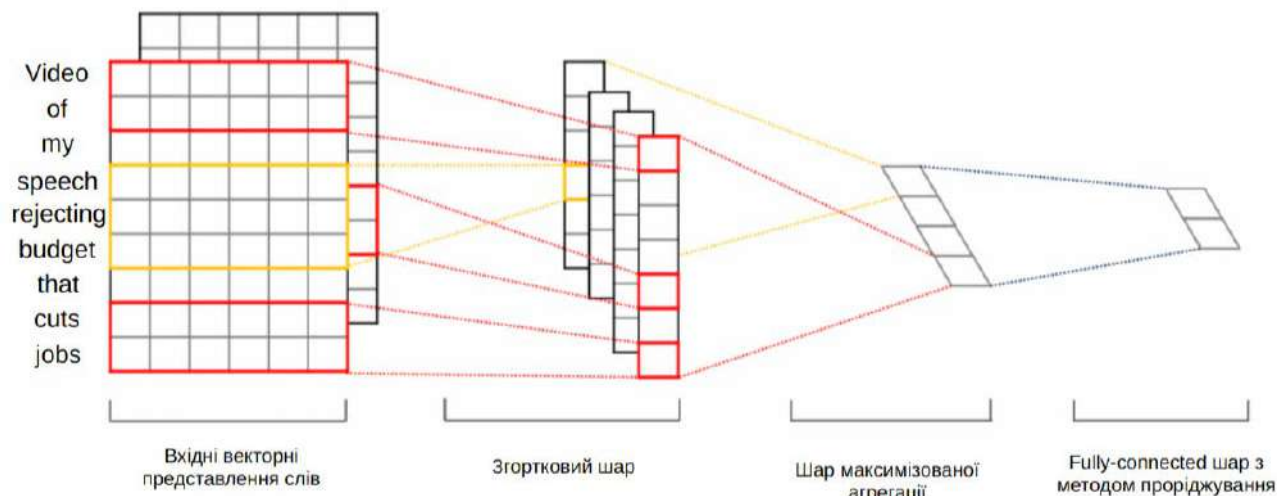
Загальноприйнято періодично вставляти шар збірки між собою

послідовними згортковими шарами в архітектурі ConvNet. Його функція полягає в тому, щоб поступово зменшувати кількість параметрів. Останнім шаром є fully-connected layer з методом проріджування. Головна ідея якого - замість навчання однієї нейронної мережі навчити ансамбль декількох нейронних мереж, а потім усереднити отримані результати[7]. Мережі для навчання виходять за допомогою виключення з мережі (dropping out) нейронів з певною ймовірністю  $q$ , таким чином, вірогідність того, що нейрон залишиться в мережі, становить  $p = 1 - q$ . "Виключення" нейрона означає, що при будь-яких вхідних даних або параметрах він повертає 0 [8].

Повнозв'язний шар виводить  $N$ -мірний вектор для визначення класу. Робота організовується шляхом звернення до карти ознак, отриманої на попередньому кроку, і визначення

властивостей, які найбільш характерні для певного класу. Архітектура використаної

згорткової мережі наведена на рисунку 3.



**Рис. 3. Архітектура згорткової нейронної мережі**

## 6. Результати

Отримані результати свідчать про те, що найкращу точність та F-score має модель, що використовує ознаки, побудовані за

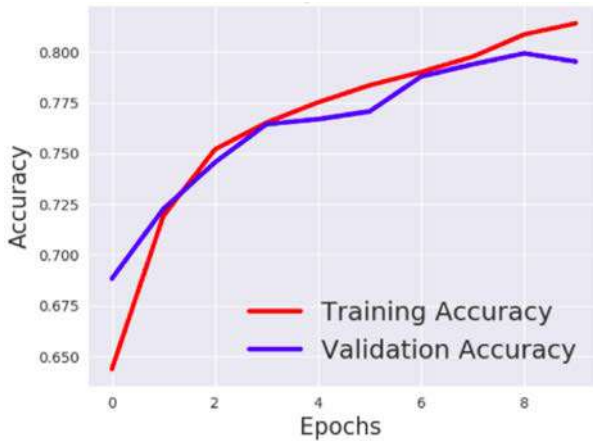
допомогою моделі word2vec та з архітектурою згорткової нейронної мережі Convolutional Neural Network (таблиця 2).

**Табл. 1. Результати випробувань**

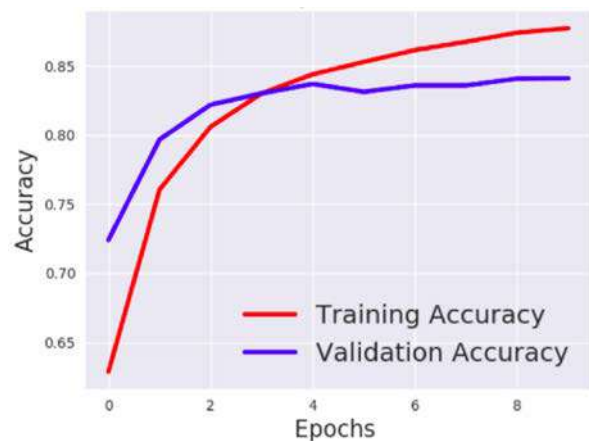
Метод побудови ознак	Архітектура нейронної мережі	Точність	F-міра
TF-IDF	LSTM	0.752	0.7
TF-IDF	H-LSTM	0.788	0.74
TF-IDF	CNN	0.761	0.72
Word2Vec	LSTM	0.837	0.80
Word2Vec	H-LSTM	0.882	0.82
Word2Vec	CNN	0.877	0.83
Glove	LSTM	0.843	0.77
Glove	H-LSTM	0.858	0.79
Glove	CNN	0.846	0.79

На рисунках 4-6 наведені графіки росту точності моделей на тренувальних даних та на валідаційних. На підставі графіків можна стверджувати, що модель,

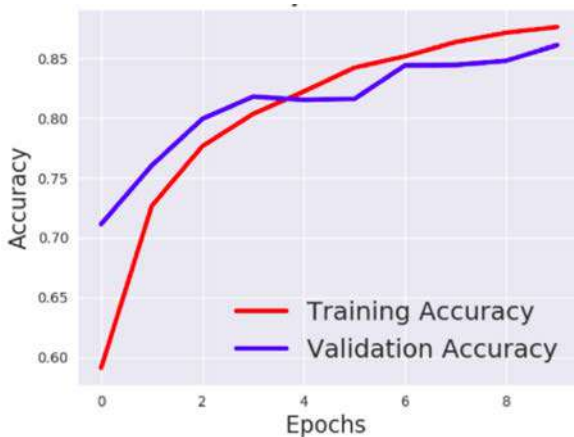
побудована за допомогою CNN, досягла високої точності валідації. На рисунку 7 наведено скільки часу займає одна епоха алгоритму для кожної архітектури.



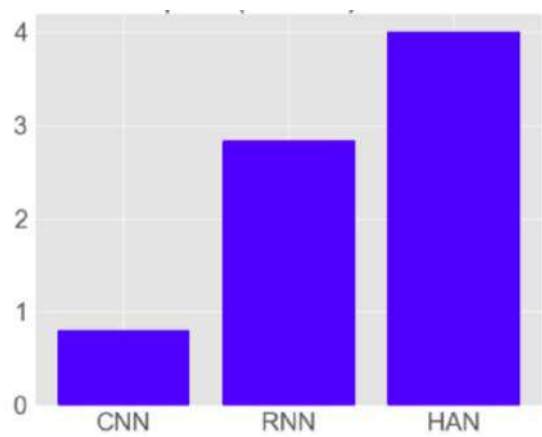
*Рис. 4. Графік точності для H-LSTM LSTM*



*Рис. 5. Графік точності для LSTM*



*Рис. 6. Графік точності для CNN*



*Рис. 7. Залежність часу тренування однієї епохи*

Отже, найкращу точність та найкращий результат F-міри має алгоритм згорткової нейронної мережі CNN.

## 7. Висновок

Дана стаття присвячена проблемі виявлення пропаганди в текстових файлах. Розробка методів і методів, які можна використовувати для цього аналізу, є важливою задачею, оскільки кількість пропаганди є величезною. Такі обсяги інформації неможливо проаналізувати спеціалістам. Дослідження того, як змінюються тематика агітації з плином часу, є цінним для розуміння того, які області нашого життя особливо охоплені пропагандою (політика, релігія, тощо), як змінюється її риторика, який вплив вона матиме. Усе вищесказане свідчить про актуальність досліджень в даній галузі.

Сучасні архітектури нейронних мереж, такі як CNN, LSTM, H-LSTM, ефективні для рішення задач обробки текстових даних. Згорткова мережа CNN показала найкращий результат - 88.2% точності.

Однією з переваг, отриманих комп'ютеризованими методами, є те, що великі обсяги даних можуть бути проаналізовані, і що легко вивчати зміни з часом. Алгоритм дозволяє реалізувати право будь-якої людини бути поінформованим про застосування примусової пропаганди. Приймати чи ні цей вплив - особистий вибір кожного.



**СПИСОК ЛІТЕРАТУРИ**

1. Agirre, E. & Edmonds, P. Word Sense Disambiguation: Algorithms and Applications (Springer Publishing Company, Incorporated, 2007). - c. 5-7.
2. D. Jurafsky and J. H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Pearson Education, Inc., 2009. - c. 130-156.
3. Y. Wang, "Various approaches in text pre-processing," TM Work Paper No, vol. 2, no. 5, 2004. - c. 1-3.
4. Berger, J. The metronome of apocalyptic time: Social media as carrier wave for millenarian contagion. Perspectives on Terror. 9 (2015). - c. 2 - 12.
5. Vinciarelli A. Noisy Text Categorization, Pattern Recognition / Vinciarelli A.: 17th International Conference on (ICPR'04), 2004. – c. 554-557.
6. S. Reid, "10 misconceptions about neural networks," May 2014. - c. 2-4.
7. C. Olah. (2015, Aug. 27). Understanding LSTM networks. colah's blog. [Online]. Available: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
8. J. Han and M. Kamber, Data Mining Concepts and Techniques. Morgan Kaufmann, 2006. - c. 12-16

## АНАЛІЗ БРОКЕРУ ПОВІДОМЛЕНЬ ДЛЯ ВИРІШЕННЯ ЗАДАЧ РОЗУМНОГО ВІДЕОРЕЄСТРАТОРА

В даній статті розглянуті найпопулярніші на даний момент брокери повідомлень, які широко застосовуються та підтримуються в різноманітних системах. Виконаний порівняльний аналіз можливих брокерів повідомлень для вирішення задачі отримання великої кількості даних від пристрою розумного відеореєстратора. Демонструється різниця обраних технологій для реалізації поставленої задачі. Описані основні технології та їх конкуренти у даній сфері, створення приладу розумного відеореєстратора.

This article presents the most popular brokers of messages that are widely used and supported in a variety of systems. The comparative analysis of possible brokers of messages is carried out for solving the problem of obtaining a large amount of data from the device of a smart DVR. Difference between selected technologies for the realization of the task is demonstrated. The main technologies and their competitors are described, so is creation of the device of smart video recorder.

Ключові слова: ВІДЕОРЕЄСТРАТОР, DVR, АРАСНЕ КАФКА, РАВВІТ MQ, АРАСНЕ АКТИВ MQ, HETZNER

### 1. Вступ

Для успішного ведення будь-якого проекту необхідно чітко розуміти клієнта та його основні вимоги. Загальною проблемою патрулювання поліцейськими районів міста є те, що дані, які поступають з транспортного засобу ніде не зберігаються та не аналізуються, тобто, не виконується ніякий аналіз транспортних засобів, які вони бачать під час патрулювання. Наразі використовуються body-камери, які записують усі дії полісмена, але встановлений в автомобілі відеореєстратор використовується тільки для запису відео. Обмін повідомленнями є важливою функцією розумного відеореєстратора. У даній статті розглянуті технології, ціль яких нормалізувати навантаження при великій кількості одночасно поступаючих вхідних повідомлень до системи.

### 2. Мета дослідження

Мета статті полягає у дослідженні можливих брокерів повідомлень [1], їх порівняння та знаходження найкращого брокера для ефективного використання в приладі розумного відеореєстратора. Визначення архітектури проекту при використанні брокера повідомлень за допомогою тестування, що підтверджені за допомогою порівняльних таблиць та графіків, що додані до викладеного матеріалу.

### 3. Особливості обробки вхідної інформації та вимоги

Даний процес розділений на 4 етапи:

- 1) збір даних з відеореєстратора та їх запис до локальної пам'яті мікрокомп'ютеру;
- 2) класифікація та розпізнавання збережених даних;
- 3) збереження проаналізованих даних до серверу;
- 4) аналіз даних всередині системи.

При проектуванні системи потрібно врахувати наступні критерії.

**Вразливість інтернету.** Через знаходження транспортного засобу, на якому встановлений прилад, у постійному русі, зв'язок з сервером може бути розірваний та може бути недосяжний протягом певного періоду часу.

**Обмежена кількість пам'яті.** Через обмеженість ресурсу пам'яті виникає необхідність механізму автоматичного очищення інформації.

**Персистентність даних.** Дані повинні в будь-якому випадку бути збережені.

**Ізольованість.** Жодний з пристроїв не може перешкоджати роботі іншого пристрою.

**Стійкість.** Результат аналізу відеопотоку не повинний бути втрачений за жодних умов.

**Паралельність виконання аналізу.** Збір інформації з приладу та аналіз вхідної інформації не повинні шкодити клієнту.

**Маштабування.** Система має бути маштабована, оскільки в будь-який момент часу можливе лінійне збільшення вхідної інформації [2].

#### 4. Аналіз існуючих рішень

Для вирішення проблеми великої кількості повідомлень використовується брокер. На даний момент існують популярні рішення.

Apache Kafka - розподілений брокер повідомлень, що працює за принципом підписок на події.

Active MQ [3] - розподілений брокер повідомлень з можливістю збереження даних: як в пам'яті, так і в будь-які бази даних.

Rabbit MQ [4] - платформа, яка реалізує систему обміну повідомленнями між компонентами програмної системи на основі стандарту AMQP (Advanced Message Queuing Protocol).

В даному випадку використовується Apache Kafka, оскільки за тестами вона показує надійні результати за наявних даних. Дана система позиціонує себе як персистентна, та система, яка може приймати велику кількість повідомлень в одну одиницю часу. Ключовими

перевагами використання Apache Kafka є наступні критерії.

1. Легко маштабована розподілена система.
2. Велика пропускна здатність.
3. Можливість зручного об'єднання за групами.
4. Використання TCP протоколу.
5. Наявність адмін-панелі для контролювання брокеру повідомлень.

Основні компоненти архітектури наведені далі.

1. Topic - потік повідомлень одного типу, в яку producer посилає повідомлення.
2. Producer - система, яка генерує повідомлення.
3. Consumer - система, яка обробляє повідомлення, які надходять від Producer.
4. Partition - частина топіку (кількість наявних черг, в кожних з яких є свій offset) [5].

На рисунку 1 наведена схема роботи брокера повідомлень від різних приладів (відеореєстратори).

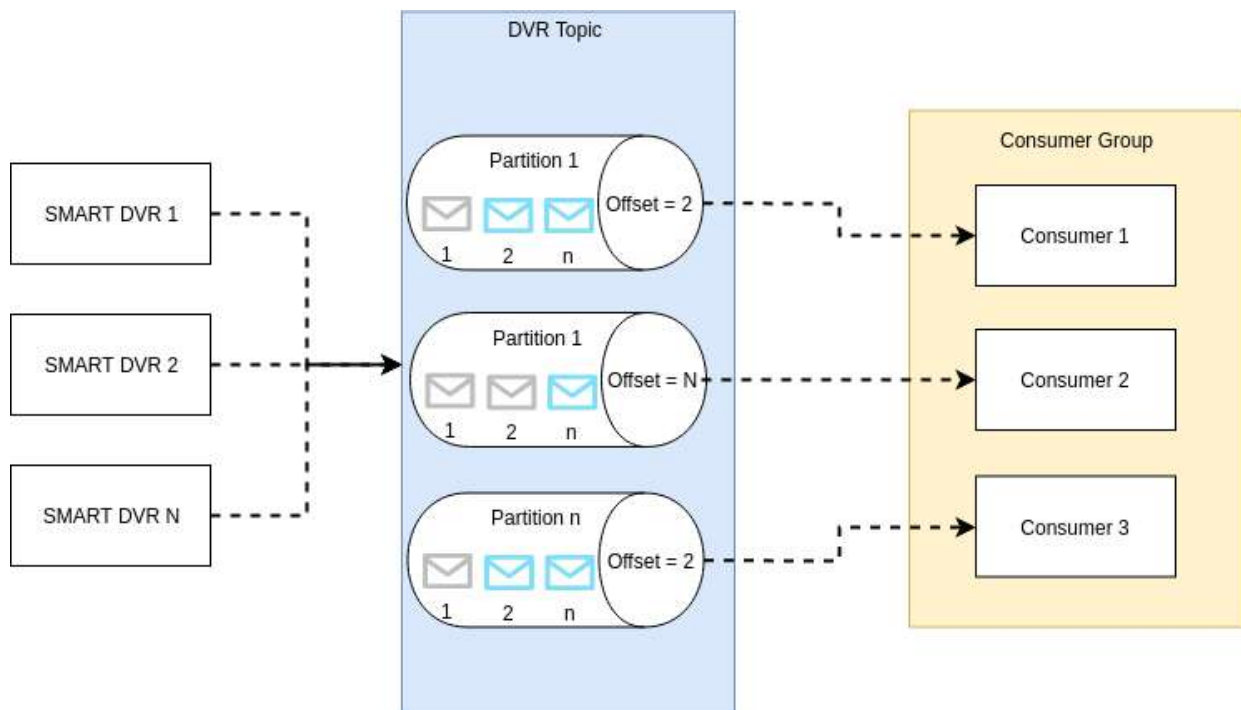


Рис. 1. Схема роботи брокера повідомлень Apache Kafka

#### 5. Методика моделювання навантаження

Для тестування використовуються 4 сервери орендовані в німецькому дата-

центрі Hetzner. Дані серверів та їх параметри наведені в таблиці 1.

Для систем Active MQ (LevelDB), Rabbit MQ (за замовчуванням) налаштована

персистентність. Система працювала 24 години (для наповнення черги), та в тесті використовуються дані за останню хвилину роботи системи.

В результаті тестувань отримані наступні дані, які демонструють затримку на відправлення повідомлення до різних брокерів (таблиця 2).

**Табл. 1 – Сервери для тестування та їх характеристики**

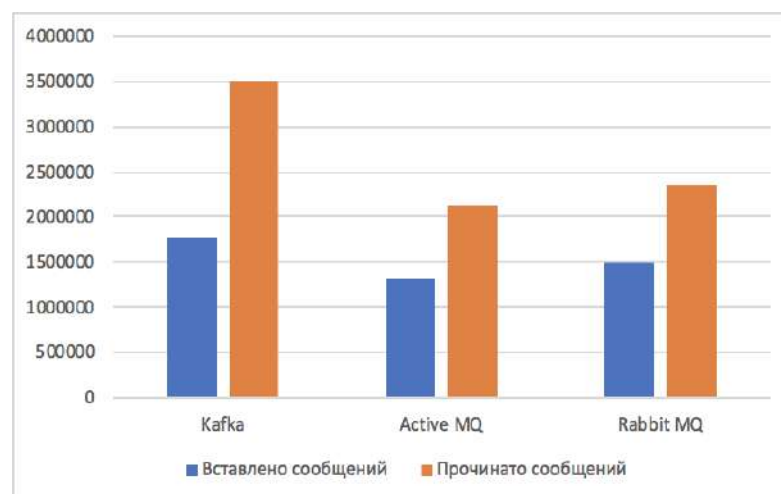
Назва серверу	Параметри серверу	Опис
Producer 1 Producer 2	4 vCPU 16 GB RAM 160 GB SSD	В 50 потоках надсилають повідомлення по 1-2 Кб (розмір повідомлень в нашій системі)
Брокер повідомлень	4 vCPU 16 GB RAM 160 GB SSD	Kafka (2 partition) Active MQ Rabbit MQ
Consumer 1 Consumer 2	4 vCPU 16 GB RAM 160 GB SSD	Приймає повідомлення та виводить інформацію в консоль

**Табл. 2 – Результати випробувань**

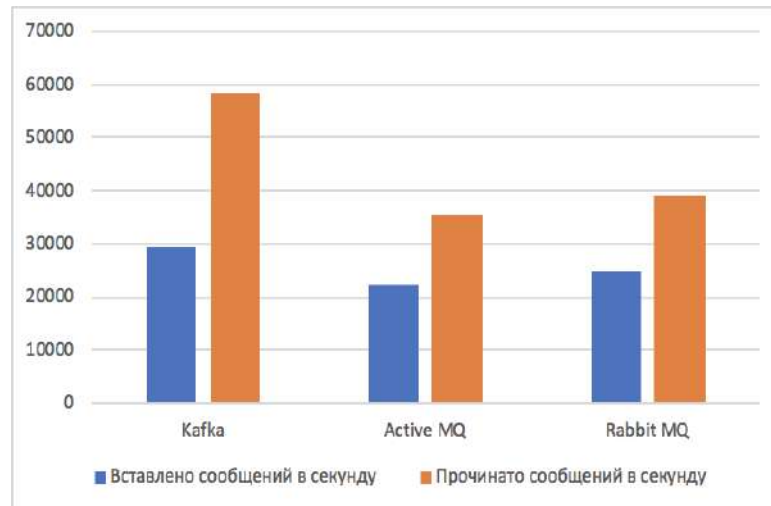
Назва брокеру повідомлень	Максимальна кількість мілісекунд, витрачених на відправку до брокеру	Мінімальна кількість мілісекунд, витрачених на відправку до брокеру
Kafka	120 (мс)	56 (мс)
Active MQ	1138 (мс)	48 (мс)
Rabbit MQ	483 (мс)	63 (мс)

На рисунку 2 наведені результати тестування відправлення та обробки повідомлень за 1 хвилину в 50-ти потоках (тобто, в 50-ти одночасно працюючих Producer-ах) для брокерів повідомлень Kafka, Active MQ, Rabbit MQ.

На рисунку 3 наведені результати тестування відправлення та обробки повідомлень на вищеписаних брокерах повідомлень за інший період часу - за 1 секунду.



**Рис. 2. Тестування відправлення та прийому повідомлень за 1 хвилину**



**Рис. 3. Тестування відправлення та прийому повідомлень за 1 секунду**

Наведені графіки показують, що саме в нашому випадку Apache Kafka є найкращим варіантом, але кожний з брокерів має свої недоліки та переваги. Якщо використати іншу базу даних для Active MQ, тоді, можливо, показники стануть кращими. Якщо налаштувати кластер, який буде зберігати дані лише в пам'яті, Apache Kafka видасть набагато гірші результати, оскільки в Apache Kafka неможливо виключити

персистентність. Кожний з даних брокерів має неповторні механізми, які підходять під різні проекти, наприклад, Scheduled механізми в Active MQ, які вміють додавати відкладені повідомлення. Отримані дані неможливо розглядати за еталон, оскільки на іншій конфігурації системи, в іншому розмірі повідомлень, розмірі кластера, можуть бути отримані протилежні результати.

## 6. Висновки

Використання брокерів повідомлень надає можливість розробити масштабовану систему з мінімальними витратами та максимальною надійністю. Отримані результати тестування показують, що Apache Kafka є найкращим механізмом для нашого проекту. Використання даного брокера повідомлень надає впевненості, що дані збережені, та завжди існує можливість їх повторно проаналізувати.

## Список літератури

1. Saxena S. Practical Real-time Data Processing and Analytics: Distributed Computing and Event Processing using Apache Spark, Flink, Storm, and Kafka / Shilpi Saxena., 2017. – 360 с.
2. Narkhede N. Kafka: The definitive guide: Real-time data and stream processing at scale. Sebastopol / Narkhede. – Sebastopol: CA: O'Reilly Media., 2017.
3. Snyder B. ActiveMQ in Action / Bruce Snyder., 2018. – 375 с. – (Manning Publications).
4. Boschi S. RabbitMQ Cookbook / Sigismondo Boschi., 2013. – 288 с.
5. Bejeck B. Kafka Streams in Action: Real-time apps and microservices with the Kafka Streams API / Bill Bejeck., 2018. – 280 с. – (1 edition).

## ПІДСИСТЕМА МОДЕЛЮВАННЯ ПОВЕДІНКИ ВЕЛИКОЇ КІЛЬКОСТІ ЛЮДЕЙ В ОБМЕЖЕНОМУ ПРОСТОРИ

У цій статті розглянуто практичне застосування методу калібрування математичної моделі алгоритму моделювання поведінки натовпу за допомогою алгоритму локального пошуку. Описана формальна процедура оцінки якості калібрування моделі. На прикладі надзвичайної ситуації, що відбулася у реальному житті описується процес калібрування та його кількісні та якісні оцінки.

**КЛЮЧОВІ СЛОВА:** МОДЕЛЬ, НАТОВП, ПАНІКА, ФІЗИЧНА МОДЕЛЬ, ДИНАМІКА НАТОВПУ, ПОЖЕЖНА БЕЗПЕКА, ЛОКАЛЬНИЙ ПОШУК

In this article the practical application of the method of calibration of the mathematical model of the algorithm for modeling the behavior of the crowd with the help of the local search algorithm is considered. The formal procedure for assessing the quality of the model calibration is described. An example of an emergency situation in real life describes the calibration process and its quantitative and qualitative assessments.

**KEYWORDS:** MODEL, CROWD, PANIC, PHYSICAL MODEL, CROWD DYNAMICS, FIRE SAFETY, LOCAL SEARCH

### 1. Вступ

У цій статті розглянуто практичне застосування методу калібрування математичної моделі алгоритму моделювання поведінки натовпу за допомогою алгоритму локального пошуку. Описана формальна процедура оцінки якості калібрування моделі. На прикладі надзвичайної ситуації, що відбулася у реальному житті описується процес калібрування та його кількісні та якісні оцінки.

### 2. Модель поведінки натовпу

Для побудови моделі було використано методи і підходи молекулярної динаміки, що довела свою ефективність при дослідженні поведінки газів і рідин. В рамках цього підходу кожна людина розглядається як окрема молекула, а правила взаємодії між людьми задаються за допомогою спеціально підібраних потенційних сил психосоціального і фізичного походження. Взагалі кажучи, за своєю природою людина незмірно складніше будь-якої фізичної системи і в загальному випадку її поведінка не може бути зведена до набору простих правил взаємодії. Однак в умовах паніки свідомість індивіда відключається і він діє подібно до автомату, рефлекторно відповідаючи на зовнішні подразники. Саме тому молекулярна динаміка

може допомогти при описі динаміки натовпу як колективної несвідомої дії.

У даному випадку розглянемо математичну модель, у якій на людину у натовпі діють 3 основні сили [1, с.20]:

- так звана «сила паніки», що зумовлена бажанням людини покинути приміщення  $\overline{F}^{pan}$ ;
- сила взаємодії між людиною та іншими людьми  $\overline{F}_{ij}$ ;
- сила взаємодії між людиною та перешкодами  $\overline{F}_{im}^{cm}$ .

Постановка задачі зводиться до розрахунку суми цих трьох сил для кожного агента на кожній ітерації процесу симуляції та розрахунок координат наступного положення агента [1]:

$$m_i \frac{d\overline{V}_i}{dt} = \overline{F}_i^{pan} + \sum_{i \neq j} \overline{F}_{ij} + \sum_m \overline{F}_{im}^{cm}$$

Розглянемо безліч з  $N$  людей, абстрактно представлених в моделі у вигляді кульок з радіусом  $R_i$  і масою  $m_i$ , де  $i = \overline{1, N}$ . У вибраній системі координат поведінка кожного індивіда характеризується радіус вектором  $\overline{r}_i$ , актуальною швидкістю пересування  $\overline{V}_i$  і швидкістю, з якою людина хотіла б пересуватися  $\overline{V}_i^{баж}$ . Саме невідповідність

бажаної швидкості з актуальною і формує основну силу, що створює рух.

$$\vec{F}_i^{nan} = \frac{\vec{V}_i^{баж} - \vec{V}_i}{\Delta t_i} \quad (1)$$

Тут  $\Delta t_i$  - параметр, що характеризує час залучення індивіда в панікуючий натовп. Сила (1) влаштована таким чином, що в разі перевищення бажаної швидкості над актуальною людина прискорюється. Якщо індивід не бажає нікуди рухатися, то його рух з часом згасає. У спрощених моделях бажана швидкість розглядається як параметр завдання, що задається перед розрахунковим циклом. Якщо предметом дослідження є процес залучення нових членів в панікуючий натовп, то ця швидкість повинна вважатися змінною і вона може змінювати своє значення в залежності від якихось чинників. Наприклад, група людей, що біжать, несподівано потрапляє в поле зору розглянутого індивіда, може різко змінити значення його бажаної швидкості і змусити приєднатися до натовпу.

Крім сили (1), яка має психологічний характер і в загальному випадку формується незалежно від групової динаміки, в процесі переміщення на людину діють інші сили, пов'язані з взаємодією людини з іншими людьми і перешкодами[3]. Розглянемо силу, діючу на індивіда  $i$  з боку індивіда  $j$ :

$$\vec{F}_{ij} = A n_{ij} \exp\left(\frac{D_{ij}}{B}\right) + k H(D_{ij}) D_{ij} \vec{n}_{ij} + k H(D_{ij}) D_{ij} ((\vec{V}_j - \vec{V}_i) \vec{\tau}_{ij}) \vec{\tau}_{ij} \quad (2)$$

Тут  $D_{ij} = R_i + R_j - |\vec{r}_i - \vec{r}_j|$ ,  $\vec{n}_{ij}$  і  $\vec{\tau}_{ij}$  - нормальний і тангенціальний одиничні вектори;  $A, B, k$  - параметри [2]. Перший доданок в (2) являє собою силу, так само як і у формулі (1), психосоціальної природи. Вона описує небажання людини вступати в занадто тісний фізичний контакт з іншими людьми. Потенціал влаштований так, що у випадку, коли відстань між індивідами стає менше суми їх радіусів, сила відштовхування стає особливо значною. На великих відстанях внесок цього доданка стає слабкішим. Друга і третя складові в (2) мають традиційний для молекулярної динаміки вид. Одне з них описує пружне відштовхування при лобовому зіткненні між індивідами, а друге - силу тертя при дотичному русі. Обидва взаємодії

виникають тільки при позитивних значеннях  $D_{ij}$ , тому ці складові множаться на функцію Хевісайда  $H(D_{ij})$ . Пружне відштовхування і тертя є принциповими членами для моделювання поведінки натовпу. Якщо знехтувати цими ефектами, багатоістотні риси паніки натовпу в моделюванні не відображаються.

За аналогією з силою (2), що діє між індивідом і іншим індивідом, можна записати силу, яка виникає між індивідом і фізичною перешкодою, наприклад стіною:

$$\vec{F}_{im}^{cm} = A^{cm} \vec{n}_{im} \exp\left(\frac{D_{im}^{cm}}{B^{cm}}\right) + k^{cm} H(D_{im}^{cm}) D_{im}^{cm} \vec{n}_{ij} - k^{cm} H(D_{im}^{cm}) D_{im}^{cm} \langle \vec{V}_i \vec{\tau}_{im}^{cm} \rangle \vec{\tau}_{im}^{cm} \quad (3)$$

де  $D_{ij}^{cm} = R_i - |\vec{r}_i - \vec{r}_m|$ , а  $\vec{r}_m$  - радіус-вектор до найближчої до індивіда точки і стіни  $m$ . Значення інших змінних і параметрів аналогічно значенням змінних і параметрів в (2). З урахуванням формул (1-3) рівняння динаміки паніки натовпу можуть бути записані в звичайному для механіки вигляді:

$$m_i \frac{d\vec{V}_i}{dt} = \vec{F}_i^{nan} + \sum_{i \neq j} \vec{F}_{ij} + \sum_m \vec{F}_{im}^{cm}$$

$$\frac{d\vec{r}_i}{dt} = \vec{V}_i$$

### 3. Постановка задачі калібрування моделі поведінки натовпу

У вищезазначених формулах (2) та (3) фігурують коефіцієнти  $A, B, k$ , найбільш оптимальні значення яких і необхідно знайти.

Коефіцієнт  $A$  відповідає за силу відштовхування агента від інших агентів та перешкод до того моменту, коли вони доторкнуться один до одного. Це так зване небажання наближатися до інших агентів та перешкод. Коефіцієнт  $B$  відповідає за швидкість зростання цієї сили відштовхування. Коефіцієнт  $k$  відповідає за характер сили відштовхування між агентами та між агентом і перешкодою, якщо вони доторкаються один до одного. Ці коефіцієнти відповідають за характер руху та поведінку агентів, тому від їх зміни напряму залежить точність та результати моделювання. Тому повстала задача розробки алгоритму пошуку оптимального набору цих коефіцієнтів.

На вхід алгоритму подається початковий набір коефіцієнтів моделі, який буде початковою точкою. Проводиться симуляція із поточним

набором коефіцієнтів, за результатами якої буде розраховано цільову функцію у даній точці - сума середніх квадратичних відхилень кожного параметру, за якими ми порівнюємо проведену симуляцію та подію, що мала місце у реальному житті. Параметри, що порівнюються – це час, за який агенти територію надзвичайної ситуації (площу, будівлю, ін.), кількість травмованих людей та кількість загиблих (якщо такі є). Після цього буде обрано наступну точку серед околу поточної точки або алгоритм локального пошуку припинить роботу, якщо досягнуто умову завершення.

#### 4. Огляд методу розв'язання

В якості основного методу розв'язання задачі було обрано алгоритм локального пошуку. Локальний пошук є евристичним методом для рішення задач обчислювальної жорсткої оптимізації. Локальний пошук може бути використаний для проблем, які можна сформулювати як пошук рішення, що максимізує критерій серед ряду рішень кандидатів. Алгоритми локального пошуку переходять від рішення до рішення в просторі рішень кандидатів (пошуковий простір), застосовуючи локальні зміни, доки не з'ясовано оптимальний варіант рішення, або термін пошуку минув.

Ключовими аспектами реалізації алгоритмів локального детермінованого пошуку є:

1. визначення околів  $O(x)$ ;
2. генерація чергової точки  $u \in O(x)$ ;
3. критерій завершення перегляду точок у поточному околі та переходу до наступного;
4. спосіб обчислення величини зміни цільової функції при переході до нового поточного варіанта;
5. критерій завершення.

Наведемо опис ключових аспектів реалізації алгоритму, що пропонується.

1. Визначення околів та генерація чергової точки околу.

Власне точкою будемо вважати набір параметрів математичної моделі. Вважатимемо два розв'язки сусідніми, якщо вони відрізняються між собою лише одним параметром та різниця складає не більше, ніж

$1/10$  від поточного значення даного параметра.

2. Критерій завершення перегляду точок у поточному околі та переходу до наступного.

Перехід до наступного розв'язку здійснюється при знаходженні першого кращого за поточний розв'язок сусіднього розв'язку.

3. Спосіб обчислення зміни цільової функції при переході до нового поточного варіанта. Цільова функція у даній задачі буде розраховуватись за наступним алгоритмом. Для початку, необхідно зібрати необхідну інформацію про будь-яку надзвичайну подію, що пов'язана зі скупченням великої кількості людей. На щастя, у інформаційній системі Інтернет є інформація про швидкість евакуації, кількість людей у натовпі, якісний склад натовпу, план приміщення для подій подібного роду. Далі для кожної розглянутої точки околу буде проведено симуляцію із значеннями параметрів у даній точці. Цільовою функцією буде сума середніх квадратичних відхилень кожного параметру, за якими ми порівнюємо проведену симуляцію та подію, що мала місце у реальному житті. Параметри, що порівнюються – це час, за який агенти територію надзвичайної ситуації (площу, будівлю, ін.), кількість травмованих людей та кількість загиблих (якщо такі є).

4. Критерій завершення.

В якості критерію завершення обрано відсутність покращення результату у останніх  $N$  точках або перевищення заданої максимальної кількості ітерацій

#### 5. Збір необхідної для калібрування інформації

Для калібрування математичної моделі було зібрано інформацію про надзвичайні події, пов'язані із скупченням великої кількості людей, що мали місце у реальному житті. Зокрема, було обрано надзвичайну ситуацію у клубі “Station” 20 лютого 2003 року, в наслідок якої від тисняви та удушення загинуло більше 100 людей. Було знайдено креслення будівлі, точна кількість жертв та поранених та час, за який люди покинули приміщення. Саме ці дані обрані як еталонні та будуть порівнюватися із результатами проведених під час калібрування симуляціями.



### **Висновки**

В процесі роботи на системою було розроблено математичну модель симуляції поведінки натовпу. Шляхом методу експертних оцінок було задано початковий набір параметрів для калібрування математичної моделі. Розроблено алгоритм пошуку оптимального набору коефіцієнтів моделі, що мав на меті збільшення точності поведінки агентів. Зібрана необхідна для процесу калібрування інформація про надзвичайні події, що сталися у реальному житті.

### **Список літератури**

1. Белкарян А. Л., Акопов А. С. «Моделирование поведения толпы на основе интеллектуальной динамики взаимодействующих агентов» // Бизнес-информатика № 1 (31), 2015 С. 69-77
2. Р. В. Гребенников «Способы оценки эффективности различных моделей поведения толпы» // Вестник Воронежского Государственного Университета, №1, 2010
3. Засоби оцінки ефективності різних моделей поведінки натовпу [Електронний ресурс] // <http://www.vestnik.vsu.ru/pdf/analiz/2010/01/2010-01-21.pdf>

УДК 004.021

ЯРОЩУК О.С.,  
ОЛІЙНИК Ю.О.

## Математичне та програмне забезпечення візуалізації та підготовки структур даних на клієнтському інтерфейсі з використання технології Javascript

Запропоновано використання алгоритму оптимізації автоматичного написання HTML/CSS та JavaScript для підвищення продуктивності коду. Проаналізовано результати ефективності використання створеного програмного рішення методом попарного порівняння та розрахунку коефіцієнта технічного рівня.

Ключові слова: HTML, JAVASCRIPT, ФРЕЙМВОРК, КОМПІЛЯЦІЯ, ДЕРЕВА

The use of the algorithm for automatic writing of HTML / CSS and JavaScript is proposed to increase the productivity of the code. The analysis of the efficiency of using the created software solution by pairwise comparison and calculation of the technical level coefficient.

Keywords: HTML, JAVASCRIPT, FRAMEWORK, COMPILATION, TREES

### 1. Вступ

Сьогодні не оптимізований громіздкий код може стати причиною того, що робота сайтів/додатків/систем управління ресурсами гальмується та, відповідно, не задовольняє користувальницькі потреби, в той час як ручне виправлення займає надто багато часу та стає фінансово не вигідним для замовника. З метою подолання даної проблеми було створено покращений алгоритм візуалізації структур даних на клієнтському інтерфейсі з використанням технології JavaScript. Метод має перевагу у тому, що завантаження ресурсів відбувається як структурованих елементів в пам'яті, де вони розподіляються між режимами редагування та відтворення, уникаючи непотрібних перезавантажень.

Представлений алгоритм дозволяє миттєво оптимізувати написаний розробником код, автоматично зменшивши в рази кількість рядків, що значно вдосконалив і пришвидшить роботу сайту/додатку/системи.

### 2. Вимоги до програмної реалізації алгоритму

При програмній реалізації розробленого алгоритму у клієнтському пристрої встановлюється двигун для рендерингу HTML/CSS коду, що дозволяє пряме редагування вихідного коду та вмісту програмного додатку першим користувачем,

в той час як другий користувач безпосередньо редагує вихідний код за допомогою цифрового середовища на іншому дистанційному пристрої. Мають бути задоволені наступні вимоги:

- програмний продукт повинен функціонувати на персональних комп'ютерах із стандартним набором компонент;
- забезпечувати високу швидкість обробки даних та відклик користувачеві у реальному часі;
- забезпечувати зручність і простоту взаємодії з користувачем;
- забезпечувати можливість зручного масштабування
- створений інструмент має структурувати вихідний код розробника за рахунок використання паттерну MVVM;
- гарантувати можливість легкої розширюваності і підтримки коду - розділити всі елементи у відповідних модулях (наприклад, модель, маршрутизація), що дозволило б швидко додавати нові функції;
- забезпечити підвищення швидкості розробки.

### 3. Опис алгоритму

Крок 1: Обхід всіх директив та збереження в METADATA всіх залежних змінних.

Крок 2: Зв'язати вхідні та вихідні параметри всіх директив шляхом пошуку розроблених анотацій.

Крок 3: Заміна інтрполяцій в HTML коді.

Крок 4: Генерація класів на основі команд, які знаходяться в METADATA в анотації до кожної директиви.

Крок 5: Обхід METADATA та застосування обробника змін Change Detection.

Крок 6: Підтягування HTML/CSS Files за посиланням, вказаним у декораторі компоненти.

Особливістю програмної реалізації є те, що алгоритм розрахований переважно на SPA, які містять велику кількість скриптів та

HTML, тому розробка програмного забезпечення відбувається з урахуванням потреби в продуманій структуризації, бо в іншому випадку підтримка та масштабування стануть неможливими. Таким чином досягається максимальна ефективність описаного алгоритму.

### 4. Оцінка ефективності

Для демонстрації роботи алгоритму, базуючись на даних про основні функції, які повинен реалізувати програмний продукт та вимог до нього, - було визначено основні параметри, що будуть використані для розрахунку коефіцієнта технічного рівня (табл. 1).

Табл. 1 - Визначення коефіцієнтів значущості

Позначення параметра	Назва параметра	Одиниця виміру	Ранг параметра за оцінкою експерта							Сума рангів $R_i$	Відхилення	Квадрат відхилення
			1	2	3	4	5	6	7			
X1	Швидкодія мови програмування	Оп/мс	3	2	2	1	1	2	2	13	-4,5	20,25
X2	Об'єм пам'яті для збереження даних	Мб	1	1	1	2	2	1	1	9	-8,5	72,25
X3	Час обробки запитів користувача	Мс	2	3	3	3	3	3	4	21	3,5	12,25
X4	Потенційний об'єм програмного коду	кількість строк коду	4	4	4	4	4	4	3	27	9,5	20,25
	Разом		10	10	10	10	10	10	10	70	0	195

X1 – швидкодія мови програмування серверної технології;

X2 – об'єм пам'яті для збереження даних;

X3 – час обробки даних;

X4 – потенційний об'єм програмного коду.

В табл.1. наведено значимість кожного параметра, що визначається методом попарного порівняння. Оцінку проводить експертна комісія. Визначення коефіцієнтів значущості передбачає:

- визначення рівня значимості параметри: а) сума рангів кожного з

параметра шляхом присвоєння різних рангів;

- перевірку придатності експертних оцінок для подальшого використання;
- визначення оцінки попарного пріоритету параметрів;
- обробку результатів та визначення коефіцієнту значимості.

З метою перевірки ступеню достовірності експертних оцінок, визначимо наступні параметрів і загальна сума рангів:

$$R_i = \sum_{j=1}^N r_{ij} R_{ij} = \frac{Nn(n+1)}{2} = 70,$$

де  $N$  – число експертів,  $n$  – кількість параметрів;

б) середня сума рангів:

$$T = \frac{1}{n} R_{ij} = 17,5.$$

г) підсумок за сумою квадратів відхилення:

$$S = \sum_{i=1}^N \Delta_i^2 = 195.$$

д) коефіцієнт узгодженості:

$$W = \frac{12S}{N^2(n^3 - n)} = \frac{12 \cdot 195}{7^2(4^3 - 4)} = 0,8 > W_k = 0,67$$

Отже, проведене ранжування можна вважати достовірним, оскільки отриманий коефіцієнт узгодженості перевищує нормативний, який складає 0,67. Кожний окремий параметр розраховуємо по коефіцієнту вагомості  $K_{bi}$  за формулою:

$$K_{bi} = \frac{b_i}{\sum_{i=1}^n b_i}, \text{ де } b_i = \sum_{j=1}^N a_{ij}.$$

різниця значень коефіцієнтів вагомості не перевищує 2%.

## 5. Висновки

Основною метою розробки алгоритму стало створення автоматизованого інструменту для побудови динамічних веб - додатків мовою JavaScript задля підвищення продуктивності коду за допомогою сучасних методів оптимізації.

За допомогою даного програмного рішення можна розробляти як повноцінні сайти, так і функціональні модулі. Значно полегшується розробка SPA-додатків (Single Pages Applications), завдяки можливості здійснювати двостороннє зв'язування для динамічної зміни даних. Така система відображає чітку структуру програми та реалізується з використанням так званих «паттернів проектування», наприклад MVVM (Model-View-ViewModel).

Також було проведено дослідження розробленого програмного рішення з технічної сторони, а саме: було визначено його основні функції та сформовано множину варіантів для їх реалізації; на основі обчислених значень параметрів, а також експертних оцінок їх важливості було обчислено коефіцієнт технічного рівня, який і дав змогу довести оптимальність та доцільність розробки даного проекту.

## Список використаних джерел

1. Ітан Браун. Веб-розробка із застосуванням Node і Express. Повноцінне використання стека JavaScript = Web Development with Node and Express / Ітан Браун. - Санкт-Петербург: Пітер, 2017. - с. 336.
2. Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. Experimentation in software engineering. Springer, 2012.
3. Herrington J., Learning AngularJS. // Herrington J. - Packt Publishing, 2015 – p. 235.
4. Andreas B. Gizas, Sotiris P. Christodoulou and Theodore S., 2012. Comparative Evaluation of JavaScript Frameworks. Papatheodorou - Presentation April 16–20, 2012

УДК 004.94:519.876.5

ЯЦУК С. М.  
СТЕЦЕНКО І. В.

## АНАЛІЗ ДОРОЖНЬОГО РУХУ З ВИКОРИСТАННЯМ UNITY 3D

В даній статті розглянуто широко використовувані програмні продукти для моделювання транспортних потоків на мікрорівні та наведено їх порівняльний аналіз. Продемонстровано використання Unity3D, як альтернативи для моделювання тривимірного транспортного руху, а також його аналізу. Проведено аналіз Т-подібного перехрестя та графічно продемонстровано залежність утворення черги від різних параметрів та факторів впливу.

**Ключові слова:** імітаційне моделювання, транспортний рух, Unity3D, тривимірна візуалізація.

In this article considered widely used software products for modeling transport flows at the micro level and presented their comparative analysis. Demonstrated using of Unity3D as an alternative for three-dimensional traffic modeling and its analysis. Carried out an analysis of the T-intersection and graphically demonstrated the dependence of the queue on various parameters and factors of influence.

**Keywords:** simulation modeling, transport movement, Unity3D, three-dimensional visualization.

### 1. Вступ

Складність поведінки дорожнього трафіку і не можливість проведення експериментів із реальним потоком транспортних засобів в реальних умовах роблять імітаційне моделювання важливим інструментом для аналізу та організації дорожнього руху. Використання математичних моделей імітації дорожнього руху дає змогу досить детально моделювати ситуації реального масштабу. Проте дані математичні моделі не можуть задовольнити сучасні потреби імітаційного моделювання транспортного руху як складного комплексу засобів управління та учасників дорожнього руху.

В залежності від рівня деталізації моделі транспортного руху діляться на макроскопічні, мезоскопічні та мікроскопічні. Мікроскопічні моделі розглядають рух кожного транспортного засобу окремо. Даний підхід теоретично дозволяє описувати рух автомобілів більш точно у порівнянні із іншими моделями. Однак він вимагає великих обчислювальних ресурсів.

Існує низка програмних продуктів для моделювання дорожнього руху на мікроскопічному рівні, найбільш широко використовуванні із них DRACULA, PARAMICS, VISSIM [1]. Кожен із даних програмних продуктів має певний функціонал для аналізу дорожнього руху,

поєднання якого у одному програмному продукті є проблематичним.

### 2. Постановка задачі

Значне збільшення кількості транспортних засобів на дорогах призвело до перевантаження окремих ділянок транспортної мережі, що погіршує життєдіяльність міста. Таке перевантаження в свою чергу призводить до збільшення кількості дорожньо-транспортних пригод. Для ефективного вирішення транспортних проблем потрібно використовувати моделювання тривимірного транспортного руху, яке дозволяє аналізувати та оцінювати те, як побудова нових транспортних вузлів чи вдосконалення існуючих може вплинути на транспортну мережу в загальному. Моделювання дає можливість визначити такі параметри, як середня швидкість руху, інтенсивність руху, максимальна довжина черги та вплив довжини фази світлофору на транспортний потік. Це допомагає при виявленні вузьких місць транспортної мережі та можливих варіантів їх вирішення. Побудова чи вдосконалення дорожніх ділянок є ресурсовитратною задачею. Моделювання змін у транспортній мережі допоможе виявити неефективні рішення та уникнути нових транспортних проблем в подальшому з метою економії ресурсів, а також зниження витрат.

### 3. Огляд існуючих програмних продуктів

Серед найбільш відомих програмних продуктів для моделювання транспортного руху можна виділити три, які широко застосовуються на практиці, а саме DRACULA, PARAMICS, VISSIM. Далі розглянемо кожний з них більш детально.

DRACULA представляє собою пакет для мікроскопічного моделювання руху транспортних засобів у дорожній мережі. Його було розроблено в Університеті Лідса Великобританії в 1993 році. Даний програмний продукт дозволяє моделювати структуру дорожньої мережі для подальшого мікромоделювання руху транспорту по створеній мережі з врахуванням індивідуально прийнятих рішень водіями, їх досвіду водіння та особливостей руху транспортних засобів. DRACULA добре підходить для моделювання мінливої поведінки транспортних засобів в реальному часі. При цьому оцінюється вплив стратегій керування громадським транспортом, зміни руху потоку транспорту впродовж дня та із дня в день, надається оцінка ціноутворення заторів та дослідження стратегій, що спрямовані на зниження витрати палива та обсягу вихлопних газів. Поведінка транспортних засобів оцінюється з врахуванням впливу таких зовнішніх факторів, як погода і освітлення, а також впливу інцидентів, дорожніх робіт та аварій на певних ділянках мережі. Конфігурація станів транспортних засобів з використанням моделей зміни смуг руху і вибору дистанцій відбувається через дискретні інтервали часу в 1 секунду. Це є однією з кращих технік реалізації моделей для врахування наступних дій, наприклад, усунення заторів, впливу руху на послідовність поведінки водіїв, динамічного поширення заторів в дорожній мережі [2].

Пакет VISSIM також призначений для моделювання дорожньої мережі та транспортного руху на мікрорівні. VISSIM застосовується для аналізу, переосмислення з подальшою оптимізацією та перепроєктування міських і міжміських транспортних мереж. Альтернативні складні стратегії управління оцінюються з

використанням VISSIM до того, як вони фактично будуть впроваджені у реальному житті. Альтернативи представляються для того, щоб переконати осіб, що приймають рішення, в доцільності тієї чи іншої стратегії. Результати роботи VISSIM використовуються для визначення оптимальних стратегій управління транспортними засобами, перевірки різних макетів і розподілів смуг складних перехресть, перевірки місцезнаходження автобусних зупинок та стоянок транспорту тощо. Він переважно використовується технічними працівниками, які відповідають за управління сигналами на дорогах, планувальниками і дослідниками, які оцінюють вплив нових рішень. Транспортні засоби у VISSIM слідує один за одним в коливальному процесі. Коли швидкий автомобіль наближається до більш повільного транспортного засобу на одній смузі, то він сповільнюється. Початок реакції на сповільнення залежить від різниць швидкості, відстані і поведінки окремо взятого водія. Вони також намагаються знайти вільні місця для перебудови на сусідні смуги руху. Так само як і в моделі DRACULA, зміна станів транспортних засобів відбувається через проміжок часу в 1 секунду. В моделі враховуються і різні людські фактори, такі як зниження уваги та необхідний час на прийняття певного рішення водіями. VISSIM дозволяє збирати статистичні дані на будь-якій ділянці транспортної мережі та формувати відповідні звіти, створювати презентації та відеоролики [3].

PARAMICS – це набір програмних інструментів для моделювання дорожнього руху на мікрорівні, що призначений для імітації транспортних перехресть, розв'язок у населених пунктах, перевантажених автострад, а також оптимізації руху та регулювання маршрутів громадського транспорту, роботи світлофорів тощо. Кожний транспортний засіб моделюється з врахуванням всіх дрібних деталей протягом всього часу руху, забезпечуючи цим дуже точний потік трафіку, час проходження шляхів сполучень транспортними засобами, а також надає інформацію про перевантаження ділянок дорожньої мережі.

Крім заздалегідь визначених класів транспортних засобів, користувач може створювати свій власний транспортний засіб. Для кожного з них задано проміжок часу в 1 секунду, через який змінюється його стан в мережі та поведінка. Зміна смуги на дорозі виконується з урахуванням попередньої інформації щодо руху автомобіля. PARAMICS враховує вплив на транспортний рух регулюючих сигналів на дорогах, знаків зміни швидкості та повідомлень про проблеми мережі та пропозицій щодо зміни маршруту руху. Маршрутизація транспортних засобів задається певною мовою правил поведінки, яка визначена користувачем, що робить модель більш гнучкою та адаптивною. PARAMICS вирізняється серед аналогічних систем високопродуктивною візуалізацією процесу моделювання та масштабованістю програмного забезпечення, що дозволяє реалізовувати моделювання трафіку на дорожній мережі будь-якого розміру.

Програмне забезпечення PARAMICS постійно розвивається, щоб відповідати новим технологіям, що застосовуються в транспортних системах реального світу. В PARAMICS наявна статистика та різноманітна звітність стосовно аналізу транспортної мережі, презентації та відеоролики [4].

Для порівняння розглянутих вище програмних продуктів виділимо наступні значущі для аналізу транспортної системи параметри: погодні умови (1), паркувальні місця (2), не організовано припарковані автомобілі (3), пішоходи (3), громадський транспорт (4), автономні транспортні засоби (5), статичне керування маршрутами (6), динамічне керування маршрутами (7), адаптивна система регулювання швидкості (8). Функціональні можливості програмного забезпечення відмічені в таблиці 1 знаком "X".

**Таблиця 1. Порівняння функціональності пакетів моделювання**

Програма	1	2	3	4	5	6	7	8
DRACULA				X				
PARAMICS	X	X		X		X	X	
VISSIM			X	X				

#### **4. Аналіз транспортної системи з використанням Unity3D**

Мікроскопічне моделювання є потужним і універсальним інструментом для планування та аналізу дорожнього руху. Моделювання транспортного руху дозволяє визначати такі параметри як середня швидкість руху, оптимальна довжина фаз світлофору, інтенсивність руху, пропускна спроможність.

Аналіз даних, отриманих при моделюванні дорожньої ділянки, дає змогу оцінювати ефективність альтернативних стратегій керування транспортною мережею, вплив впровадження нових дорожніх вузлів на існуючу транспортну мережу. Це допомагає виявляти неефективні стратегії та підходи до вдосконалення транспортної системи до моменту їх фактичного впровадження у життя, уникати

транспортних проблем та економити ресурси.

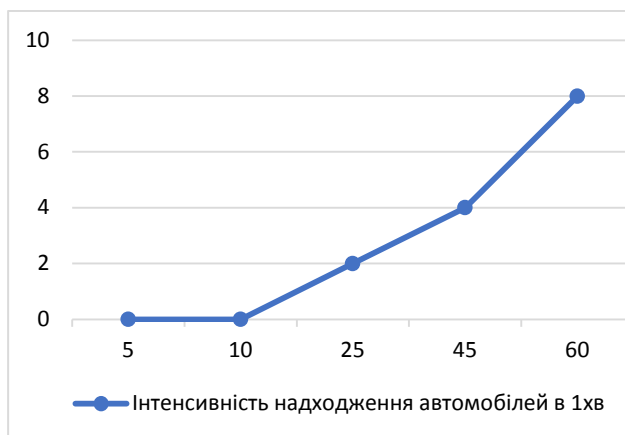
Тривимірна візуалізація є важливим аспектом моделювання дорожнього руху, так як дозволяє будувати транспортні системи максимально точно у відповідності з реальними умовами. Unity3D є потужним інструментом для тривимірної візуалізації даних, оскільки має широкий функціонал для роботи із тривимірною графікою та дозволяє швидко імпортувати 3D моделі, що були створені у різних 3D редакторах [5]. Даний інструмент містить у собі ряд реалізованих фізик та механік, що дає можливість швидко та якісно створювати транспортну мережу. До переваг Unity3D можна віднести підтримку високоякісних візуальних ефектів, можливість взаємодії між різними фізичними тілами, високу швидкодію, наявність редактора для роботи

із тривимірною графікою та гнучкість у роботі із 3D моделями. Сукупність даних переваг дає змогу будувати максимально наближені до реальних умов транспортні мережі.

Для проведення аналізу було обрано T-подібне перехрестя із двома смугами руху у кожному напрямку, яке зображено на рисунку 1. В результаті проведення моделювання із різною інтенсивністю надходження автомобілів до обраної ділянки транспортної мережі було отримано криву залежності довжини черги від інтенсивності руху, яка зображена на рисунку 2. Після перекриття по одній смузі

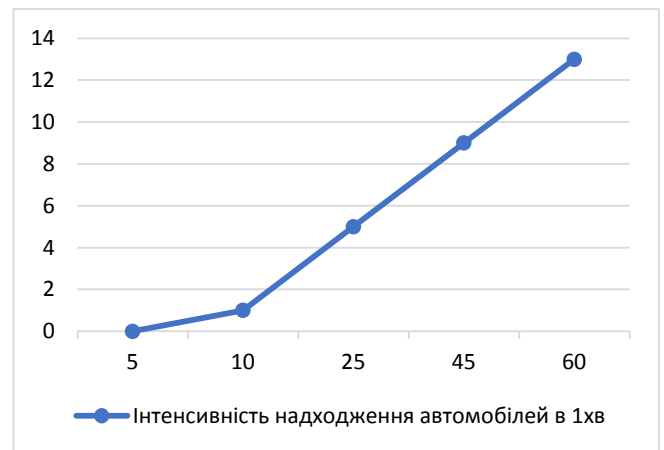


**Рисунок 1.** T-подібне перехрестя обране для аналізу

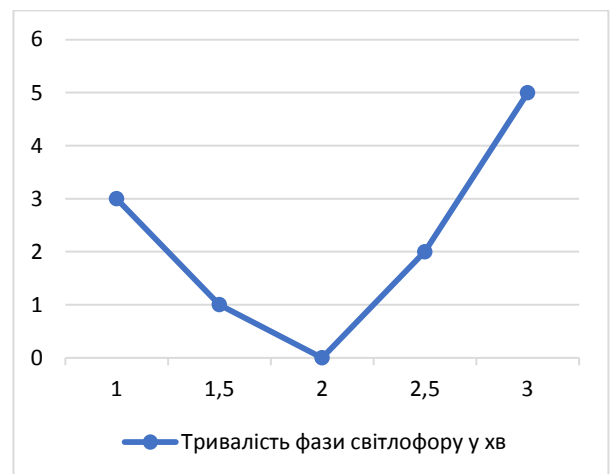


**Рисунок 2.** Графік залежності довжини черги від інтенсивності руху

руху у кожному напрямку було визначено залежність довжини черги від інтенсивності руху при умові, коли працює лише одна смуга руху. Відповідна цій ситуації крива залежності представлена на рисунку 3. При зміні тривалості довжини фази червоного світла світлофору на зелений було отримано залежність довжини черги від тривалості фази світлофору, що представлено кривою залежності на рисунку 4. На отриманих графіках можна чітко прослідкувати залежність довжини черги, яка утворюється на проїжджій частині, від різних параметрів та факторів впливу.



**Рисунок 3.** Графік залежності довжини черги від інтенсивності руху з однією смугою руху



**Рисунок 4.** Графік залежності довжини черги від тривалості фази світлофору



### **Висновок**

Результати проведення дослідження існуючих програмних рішень з моделювання транспортного руху показують, що існуючі імітаційні системи можуть оцінювати поточну ситуацію дорожнього руху і прогнозувати умови руху, але жоден із розглянутих програмних продуктів не включає у собі весь необхідний функціонал для аналізу транспортної мережі.

На даний момент відсутнє програмне забезпечення, що може задовольнити усі сучасні потреби у сфері моделювання транспортного руху, та яке б поєднувало у собі функціонал для оцінки основних значущих параметрів моделювання дорожнього руху, що були розглянуті вище, а також жоден із розглянутих інструментів не реалізує такі важливі функції, як автономні транспортні засоби та адаптивна система регулювання швидкості. Що стосується підвищення ефективності систем моделювання трафіку, то великим кроком вперед було б об'єднання у одному інструменті всіх перелічених вище функцій, а також реалізації ще однієї важливою функції, як динамічне керування дорогами (перекриття, обслуговування, ремонт) у реальному часі. Для реалізації нового інструменту моделювання дорожнього руху можна використати сучасний кросплатформний ігровий рушій Unity3D, який дає багато можливостей для реалізації інструментів такого роду. Вище на невеликому прикладі було продемонстровано деякі доступні можливості та отримані результати моделювання з використанням цього програмного засобу, проте це лише незначна частина того, що може бути зроблено у сфері тривимірного моделювання руху з використанням Unity3D.

### **Список літератури**

1. Review of Micro-Simulation Models [Електронний ресурс]. – <http://www.its.leeds.ac.uk/projects/smartest/append3d.html#a3>
2. Jaume Barcelo, Fundamentals of Traffic Simulation – Springer New York Dordrecht Heidelberg London, 2010. – 459 с.
3. VISSIM 5.30-05 User Manual – Planung Transport Verkehr AG 2011 – 680 с.
4. Mustapha Saidallah, Abdeslam El Fergougui, Abdelbaki Elbelrhiti Elalaoui A Comparative Study of Urban Road Traffic Simulators – MATEC Web of Conferences 81, 2016. – 6 с.
5. Unity3D документація [Електронний ресурс]. – <https://docs.unity3d.com/Manual/>

## ВИБІР АЛГОРИТМІЧНОГО ЗАБЕЗПЕЧЕННЯ ПІДТРИМКИ ПРОЦЕСУ СЕГМЕНТАЦІЇ КОРИСТУВАЧІВ СИСТЕМИ ТА ПОЗИЦІОНАННЯ ТОВАРІВ НА ПРИКЛАДІ КНИЖКОВОГО РИНКУ

Робота присвячена дослідженню алгоритмів, що використовуються рекомендаційними системами різного призначення, вивченню їхніх переваг та недоліків для застосування у системі формування пропозицій користувачам книгарні.

**КЛЮЧОВІ СЛОВА:** рекомендаційна система, колаборативна фільтрація, фільтрація вмісту.

The work is devoted to the research of algorithms, that are used by advisory systems of different purposes, and to the study of their advantages and disadvantages for use in the system of forming proposals to users of bookstores.

**KEYWORDS:** reference system, collaborative filtering, content-based filtering.

### 1. Вступ

Інформація є важливим компонентом людської діяльності. Сьогодні вона є цінним національним ресурсом, тому що може зберігатися, змінюватися та з користю використовуватися у процесі задоволення інформаційних потреб людини в електронному вигляді за допомогою різних систем. Та, не зважаючи на те, що ми живемо в період розвитку інформаційних технологій, велику кількість інформації продовжують використовувати у звичному колись паперовому вигляді. Це стосується в першу чергу бібліотек і книгарень.

Власники крамниць роздрібною торгівлі книгами й іншим інформаційним товаром зацікавлені в отриманні максимального прибутку з мінімальними затратами ресурсів. Автоматизація роботи таких організацій значно поліпшує їх діяльність, і, відповідно, прибуток. Тож питання впровадження підтримки роботи технічно-інформаційними засобами давно набуло популярності у цій сфері людської діяльності, а вдосконалення систем книгарень залишається актуальним завданням розробників.

Функції вивчення поведінки користувача системи та рекомендації на основі отриманих результатів найбільш актуального для користувача товару значно поліпшують роботу будь-якого підприємства, тому, незалежно від предметної області, велика увага при розробці систем та сайтів продажу

продукції приділяється зараз саме цим завданням.

### 2. Огляд досліджень та публікацій

З метою вивчення даної теми було розглянуто кілька робіт з використання методів та алгоритмів формування рекомендацій.

У роботі [1] описується актуальність вирішення проблеми збільшення обсягів даних, які зберігаються у мережі і пропонуються у користування, шляхом випередження запиту користувача через пропозицію йому потенційно необхідної інформації. Автор нагадує, що головною відмінністю алгоритмів систем надання рекомендацій від алгоритмів пошукових систем є знаходження відповіді без явного запиту користувача тоді як пошукові методи пропонують результат на конкретний запит.

Важливими завданнями [1] розвитку рекомендаційних систем сьогодні є:

- підвищення точності прогнозування рекомендацій,
- вирішення питання впливу розрідженості і розмірності матриці користувач-предмет на точність прогнозування рекомендацій,
- розв'язання проблеми нового користувача і предмету.

Перші дві проблеми можна вирішити методом зваженої суми та методом чіткої кластеризації відповідно, а остання задача лишається актуальною і саме для її вирішення автор праці [1] проводить

дослідження з розробки методів і засобів побудови рекомендаційних систем, котрі би містили в собі переваги існуючих алгоритмів і, крім того, підвищили б конверсію та лояльність користувачів системи для більш ефективної роботи з метою отримання кращого прибутку за рахунок якіснішого і швидшого задоволення потреб користувачів.

Проаналізувавши у цій праці [1] сучасний стан інформаційних технологій електронної комерції і методів побудови рекомендаційних систем, було проведено класифікацію існуючих підходів до побудови вищезгаданих систем та наведено переваги й недоліки кожного підходу, а також обґрунтовано гібридний підхід створення рекомендаційних систем як найбільш перспективний.

Також автор наводить формальну постановку задачі прогнозування персоналізованих оцінок предметів та на основі концепції асоціативних правил пропонує технологію надання рекомендацій новому користувачу, що враховує інтереси існуючих в системі користувачів і не вимагає реєстрації нового користувача в системі, використовуючи метод нечіткої кластеризації *c-means*.

Розробивши методіку тестування точності алгоритмів прогнозування рекомендацій, було запропоновано інформаційне та структуру програмного забезпечення дослідження і продемонстровано його результати.

Роботи [2-6] також присвячені вивченню методів створення рекомендацій користувачам, а саме алгоритму колаборативної фільтрації за сусідством користувачів [2] та формуванню групових рекомендацій, на основі яких запропоновано метрику оцінювання якості, та деяким алгоритмам контентної фільтрації [3].

У праці [4] автор розповідає про проблеми обслуговування читачів бібліотек і описує примітивний алгоритм з використанням архіву прочитаних книг для визначення вподобань читача з подальшою кластеризацією отриманих даних.

Автор роботи [5], як, власне, і автор роботи [2], запевняє, що забезпечення користувачів інформаційного ресурсу якісними рекомендаціями щодо його контенту можна досягнути, впровадивши рекомендаційну систему інтелектуального типу з використанням саме комбінованого, або гібридного, підходу, який враховуватиме загальні тенденції в розвитку інформаційного суспільства.

В роботі [6] теж основну увагу приділяють методу колаборативної фільтрації, розглянувши його переваги і недоліки, та використовує його для проектування власної рекомендаційної системи, описуючи вимоги до неї та структурну схему, яка зображує головні модулі.

У [7] було на прикладі розглянуто та реалізовано один із найпростіших методів колаборативної фільтрації за схожістю користувачів з використанням косинусної міри подібності, оскільки, за словами автора, серед списку можливих алгоритмів цей є найбільш поширеним у рекомендаційних системах поряд із методом коефіцієнта кореляції Танімото.

Стаття [8] описує формально задачу колаборативної фільтрації та аналізує основні проблеми, що виникають при розробці рекомендаційних систем за допомогою методів колаборативної фільтрації, а також пропонує підходи для їх вирішення, використовуючи кластерний аналіз і згладжування нейронною мережею. Автор зазначає, що таке поєднання технологій допоможе забезпечити високу якість рекомендацій.

Один із основних алгоритмів для роботи з великою кількістю даних – SVD досліджено у праці [9], де показано гнучкість і легкість застосування цього методу, але наголошено на необхідності його доопрацювання і вдосконалення, попри описані переваги.

Окрім дослідження типів рекомендаційних систем, у статті [10] описується система, створена на основі онтології із застосуванням додаткової семантичної підтримки та можливості здійснення логічного виведення, її функціонування, структура та реалізація.

Обґрунтувавши обраний метод створення рекомендаційної системи, визначено певні проблеми реалізації та перспективи подальшого дослідження.

Проблеми вдосконалення алгоритмів рекомендаційних систем також детально описуються у роботі [11], де для їх вирішення запропоновано модель системи, що належить до класу *item-based collaborative filtering systems*, що є відносно новим підходом. Дослідивши процес створення рекомендацій, автор пропонує реалізувати систему засобами JADE – фреймворку з відкритим кодом, написаному на Java. Але за словами самого ж автора ще існують варіанти вдосконалення й покращення запропонованої ним системи за рахунок використання інших, не менш цікавих, методів та засобів.

Стаття [12] аналізує результати порівняння рекомендаційних систем на основі коефіцієнту кореляції Танімото у порівнянні із алгоритмом «*item to item*» колаборативної фільтрації шляхом оцінки релевантності. За рахунок цього сформовано дані для досліджень у вигляді користувачів із унікальними ID. Алгоритм колаборативної фільтрації у даному випадку також заснований на косинусній мірі, яка представляє подібність предметів у вигляді косинуса між векторами покупок в матриці користувачів та предметів.

Автор статті [13] пропонує при розробці рекомендаційної системи використовувати метод індуктивного навчання, який буде нечітке дерево прийняття рішень, використовуючи алгоритм FuzzyID3. Описаний метод покликаний давати рекомендації подарунків з фактором впевненості в кожній рекомендації.

Але не вирішеними залишились два важливих питання:

- запропонований метод має суттєвий недолік, який полягає у підсиленні типових рекомендацій;
- при практичному застосуванні системи можна стикнутись із великою кількістю категорій подарунків, які варто було би об'єднати у деяку таксономію.

Тому автор пропонує модифікувати запропонований алгоритм.

Порівнюючи ж алгоритм формування рекомендацій на основі моделі прихованих факторів [14] з алгоритмом SVD за деякими критеріями, було зроблено висновок, що перший метод є кращим.

Хоча, автор [15] роботи, провівши власне дослідження алгоритмів рекомендаційних систем і вивчивши їхні недоліки та переваги, зупинився саме на методі SVD, оскільки той є гібридним алгоритмом. Але і тут виділяють таку проблему, як «холодний старт», коли без тестової вибірки нейронна мережа не зможе коректно передбачувати оцінки.

Досить детально вивчено методи колаборативної фільтрації на основі сусідства у праці [16], де було здійснено порівняння фільтрації за користувачами і за об'єктами. Вони визнані відносно простими в реалізації, але для виконання рекомендацій вимагають обробки всієї матриці оцінок, що є незручно при обслуговуванні користувачів у режимі реального часу, тому автор пропонує звернути увагу на деякі модифікації цих алгоритмів.

Про побудову асоціативних правил на основі визначеної множини спостережень йдеться в роботі [17], де запропоновано стохастичний підхід створення чисельних асоціативних правил, що враховує апріорну інформацію про значущість термів і ознак та використовує ймовірнісний підхід для перебору різних сполучень антецедентів і консеквентів асоціативних правил. Також було розроблено програмне забезпечення, що реалізує запропонований метод, та проведено експерименти з його дослідження при вирішенні практичних завдань.

### 3. Постановка проблеми

Як відомо, рекомендаційними системами є підклас систем фільтрації інформації, з допомогою яких можна визначити рейтинговий список деяких об'єктів, яким користувач надає перевагу, наприклад, фільмів, музики, тощо.

Серед різноманіття стратегій створення рекомендаційних систем виділяють дві наступні:

- content-based, або фільтрація вмісту;
- collaborative filtering, або

колаборативна фільтрація.

Крім того, останнім часом досить популярним явищем стало поєднувати методи обох вищезгаданих стратегій.

Кожна зі стратегій складається з множини алгоритмів та методів, котрі мають свої переваги і недоліки і можуть бути використані в залежності від конкретної ситуації та умов.

Отже, задача полягає в аналізі існуючого алгоритмічного забезпечення підтримки процесу сегментації споживачів та позиціонування товарів.

#### 4. Загальний опис

Наше дослідження має на меті ознайомитись із деякими прикладами алгоритмів рекомендаційних систем для оцінки й подальшого застосування їх у системі формування пропозицій користувачам книгарні.

Множина альтернатив, що розглядатимуться, буде складатись із семи елементів – алгоритмів:

- 1) Коефіцієнт кореляції Пірсона (колаборативна фільтрація);
- 2) Метод SVD (колаборативна фільтрація);
- 3) Метод найближчого сусіда (кластеризація);
- 4) K-means метод (кластеризація);
- 5) TF-IDF (контентна фільтрація);
- 6) Метод гібридної фільтрації;
- 7) Apriori (асоціативні правила).

Зазвичай алгоритм колаборативної фільтрації працює за наступним принципом: він обробляє інформацію, отриману від поведінки певної кількості людей, та знаходить окремі групи з подібними смаками.

Алгоритм аналізує вподобання, об'єднує їх та створює ранжований список інтересів.

Метод, що застосовує коефіцієнт кореляції Пірсона, використовується для вимірювання ступеня лінійної залежності між двома змінними. Коефіцієнт кореляції набуває значень від -1 до 1. Значення +1 означає, що смаки користувачів однакові. Після ранжування можна знаходити і

рекомендувати об'єкти, які сподобаються поточному користувачеві.

SVD – сингулярне розкладання матриці; є досить новим алгоритмом. Він призначений вирішувати типові проблеми методів колаборативної фільтрації.

Загалом підхід полягає у тому, що для того, щоб передбачити оцінку користувача для, наприклад, книги, береться деякий вектор, що містить набір параметрів користувача, та вектор параметрів книги. Їх скалярний добуток і буде необхідним передбаченням. Але даний метод має деякі недоліки, які пропонується вирішувати шляхом оптимізації, або машинного навчання.

У результаті роботи алгоритмів кластеризації виявляються ознаки, що характеризують групи об'єктів досліджуваного набору даних – класи. За цими ознаками новий об'єкт можна віднести до того або іншого класу.

На кожному кроці виконання методу найближчого сусіда до знайденої частини маршруту додається нове ребро. Алгоритм припиняє роботу, коли розв'язок знайдено, і не намагається його покращити, що є недоліком.

K-means метод будує задане число кластерів, розташованих якнайдалі один від одного.

Робота алгоритму ділиться на кілька етапів:

- 1) випадково вибрати k точок, які є початковими «центрами мас» кластерів;
- 2) віднести кожен об'єкт до кластеру з найближчим «центром мас»;
- 3) перерахувати «центри мас» кластерів відповідно до їх поточного складу;
- 4) якщо критерій зупинки алгоритму не задоволений, повернутися до п. 2.

Критерієм зупинки роботи алгоритму зазвичай вибирають мінімальну зміну середньоквадратичної помилки.

TF-IDF – статистичний показник, який застосовують для визначення важливості слів у контексті документа, котрий є частиною множини документів. Вага (значимість) слова пропорційна кількості вживань цього слова у документі, та

обернено пропорційна частоті використання цього слова в інших документах.

Гібридні підходи, поєднуючи колаборативну та контентну фільтрацію, підвищують ефективність (і складність) рекомендаційних систем.

Гібридний підхід може бути вигідним, коли використання колаборативної фільтрації відбувається на сильно розріджених даних (приклад холодного старту). Даний підхід дозволяє спочатку зважувати результати згідно контентної фільтрації, а потім зміщувати ці ваги у напрямку до колаборативної фільтрації.

За допомогою алгоритмів асоціативних правил можна отримати всі можливі правила виду “З А впливає Б” з різними значеннями підтримки та достовірності.

Алгоритм Apriori є поетапним та використовує стратегію пошуку в ширину. Його перевагою є властивість масштабованості. Він призначений для пошуку всіх множин ознак, що часто повторюються.

Етапи алгоритму:

- 1) формалізація даних;
- 2) пошук одноелементних наборів, що часто використовуються;
- 3) пошук  $k+1$ -елементних наборів, які часто трапляються;
- 4) побудова асоціативних правил.

Методи колаборативної фільтрації SVD та коефіцієнт кореляції Пірсона є не дуже складними в реалізації, бо не потребують значних ресурсів для цього. Але перший алгоритм, все ж таки, є простішим за другий. Якщо ж порівнювати їхню

ефективність, то вони обидва досить швидкодіючі.

Також метод коефіцієнту кореляції Пірсона вважають більш точнішим за результатами від методу SVD, хоча проблему «холодного» старту не вирішує жоден з двох алгоритмів, що є їх суттєвим недоліком. Але вони добре взаємодіють з іншими алгоритмами, тому дану проблему можна спробувати вирішити шляхом певних модифікацій.

Методи кластеризації теж досить легкі в реалізації, але мають меншу швидкодію. Можна виділити K-means метод як більш точніший за метод найближчого сусіда, крім того він непогано поєднується з іншими алгоритмами.

Методи контентної фільтрації, порівняно із методами колаборативної фільтрації, є складнішими в реалізації, та й ефективність роботи їхня менша.

Але TF-IDF метод дає досить точні результати і також може вирішити проблему «холодного» старту, оскільки алгоритм не потребує попередніх оцінок користувачів, яких може не бути.

Методи гібридної фільтрації є чи не найскладнішими в реалізації, але вони досить ефективні та точні.

Великим недоліком може стати ціна реалізації, особливо для стартаперів, які починають свій проект.

Асоціативні правила у реалізації та її ціні не складні, але їх точність, як і ефективність, невисока, та і питання «холодного» старту вони також не вирішують.

### Висновки

Отже, у даній роботі було здійснено ознайомлення із кількома стратегіями реалізації рекомендаційних систем, вивчено їхні переваги та недоліки за рядом деяких критеріїв, з допомогою яких ці стратегії порівнювались, для подальшого дослідження і застосування у проектованій системі формування пропозицій користувачам книгарні.

Наступним завданням наукової роботи буде детальніше дослідити відомі альтернативи методами теорії прийняття рішень.

### Список літератури:

1. Арцибасов В. Є. Методи та засоби побудови рекомендаційних систем для задач електронної комерції : автореферат дисертації на здобуття наукового ступеня кандидата технічних наук : 05.13.06 – інформаційні технології / Віталій Євгенович Арцибасов; Міністерство освіти і науки України, Національний університет «Львівська політехніка». –

Львів, 2015. – 28 с. – Бібліографія: с. 19-21, Електронний ресурс: <http://ena.lp.edu.ua:8080/handle/ntb/29589>

2. Метелиця М. А. Методи формування та оцінювання групових рекомендацій; Міжнародний науковий журнал Інтернаука №6. – Київ, 2016. – с. 64-67, Електронний ресурс: <https://elibrary.ru/item.asp?id=26469524>

3. Мелешко Є.В., Семенов С.Г., Хох В.Д. Дослідження методів побудови рекомендаційних систем в мережі Інтернет; Системи управління, навігації та зв'язку №1 (47). – 2018. – с. 131-136, Електронний ресурс: <https://doi.org/10.26906/SUNZ.2018.1.131>

4. Панарин Г. С. Можливості аналізу читацьких інтересів у сучасній системі автоматизації бібліотек; Репозитарій Харківського державного університету харчування та торгівлі. – 2017. – 6 с., Електронний ресурс: <http://elib.hduht.edu.ua/bitstream/123456789/2132/1/%D0%9F%D0%B0%D0%BD%D0%B0%D1%80%D0%B8%D0%BD.pdf>

5. Радчук О. В. Деякі питання розробки рекомендаційної системи для інформаційного сайту; SWorld. – 2013. – 4 с., Електронний ресурс: <https://www.sworld.com.ua/konfer33/1157.pdf>

6. Лесна Н.С. Методи пошуку та фільтрації інформації з використанням методу колаборативної фільтрації / Н.С. Лесна, С.М. Гайдамака // Системи обробки інформації. – 2013. – № 5(112). – с. 80-82, Електронний ресурс: <http://www.hups.mil.gov.ua/periodic-app/article/10870>

7. Савчук Т. О. Застосування кластерного аналізу для колаборативної фільтрації / Т. О. Савчук, А. В. Сакалюк // Вісник Хмельницького національного університету. Серія «Технічні науки». – 2011. – № 1. – с. 186-192, Електронний ресурс: <http://ir.lib.vntu.edu.ua/handle/123456789/17864>

8. Вараниця С. А., Білас О. Є. Гібридні рекомендаційні системи на основі медійних вподобань користувача; Міжнародний науковий журнал Інтернаука. – Київ. – 6 с., Електронний ресурс: <https://www.inter-nauka.com/uploads/public/1479375431886.pdf>

9. Жежерун О. П. Побудова рекомендаційних систем на основі онтологій / Жежерун О. П., Яремко С. А. // Наукові записки НаУКМА. Комп'ютерні науки. – 2017. – Т. 198. – с. 36-41, Електронний ресурс: <http://ekmair.ukma.edu.ua/handle/123456789/12536>

10. Глибовець М.М., Конюшенко О.В. Розробка агентної рекомендаційної системи віртуального університету засобами JADE; Наукова електронна бібліотека періодичних видань НАН України; Проблеми програмування. – 2015. – №1. – с. 104-115, Електронний ресурс: <http://dspace.nbuv.gov.ua/handle/123456789/113703>

11. Кучерук В.Ю., Глушко М.В. Покращення алгоритму «Item To Item» методу колаборативної фільтрації для розробки рекомендаційних систем на основі косинусної міри шляхом оцінки релевантності. – SCIENCEWISE №1 (42). – Харків, 2018. – с. 20-24, Електронний ресурс: <https://elibrary.ru/item.asp?id=32357205>

12. Молчановський О. І. Метод індуктивного навчання в основі рекомендаційної системи подарунків / О. І. Молчановський, В. П. Знахуренко // Вісник Національного технічного університету України "КПІ". Інформатика, управління та обчислювальна техніка. – 2013. – Вип. 58. – с. 61-67. – Електронний ресурс: [http://nbuv.gov.ua/UJRN/Vkpi\\_iuot\\_2013\\_58\\_10](http://nbuv.gov.ua/UJRN/Vkpi_iuot_2013_58_10)

13. Яковець М. В. Формування рекомендацій на основі моделі прихованих факторів; Міжнародний науковий журнал №6. – 2016. – с. 54-56.

14. Мазурік О. Ю. Покращення результатів роботи рекомендаційних систем за допомогою алгоритму SVD / О. Ю. Мазурік // International scientific journal. – 2015. – № 9. – с. 61-64. – Електронний ресурс: [http://nbuv.gov.ua/UJRN/mnj\\_2015\\_9\\_16](http://nbuv.gov.ua/UJRN/mnj_2015_9_16)

15. Олійник А.О. Побудова асоціативних правил на основі інтелектуального стохастичного пошуку / А.О. Олійник // Математичні машини і системи. – 2015. – № 4. – с. 45-56, Електронний ресурс: [http://nbuv.gov.ua/UJRN/MMS\\_2015\\_4\\_6](http://nbuv.gov.ua/UJRN/MMS_2015_4_6)

## ВЕБ-ЗАСТОСУВАННЯ ДЛЯ ОНЛАЙН-ОБРОБКИ ФОТОГРАФІЙ

Розглянута реалізація веб-застосування для онлайн-обробки фотографій та інших графічних зображень, яке підтримує можливість зміни їх розмірів з урахуванням контенту зображення та можливістю задання зон «недоторканості», що підвищує якість обробки таких зображень.

Reviewed the implementation of a web application for online photos and images processing, which supports possibility to change their dimensions according to the image content and the ability to specify areas of integrity, which improves the quality of processing such images.

### Постановка проблеми.

У сучасному суспільстві важливим засобом масової комунікації є різного роду соціальні мережі, які допомагають людям не тільки спілкуватися між собою, а й вирішувати певні бізнес-задачі. Невід'ємним функціоналом найпоширеніших сучасних соціальних мереж є публікація фотографій та обмін ними. При цьому актуальною проблемою є підвищення якості відображення пересланих зображень, що сприяє як візуальному їх сприйняттю звичайними користувачами мережі, так і є способом задовольнити ті чи інші потреби спеціалістів, зокрема, веб-дизайнерів, при вирішенні ними їхніх професійних задач.

Важливим аспектом даної проблеми є реалізація можливості зміни розмірів фотографій та інших графічних зображень відповідно до різних макетів без явних їх деформацій.

Найвні способи реалізації даного функціоналу не завжди задовольняють потребам користувачів. Зокрема, при стандартному масштабуванні має місце рівномірна зміна розмірів, що часто призводить до порушення пропорцій у відображенні окремих елементів зображення; при обрізці - може видалятися контент зображення, який бажано зберегти.

Окрім того, існуюче програмне забезпечення для вирішення даних задач має ряд супутніх недоліків, таких як:

- значна ресурсоємність,
- складність у використанні,
- комерційний характер,

– реалізація у вигляді десктопного або веб-застосунку, тоді як доступ до соціальних мереж здійснюється здебільшого через засоби стільникового зв'язку.

Тому актуальною є задача розробки веб-застосування для онлайн-обробки фотографій та інших графічних зображень, яке би підтримувало можливість зміни їх розмірів з урахуванням вмісту зображення.

### Огляд існуючих методів.

Існуючі підходи до вирішення даної задачі передбачають використання методу масштабування з урахуванням контенту (Content-aware scaling)[1, 2, 3], який ґрунтується на використанні алгоритму вирізки швів (Seam Carving) [1, 2]. Цей метод реалізований, зокрема, у графічному редакторі Adobe Photoshop[4].

Даний метод передбачає зміну розмірів зображення шляхом вирізання або вставки пікселів в різні частини зображення.

Процес вирізання або вставки пікселів ґрунтується на аналізі спеціальної енергетичної функції, яка визначає важливість тих чи інших пікселів з точки зору величини їх енергії - міри контрастності пікселя відносно сусідніх пікселів [1, 2].

Послідовність пікселів з низькою енергією, що перетинають зображення зверху вниз або зліва направо, вважають швом. Послідовно видаляючи або вставляючи шви, можна зменшити або збіль-



шити розмір зображення в обох напрямках [1, 2].

Вирізка саме швів гарантує, що при зменшенні зображення вилучається більше пікселів з низькою енергією і менше високоенергетичних. При збільшенні зображення вставка саме швів забезпечує баланс між вихідним вмістом зображення та штучно вставленими пікселями [1, 2].

Алгоритм вирізки швів може підтримувати декілька видів енергетичних функцій, таких як градієнтна величина, ентропія тощо [1, 2]. Зокрема, підрахунок енергії пікселя може здійснюватися відповідно (1) [1, 2]:

$$E = \left| \frac{\partial I}{\partial x} \vec{i} + \frac{\partial I}{\partial y} \vec{j} \right| = |\nabla I|, \quad (1)$$

де  $E$  – це енергія пікселя,  $I$  – інтенсивність його кольору.

Формування шва полягає у пошуку мінімального шляху зверху до низу (або зліва направо) у орієнтованому графі, вузли якого – це пікселі, а ребра направлені з поточного вузла у три сусідніх нижніх (правих) вузла, та мають вагу, рівну енергії пікселя, з якого ці ребра виходять. Визначений мінімальний шлях і буде швом пікселів з мінімальною сумарною енергією, що підлягає видаленню із зображення [1, 2].

Мінімальний шлях до поточного пікселя з координатами  $(x, y)$  може визначатися за динамічним алгоритмом відповідно до (2) [1, 2]:

$$M_{x,y} = E_{x,y} + \min \begin{cases} M_{x-1,y-1} \\ M_{x,y-1} \\ M_{x+1,y-1} \end{cases}, \quad (2)$$

де  $M_{x,y}$  – це мінімальний шлях до пікселя з координатами  $(x, y)$ ,  $E_{x,y}$  – енергія пікселя з координатами  $(x, y)$ .

Як показує практика, використання даного методу має значний недолік: після виконання послідовності вирізань швів на зображенні можуть з'явитися виражені шрами - лінійні контрастні переходи кольорів на об'єктах фотографії, що значно погіршують ефект застосування даного

методу. Їх енергетична оцінка кількісно виражається різким зростанням середньої енергії всіх пікселів зображення.

### Метод масштабування зображень.

Для удосконалення розглянутого методу масштабування зображень з урахуванням контенту пропонується інший підхід до визначення енергії пікселів та способу визначення шва - мінімального шляху для поточного вузла графа (пікселя).

Так, підрахунок енергій пікселів пропонується виконувати наступним чином (3):

$$E = \left| \frac{\partial I}{\partial x} \vec{i} + \frac{\partial I}{\partial y} \vec{j} \right| * e(\text{hog}(x, y)) + 1 \quad (3)$$

де функція  $\text{hog}()$  – це гистограма орієнтованих градієнтів в решітці пікселів  $11 \times 11$  навколо поточного пікселя із вісьма комірками, а функція  $e()$  – це ентропія розподілення вищевказаної гистограми.

Мінімальні шляхи (шви) пропонується підраховувати наступним чином (4):

$$M_{xy} = E_{xy} + \min \begin{cases} M_{x-1,y-1} * \sum \text{hog}_l(x, y) \\ M_{x,y-1} * \sum \text{hog}_c(x, y) \\ M_{x+1,y-1} * \sum \text{hog}_r(x, y) \end{cases}, \quad (4)$$

де функції  $\sum \text{hog}_l(x, y)$ ,  $\sum \text{hog}_c(x, y)$ ,  $\sum \text{hog}_r(x, y)$  - це сума значень для лівонаправлених, вертикально-направлених та правонаправлених комірок відповідно у шестикомірковій гистограмі орієнтованих градієнтів.

Сукупність двох вищевказаних підходів буде стимулювати підрахунок направленості мінімальних шляхів у напрям ребер об'єктів на фотографії, що зменшить кількість шрамів на зображенні і, відповідно, зменшить зростання середньої енергії його пікселів.

Для надання можливості при масштабуванні зображення даним методом вказувати зони «недоторканості», в цих зонах пропонується штучно завищувати енергію пікселів (5):

$$E \rightarrow \max(\{E_i\}), \quad (5)$$

де  $E$  – це енергія пікселя,  $\{E_i\}$  – усі можливі значення енергії пікселів.

Завдяки такому прийому шви (мінімальні шляхи), що будуть видалятися із зображення, не будуть проходити крізь зони, вказані користувачем.

### Реалізація веб-застосування.

Зазначений функціонал реалізований в рамках розробленого веб-застосування, який представляє собою клієнт-серверний додаток, архітектура якого представлена на рис. 1.

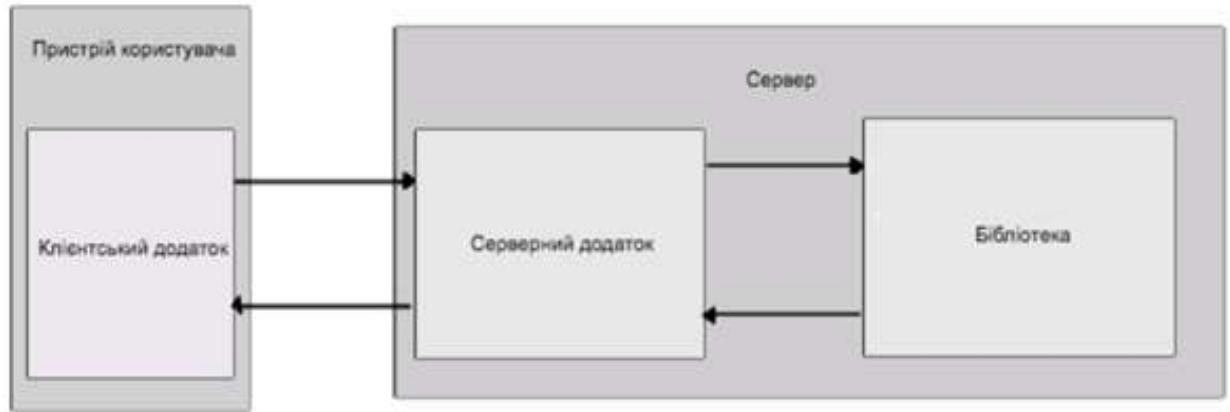


Рис. 1. Архітектура додатку

Клієнт представляє собою браузерний додаток, написаний на JavaScript, сервер – додаток, написаний з використанням Node.js. Структурно сервер складається з двох частин: модуля, що відповідає за обмін даними з клієнтською частиною, та бібліотеки, що реалізує основну функціональність і, зокрема, удосконалений алгоритм вирізки швів при масштабуванні зображень з урахуванням контенту.

Для реалізації функціоналу надання можливості користувачу вказувати зони «недоторканості», розроблений відповідний інструмент (рис. 2).

Результат роботи розробленого веб-застосування після виконання масштабування вихідного зображення з урахуванням контенту із застосуванням запропонованих підходів до реалізації алгоритму вирізки швів, представлений на рисунках 3 та 4.

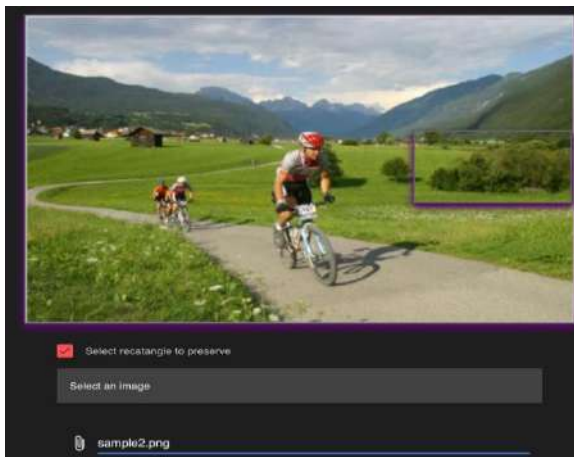


Рис. 2. Зображення із виділеною зоною цілісності



Рис. 3. Масштабування зображення без виділення зони «недоторканості»



Рис. 4. Масштабування зображення із виділенням зони «недоторканості»

Відмінність між даними рисунками полягає у застосуванні при масштабуванні вихідного зображення механізму задання зон «недоторканості»: рис. 3 є результатом масштабування без використання зон «недоторканості», рис. 4 - результат масштабування зображення, на якому була виділена зона «недоторканості», показана на рис. 2

Слід зазначити, що важливою вимогою для прискорення роботи розробленого застосування є необхідність обладнання сервера потужним апаратним забезпеченням.

### Висновки.

Розроблено веб-застосування для онлайн-обробки фотографій та інших графічних зображень, що підтримує можливість зміни їх розмірів з урахуванням контенту зображення та вказаних зон «недоторканості».

Даний функціонал ґрунтується на реалізації удосконаленого алгоритму вирізки швів, що підвищує якість масштабування таких зображень.

### Список літератури

1. Ariel Shamir, Shai Avidan. Seam Carving for Content-Aware Image Resizing. Електронний ресурс] – Режим доступу: <http://www.cs.toronto.edu/~kyros/courses/2530/papers/Lecture-08/Shamir2009.pdf>
2. Michael Rubinstein, Ariel Shamir, Shai Avidan. Improved Seam Carving for Video Retargeting. Електронний ресурс] – Режим доступу: <http://www.eng.tau.ac.il/~avidan/papers/vidret.pdf>
3. Z. Karni, D. Freedman, C. Gotsman. Energy-Based Image Deformation. Електронний ресурс] – Режим доступу: <https://core.ac.uk/download/pdf/21720496.pdf>
4. HOW TO USE CONTENT AWARE SCALING IN PHOTOSHOP CS6. Електронний ресурс] – Режим доступу: <https://www.dummies.com/software/adobe/photoshop/how-to-use-content-aware-scaling-in-photoshop-cs6/>

УДК 004.67

СЯСЬКИЙ Д.П.

ЖДАНОВА О.Г.

## ІНФОРМАЦІЙНА ПІДСИСТЕМА МУЛЬТИСПЕКТРАЛЬНОГО АНАЛІЗУ АЕРОФОТОЗНІМКІВ У СКЛАДІ СИСТЕМИ МОНІТОРИНГУ СТАНУ ПОСІВІВ

В даній статті розглянуто основні складові системи моніторингу стану посівів, перспективи та можливості її впровадження. Наведено опис практичного застосування методів кластеризації зображень, як однієї із задач інформаційної підсистеми мультиспектрального аналізу аерофотознімків. Описана формальна процедура оцінки якості кластеризованих зображень та наведено можливі переваги та недоліки використання різних алгоритмів, враховуючи специфіку вхідних даних.

**КЛЮЧОВІ СЛОВА:** АЕРОФОТОЗНІМОК, ОРТОФОТОПЛАН, НОРМАЛІЗОВАНИЙ ВІДНОСНИЙ ВЕГЕТАЦІЙНИЙ ІНДЕКС, КЛАСТЕРНИЙ АНАЛІЗ, DBSCAN, OPTICS

In this article the main components of the system of monitoring the state of crops, prospects and possibilities of its implementation are considered. The description of the practical application of image clustering methods as one of the tasks of the information subsystem of the multispectral analysis of aerial photography is given. The formal procedure for evaluating the quality of clustered images is described, and the possible advantages and disadvantages of using different algorithms are given, taking into account the specifics of the input data.

**KEYWORDS:** AERIAL PHOTOGRAPHY, ORPHOTOPLAN, NORMALIZED DIFFERENCE VEGETATION INDEX, CLUSTER ANALYSIS, DBSCAN, OPTICS

### 1. Вступ

Важливою задачею при веденні сільського господарства, а саме вирощуванні різноманітних агрокультур, є своєчасність виявлення різних шкідливих факторів, що ведуть до зниження продуктивності і навіть повного знищення посівів.

В наш час, використання безпілотних літальних апаратів (БПЛА) є відносно новим напрямком, але він вже встиг широко зарекомендувати себе. Існуючі засоби отримання та аналізу аерофотознімків полів є досить дорогими та мають багато недоліків. Ефективне вирішення поставленої задачі вимагає задоволення досить високих вимог до якості первинних знімків, швидкості і якості їх обробки. Крім того засоби, що використовуватимуться, повинні бути простими в експлуатації та обслуговуванні.

Обліт величезних площ посівів з використанням спеціалізованого обладнання закріпленого на корпусі апарату, дозволяє збирати значні об'єми інформації, за допомогою якої можна корегувати стратегію обробки та догляду рослин. Інформаційна технологія, яка об'єднує у собі засоби отримання аерофотознімків та технології їх обробки та аналізу, має великі перспективи розвитку та може допомогти в подоланні багатьох проблем.

### 2. Підсистема мультиспектрального аналізу аерофотознімків

Система моніторингу стану посівів складається з двох частин: підсистеми автопілотування безпілотного літального комплексу та інформаційної підсистеми мультиспектрального аналізу аерофотознімків. До основних задач першої з підсистем відноситься планування та моніторинг обльоту заданих територій та збір необхідних даних для подальшої обробки. Головними задачами підсистеми мультиспектрального аналізу аерофотознімків є створення ортофотоплану поля та проведення різноманітних обчислень для оцінки стану посівів, що і буде розглянуто в даній статті.

Найбільш відомими і розповсюдженим способом оцінки стану рослинності, що застосовується для даних отриманих за допомогою аерофотозйомки являється обчислення так званого *вегетатійного індексу* (англ. NDVI, Normalized Difference Vegetation Index) – нормалізованого відносного вегетатійного індексу фотосинтетично активної біомаси. Цей індекс розраховується за формулою:

$$I_{NDVI} = \frac{r_{NIR} - r_{RED}}{r_{NIR} + r_{RED}}, \quad (1)$$

де  $r_{NIR}$  – відображення в ближній інфрачервоній області спектра, а  $r_{RED}$  – відображення в червоній області спектра [1].

Згідно з (1), значення індексу в заданій точці зображення дорівнює різниці інтенсивності відбитого світла в червоному та інфрачервоному діапазонах, поділеній на суму їх інтенсивностей.

В основу даної формули покладено той факт, що висока фотосинтетична активність, як правило, пов'язана з густою рослинністю, що приводить до зменшення здатності об'єкта відбивати світло в червоній зоні спектру і до збільшення в ближній. Завдяки цьому з'являється можливість на основі аерофотознімків проводити картографування рослинного покриву, виявляти площі покриті і не покриті рослинністю, оцінювати щільність, стан, а за допомогою регулярного моніторингу і динаміку процесів розвитку рослин.

Згідно з (1), вегетаційний індекс може приймати значення від -1 до 1, проте для рослинності значення індексу не може бути менше 0. За допомогою інформації про певні характеристики об'єктів і характерні для них значення індексу, можна проводити їх розпізнавання та класифікацію на спектральних знімках.

Формула, що задає правило розрахунку індексу, надає можливість програмно сформувати гістограму значень, після чого користувач може визначити кольори, котрі будуть співставлені з відповідними діапазонами значень індексу.

Більш детально особливості індексу NDVI та його застосування для прогнозування урожайності описано в роботі [2].

### 3. Постановка задачі кластеризації зображень

Для оцінки областей та визначення регіонів посівів що вимагають подальшої обробки, необхідно об'єднати дані фрагменти в групи, що дозволить класифікувати їх та на основі отриманих даних приймати наступні рішення. Тому однією з задач підсистеми мульти-спектрального аналізу аерофотознімків є задача кластеризації елементів зображення орто-фотоплану з порохом індексом NDVI.

Кластер – група однорідних елементів, які характеризуються загальною властивістю [3]. Однорідність кластерів означає, що об'єкти, віднесені до одного кластеру, повинні бути схожі (близькі) щодо вибраної метрики. Об'єкти з різних кластерів повинні суттєво

відрізнятися. Кластерний аналіз являє собою розділ статистичного аналізу даних, який об'єднує методи розбиття (угруповання) безлічі об'єктів на групи, що називають кластерами. Термін «кластерний аналіз» був введений Робертом Трайном в 1939 році.

Нехай  $X = \{x_1, x_2, \dots, x_m\}$  – множина об'єктів, заданих значеннями в просторі ознак  $P = \{P_1, P_2, \dots, P_n\}$  (тобто кожен об'єкт  $x_i = (x_i^1, x_i^2, \dots, x_i^n)$ ) і задана функція відстані (метрика) між об'єктами  $\rho(x_i, x_j), x_i, x_j \in X$ .

Функцією кластеризації називається функція  $f: X \rightarrow Y$ , яка будь якому об'єкту  $x \in X$  ставить у однозначну відповідність номер  $y \in Y = \{1, \dots, k\}, k \leq m$ .

Множина кластерів  $C = \{C_1, C_2, \dots, C_k\}, k \leq m$ , являє собою розбиття множини об'єктів  $X$  таке, що кластер  $C_l = \{x \in X, f(x) = l\}; C_l \cap C_s = \emptyset, l \neq s, l, s = 1, \dots, k$ .

При чому для  $C$  справедливо те, що якщо  $x_i, x_j \in C_l$ , то  $\rho(x_i, x_j)$  мінімальне. Якщо  $x_i \in C_l, x_j \in C_s$ , то  $\rho(x_i, x_j)$  максимальне.

Тоді постановку задачі кластеризації даних можна сформулювати наступним чином:

Необхідно знайти таку функцію кластеризації  $f^*$ , що

$$Q(f^*, C, \rho) = \min_f Q(f, C, \rho), \quad (2)$$

де  $Q(f, C, \rho)$  – критерій якості кластеризації

Як вже було сказано вище, кожен об'єкт описується набором своїх характеристик, що називаються ознаками. Ознаки можуть бути наступних типів:

- бінарна ознака:  $P_i = \{0,1\}$ ;
- якісна(номінальна):  $P_i$  – скінченна множина;
- порядкова ознака:  $P_i$  – скінченна впорядкована множина;
- кількісна ознака:  $P_i \in R$  – множина дійсних чисел.

Найбільш розповсюдженою ситуацією є кластеризація об'єктів, у який всі ознаки кількісні, що і буде розглянуто далі.

### 4. Кластеризація по щільності

В даному випадку, під кластеризацією розуміється розподіл заданої множини точок (об'єктів) на підгрупи, кожна з яких, наскільки це можливо, однорідна.

Щільність точок для кожної точки множини  $X$ , визначається двома параметрами. Першим з них є  $\varepsilon$  – радіус «сусідства» (наближеності) точки  $x$ . Тоді множина  $M_\varepsilon(x)$  буде включати в себе такі точки  $f_i, (i = \overline{1, n})$ , для яких справедливо наступне

$$\rho(x, f_i) \leq \varepsilon, (i = \overline{1, n})$$

Функція  $\rho(x_1, x_2)$  визначає відстань між об'єктами  $x_1$  та  $x_2$  вибірки. Дана відстань може обчислюватись різнманітними способами, наприклад, як евклідова відстань.

Другим параметром визначення щільності точок є *мінімальна кількість точок* (МКТ), що розташовані найближче до даної точки відповідно до заданого радіуса  $\varepsilon$ , що і проілюстровано на рис.1.

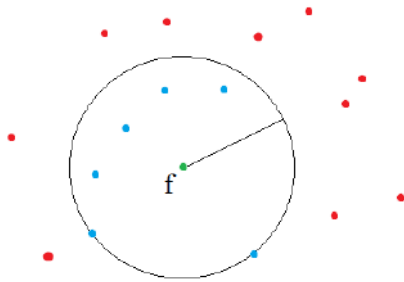


Рис. 1. Окіл точки при МКТ = 6

Точка  $x$  є прямо доступною по щільності від точки  $f$  (при відповідних  $\varepsilon$  та МКТ), якщо точка  $x \in M(x)$ , тобто точка  $x$  – це одна з точок  $f$  для другого околу(сусідства). На рис.2 наведено приклад доступності точки  $x$  по щільності від  $f$ , при МКТ = 5.

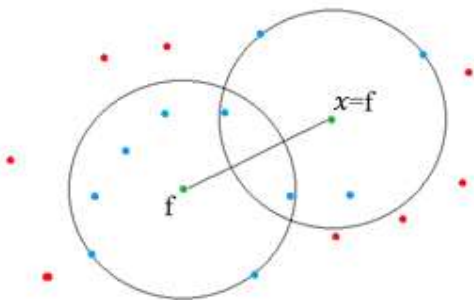


Рис. 2. Доступність точки  $x$  по щільності

### 5. Алгоритм DBSCAN

Алгоритм DBSCAN (Density Based Spatial Clustering of Applications with Noise) – щільнісний алгоритм для кластеризації просторових даних з присутністю шуму. Він оснований на припущенні, що щільність точок всередині кластерів більша, ніж поза кластерами, тому він дає змогу знаходити кластери довільної форми. Алгоритм запропонували Мартін Естер та Ганс-Пітер Кригель для вирішення проблеми розбиття даних (спочатку просторових) на кластери довільної форми [4].

Представити поняття кластерного упорядкування на основі щільності, можна

наступним чином: для постійної МКТ кластери на основі більшої щільності (тобто менше значення для  $\varepsilon$ ) об'єднуються в кластери з меншою щільністю (тобто більше значення для  $\varepsilon$ ). Цей факт проілюстрований на рис.3, де  $C_1$  та  $C_2$  кластери на основі щільності по відношенню до  $\varepsilon_2 < \varepsilon_1$  та  $C$  є щільнісним кластером щодо  $\varepsilon_1$ , що повністю містить множини  $C_1$  і  $C_2$ .

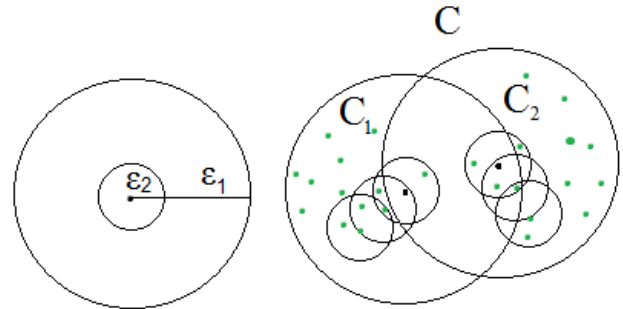


Рис. 3. Демонстрація вкладеності кластерів на основі щільності

Складність алгоритму DBSCAN оцінюється як  $O(N^2)$  у найгіршому випадку, хоча середня складність часу виконання може бути зведена до  $O(N \log N)$ , якщо розмірність даних не надто велика, а бінарні дерева використовуються для індексації даних. Для порівняння найгірша складність часу роботи алгоритму  $k$ -means  $O(INK)$ , де  $I$  - кількість ітерацій,  $N$  - кількість векторів даних, а  $K$  - кількість кластерів. Ієрархічно агломераційна кластеризація може бути оцінена як  $O(\tau N^2)$ , де  $\tau$  мале число, що відноситься до числа найближчих сусідів, які потребують оновлення після об'єднання кластерів.

DBSCAN має ряд переваг, а саме: алгоритм не чутливий до викидів, тобто в процесі кластеризації всі викиди виносяться в окремий кластер із заздалегідь заданою міткою; не вимагає попереднього задання кількості кластерів; використання даного методу дозволяє працювати з кластерами різної природи (форми).

Однак суттєвим недоліком даного алгоритму є досить трудомістка процедура визначення необхідних параметрів для коректної роботи. Проте, використання алгоритмів кластеризації заснованих на щільності на сьогоднішній день є досить ефективним і перспективним. Алгоритм DBSCAN став основою для цілого ряду ієрархічних методів кластеризації.

## 6. Алгоритм OPTICS

Алгоритм OPTICS (Ordering Points To Identify The Clustering Structure) - працює за схожим принципом, проте як більш розширений алгоритм DBSCAN для нескінченної кількості відстаней  $\varepsilon_i$ , які менші, ніж основний "радіус сусідства"  $\varepsilon$  (тобто  $0 \leq \varepsilon_i \leq \varepsilon$ ) [5]. Різниця полягає в тому, що членство кластерів не задається, замість цього зберігається порядок, в якому об'єкти обробляються і інформація, яка буде використовуватися розширеним алгоритмом DBSCAN для призначення членства в кластері (якщо це взагалі можливо для нескінченного числа параметрів). Ця інформація складається лише з двох значень для кожного об'єкта: основна відстань та відстань доступності.

Основна відстань околу  $p$  - це найменша відстань  $\varepsilon_i$  між  $p$  та об'єктом сусідом, таким що  $p$  буде основним об'єктом по відношенню до  $\varepsilon_i$ -сусідства, якщо даний сусід міститься в  $\varepsilon$  сусідстві об'єкта  $p$ . В іншому випадку основна відстань не визначена.

Далі наведено псевдокод алгоритму OPTICS [5]:

```
OPTICS (SetOfObjects,  $\varepsilon$ , MinPts, OrderedFile)
  OrderedFile.open();
  FOR i FROM 1 TO SetOfObjects.size DO
    Object := SetOfObjects.get(i);
    IF NOT Object.Processed THEN
      ExpandClusterOrder(SetOfObjects, Object,  $\varepsilon$ ,
        MinPts, OrderedFile)
  OrderedFile.close();
END: // OPTICS
```

Головна перевага даного підходу полягає в тому, що в порівнянні з іншими алгоритмами

кластеризації, немає обмеження в одному глобальному параметрі налаштувань. Натомість, кластерне сортування містить додаткову інформацію, еквівалентну щільності кластеризації, що відповідає широкому діапазону параметрів налаштування і, таким чином, є універсальною основою для автоматизованого та інтерактивного кластерного аналізу.

## 7. Практичне застосування кластеризації зображень

Після обчислення індексу NDVI та формування зображення на якому відображається значення стану рослинності по відношенню до заданої шкали, проводиться кластеризація. В результаті на вхід подаються зображення на яких більш детально можна виділити регіони що потребуються додаткового втручання та обробки. Враховуючи специфіку цих регіонів, а саме їх форму та розміри, агрономи можуть підібрати стратегію обробки, яка дозволить мінімізувати витрати та максимізувати ефективність. Далі наведено приклади кластеризації ортофотопланів полів.

На рис. 4 наведено ортофотоплан поля, з обрахований NDVI і виконана попередня обробка, а на рис. 5 наведено приклад результату кластеризації ортофотоплану за допомогою алгоритму DBSCAN. Як бачимо, після кластеризації, регіони що потребують додаткової обробки виділяються значно чіткіше, відсутні шуми що можуть бути спричинені різними перешкодами, зокрема хмарами та відбитим світлом.

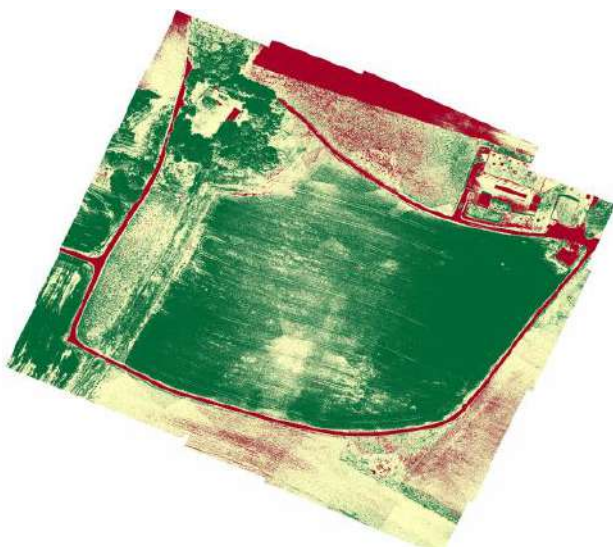


Рис. 4. Ортофотоплан з порахованим індексом NDVI

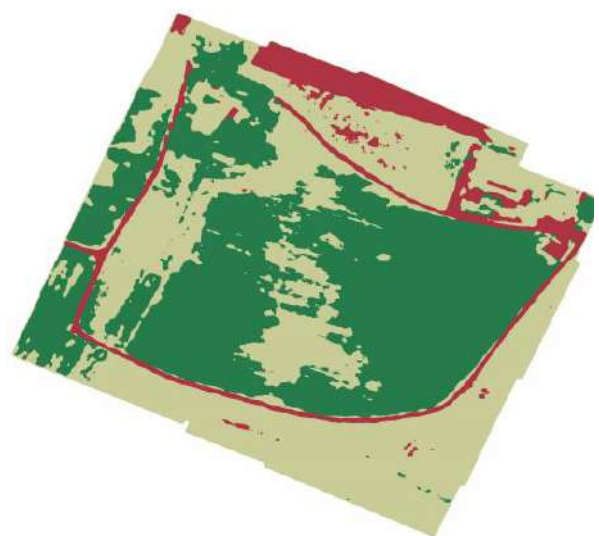


Рис. 5. Зображення, кластеризоване за допомогою DBSCA

### Висновки

В процесі роботи на системою було розглянуто основні алгоритми кластерного аналізу, описані їх переваги та недоліки для подальшого вибору, з урахуванням специфіки даних, що обробляються. Створена програмна реалізація алгоритмів DBSCAN та OPTICS, проведено порівняння їх роботи на даних різного об'єму та певних особливостей. Зібрана необхідна для аналізу інформація, а саме аерофотознімки з спеціальних камер, що дозволяють обчислювати NDVI індекс.

### Список літератури

1. Benedetti, R., & Rossini, P. (1993). On the use of NDVI profiles as a tool for agricultural statistics: The case study of wheat yield estimate and forecast in Emilia Romagna. *Remote Sensing of the Environment*, 45, 311-326.
2. Kogan F., Salazar L., Roytman L. Forecasting crop production using satellite-based vegetation health indices in Kansas, USA // *International Journal of Remote Sensing*. — 2012. — 33, N 9. — P. 2798–2814
3. Abonyi, J., & Feil, B. (2007). *Cluster analysis for data mining and system identification*. Boston, MA: Birkhäuser Basel.
4. Ester, Martin; Kriegel, Hans-Peter; Sander, Jörg; Xu, Xiaowei (1996). Simoudis, Evangelos; Han, Jiawei; ayyad, Usama M., eds. A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*. AAAI Press. pp. 226–231. ISBN 1-57735-004-9. CiteSeerX: 10.1.1.71.1980.
5. Ankerst, M. OPTICS: ordering points to identify the clustering structure / M. Ankerst, M. M. Breunig, H.-P. Kriegel, J. Sander // *Proc. 1999 ACM SIGMOD Intern. Conf. on Management of data*. - 1999. - P. 49 - 60.



УДК 683.519

*КОТЛЯР І. С.*

*ГРИША О. В.*

## **АНАЛІЗ ВИКОРИСТАННЯ СУЧАСНИХ АЛГОРИТМІВ МАСШТАБУВАННЯ**

Дана стаття містить опис сучасних засобів масштабування зображень: їх загальний опис, аналіз використання та порівняння їх характеристик (швидкість роботи, якість вихідного зображення). На прикладах показані результати роботи алгоритмів.

Ключові слова: масштабування, зображення, сучасне використання.

This article contains a description of modern image scaling tools: general description, analysis of the use and comparison of their characteristics (speed, quality of the final image). The examples show the results of the algorithms.

Keywords: scaling, image, modern use.

### **1. Введення**

Масштабування зображень – важливе питання сьогодні. Популярність соціальних мереж призвела до стрімкого зросту конкуренції в галузі обробки зображень, що сприяло появі нових виробників програмного забезпечення з обробки щойно зроблених фотографій. Конкуренція призвела до того, що для покращення швидкості роботи програми необхідно було розробляти власні алгоритми з масштабування. Тому в різних програмах можна отримати або різні результати, або однаковий результат але за різний час.

У сучасних розробках використовують векторні зображення (об'єктно-орієнтована графіка). У такому підході зображення представляється набором векторів, а у растровому – масив пікселів. Уся обробка виконується саме у растровому форматі, тому векторний файл містить інструкцію до відображення та утворення саме такого зображення. Тому векторне зображення легко переводиться в растрове але не навпаки.

Іншою галузю застосування алгоритмів масштабування растрових зображень є емулятори старих консольних ігор, багато з яких розраховані на показ динамічно змінюючихся зображень в реальному часі (при досить малій якості картинки на вході). Алгоритми працюють тільки при збільшенні в ціле число разів: 2x, 3x і 4x.

### **2. Характеристика сучасних засобів масштабування**

Для збільшення малокольорових зображень в малому розширенні найкраще підходять спеціальні алгоритми, розроблені для растрової графіки, що дозволяють при незначних спотвореннях точності промальовувати форми, зберігаючи і підкреслюючи чіткі контури і дрібні деталі. Також існують згладжують алгоритми, які підходять для обробки фотографій і кольорових растрових зображень з ефектом драбинки, що об'єднуються під назвою «методи сверхмасштаба».

Розмір зображення може бути змінений кількома способами. Існують два стандартних алгоритма масштабування зображень, білінейна і бікубічна інтерполяції. Колірні координати поточного пікселя в цих алгоритмах зазвичай обчислюються шляхом інтерполяції чотирьох сусідніх, то оброблене зображення виходить розмитим. Метод білінейної інтерполяції забезпечує деяке збереження гладкості контурів малюнка, але іноді викликає небажані ефекти згладжування деталей і все одно породжує досить помітний «ефект сходи». Більш досконалим методом є використання бікубічної інтерполяції.

Найпримітивнішим але досить швидким способом подвоєння роздільної здатності є метод копіювання найближчого пікселя. Для кожного пікселя кінцевого зображення вибирається один піксель

вихідного, найбільш близький до його положенню з урахуванням масштабування. Такий метод дає пікселізоване зображення при збільшенні і сильно зернисте зображення при зменшенні. Цей же спосіб може бути застосований для більш тонких змін дозволу, наприклад, для масштабу 99% або 101% відповідно прибирають або дублюють кожну соту точку.

Афінне перетворення - загальний метод для спотворення зображень. Воно дозволяє за одну операцію повернути, розтягнути і відобразити зображення. Тому в багатьох додатках і бібліотеках, що реалізують метод афінних перетворень, функція зміни зображень є просто обгорткою, яка розраховує коефіцієнти для перетворення. Принцип дії полягає в тому, що для кожної точки кінцевого зображення береться фіксований набір точок вихідного і інтерполюється відповідно до їх взаємним становищем і обраним фільтром. Кількість точок теж залежить від фільтра. Для білинейної інтерполяції береться 2x2 вихідних пікселя, для бікубічної 4x4. Такий метод дає гладке зображення при збільшенні, але при зменшенні результат дуже схожий на найближчого сусіда. Час роботи помітно більше, ніж у найближчого сусіда, і залежить від розміру кінцевого зображення і розміру вікна обраного фільтра. Від промахів кеша вже практично не залежить, тому що вихідні пікселі використовуються як мінімум по двох.

За допомогою алгоритма Суперсемплінг працює повноекранне згладжування в іграх. Його суть полягає у розбитті вихідного зображення на мережу пікселів кінцевого та складання всіх початкових пікселів, що припадають на кожний піксель кінцевого відповідно до площі, що потрапила під кінцевий піксель. При використанні цього методу для збільшення, на кожний піксель кінцевого зображення припадає рівно один піксель вихідного. Тому результат для збільшення збігається з методом найближчого сусіда. Продуктивність цього методу для зменшення нижче, ніж у афінних перетворень, тому що в розрахунок кінцевого зображення беруть участь всі пікселі вихідного.

Метод згортки схожий на афінні перетворення тим, що використовуються фільтри, але має не фіксоване вікно, а вікно, пропорційне масштабу. Наприклад, якщо розмір вікна фільтра дорівнює 6, а розмір зображення зменшується в 2,5 рази, то у формуванні кожного пікселя кінцевого зображення бере участь  $(2,5 * 6)^2 = 225$  пікселів, що набагато більше, ніж в разі суперсемплінга (від 9 до 16). Швидкість роботи цього методу залежить від усіх параметрів: розмірів вихідного зображення, розміру кінцевого зображення, розміру вікна фільтра. Саме цей метод реалізований в ImageMagick, GIMP, в поточній версії Pillow з прапором ANTIALIAS. Одна з переваг цього методу в тому, що фільтри можуть задаватися окремою функцією, ніяк не прив'язаною до реалізації методу. При цьому функція самого фільтра може бути досить складною без особливої втрати продуктивності, тому що коефіцієнти для всіх пікселів в одному стовпці і для всіх пікселів в одному рядку вважаються тільки один раз.

### 3. Характеристика засобів збільшення зображень

Алгоритм EPX/Scale2x/AdvMAME2x розшифровується як Eric's Pixel eXpansion, так як створений він був Еріком Джонтсоном. Цей алгоритм застосовувався при перенесенні двійка SRUMM з IBV PC на Macintosh, де роздільна здатність зображення була вдвічі більшим. Більш іноваційні Scale2x/AdvMAME2x було розроблені в 2001 році. Функціонально вони ідентичні, але останні мають більш ефективну реалізацію.

Алгоритми Scale3x/AdvMAME3x схожі на попередні, але для обчислення ім потрібно дев'ять сусідніх пікселів, тому на виході можна отримати зображення кращої якості. Звичайно, що обробка вхідних даних займе трішки більше часу і передбачає більше випадків.

Алгоритм Eagle подібний до метода сусідів але аналізує більше даних. За рахунок цього має підвищену якість вихідного зображення. Але алгоритм має певні недоліки, так, наприклад, чорний

піксель на білому фоні після використання алгоритму зникне. Цей недолік вирішений в наступних алгоритмах.

Алгоритм 2xSaI є удосконаленою версією Eagle, який був розроблений Дереком Лиану Кіе Фа для різних консольних емуляторів таких, як ZSNES, Jnes, DOSBox. Вони дозволяли запускати консольні ігри на ПК та емулювали роботу інших операційних систем та компіляторів для різних консольних мов програмування.

Алгоритм hqnx передбачає масштабування в пропорціях 2:1, 3:1 та 4:1. Колір кожного пікселя порівнюється з вісьмома сусідніми, сусіди позначаються як близькі й далекі, далі використовується прегенерірованна таблиця для відшукування необхідного співвідношення значень для кожного з 4, 9 або 16 вихідних пікселів. Алгоритм hq3x відмінно згладжує діагональні лінії з нахилом  $\pm 1:2$ ,  $\pm 1:1$  і  $\pm 2:1$  (за умови відсутності антиалиасинга на вході алгоритму); лінії з іншим

коефіцієнтом нахилу будуть представлені як ламані з вищезазначених діагональних. Добре згладжуються круті криві. Алгоритм дув розроблений для ігрової консолі Super Nintendo, ZSNES.

#### 4. Результати порівняльного аналізу

Для аналізу та реалістичного порівняння алгоритмів масштабування необхідно використовувати різні зображення з однаковою роздільною здатністю. Треба порівнювати наступні показники:

1. Час, який треба на зміну розміра зображення (в мілісекундах).

2. Якість вихідного зображення.

Для прикладу використовуються зображення розміром 1920x1080 та масштабувати їх в менший та більший розмір. Середнестатистичні показники масштабування кожного алгоритма наведений в таблиці 1.

*Табл. 1. Результати тестів на швидкість*

Назва алгоритму	Зменшення (ms)	Збільшення (ms)
Найближчий сусід	0.15	21
Афінні перетворення	7	851
Суперсемплінг	7	21.5
Згортки	19	115
Scale2x	-	187
Scale3x	-	68
Eagle	-	55
2xSaI	-	51
hqnx	-	60

Згідно результатів метод найближнього сусіда є найшвидким та найтростішив в реалізації, але вихідне зображення має дуже погану якість на відміну від інших.

Для порівняння якості зображення введемо умовний показник, де 1 - це

максимальна якість (вхідне зображення), 0 - уся картинка пікселізована, важко зрозуміти її зміст. Результати тесту за якістю вихідного зображення наведені в таблиці 2.

**Табл. 2. Результати тестів на якість**

Назва алгоритму	Показник якості
Найближчий сусід	0.3
Афінні перетворення	0.8
Суперсемплінг	0.75
Згортки	0.75
Scale2x	0.6
Scale3x	0.85
Eagle	0.4
2xSaI	0.8
hqnx	0.7

### Висновок

На сучасних комп'ютерах дані методи дозволяють реалізувати вивід масштабованого зображення, по швидкості прийнятний для додатків реального часу, зокрема, комп'ютерних ігор. Високооптимізовані алгоритми дають чітку і різку картинку з мінімальним розмиттям. Алгоритми масштабування піксельної графіки були реалізовані для безлічі емуляторів, 2D-ігрових движків і двигунів для ігор-переробок, наприклад, для AdvanceMAME, DOSBox, і ScummVM. В даний час такі фільтри використовуються в комерційних емуляторах Xbox Live, Virtual Console, і PSN для того, щоб класичні ігри низької роздільної здатності краще виглядали на дисплеях з високим показником роздільної здатності. Також, алгоритми масштабування мають великий попит в програмах редагування зображень, таких як Adobe Photoshop, GIMP, CoralDRAW та інші.

### Список використаної літератури

1. Thomas, Kas Fast Blit Strategies: A Mac Programmer's Guide – 1999. – С. 16
2. Hunter K., Filthy Pants: A Computer Blog - <http://filthypants.blogspot.com/2014/06/true-hq2x-shader-comparison-with-xbr.html>

УДК 004.93(015.7)

*ПАСТРЕЛО Н.*

*КОГАН А.В.*

## СПОСІБ ОРГАНІЗАЦІЇ БАГАТОШЛЯХОВОЇ МАРШРУТИЗАЦІЇ В SDN

У даній статті проаналізовано протоколи маршрутизації, що використовуються на даний час у класичній архітектурі мережі. Запропоновано новий алгоритм знаходження множини існуючих шляхів, що не перетинаються в мережах SDN. Проведений порівняльний аналіз між архітектурою SDN та класичною архітектурою.

The article analyzes the routing protocols currently used in the traditional network architecture. It is proposed a new algorithm for finding paths that do not intersect in SDN networks. A comparative analysis between SDN architecture and traditional architecture is presented.

Keywords: multipath routing algorithm, routing protocols, software-defined network, SDN.

### Традиційна архітектура vs. SDN архітектура

#### Вступ

Збільшення кількості і різноманітності пристроїв комп'ютерної мережі, розвиток різних технологій призвели до того, що процес контролю мережі таких розмірів стає практично не контрольованим. Обсяг мобільного трафіку зростає в геометричній прогресії, а види трафіку стають все більш різноманітними. За прогнозами компанії Cisco [1] в найближчі 5 років обсяг трафіку збільшиться в 4 рази, причому, мобільний трафік буде подвоюватися щороку. Сучасні комп'ютерні мережі складаються з окремих елементів, що виконують специфічні функції, наприклад: маршрутизатори, комутатори, NAT (Network Address Translation), брандмауери та інші. Технологія SDN (Software-Defined Networks) пропонує відмовитися від такої кількості проміжних пристроїв мережі та проводити централізовано управління мережею. З використанням нових підходів адаптованих під мережу SDN можна оптимізувати транспортні потоки, а також оптимізувати мережу для максимального використання маршрутів передачі даних.

Головна ідея SDN полягає у відділенні функцій передачі трафіку від функцій управління (включаючи контроль як самого трафіку, так і пристроїв, що здійснюють його передачу). У традиційних комутаторах і маршрутизаторах ці процеси невіддільні один від одного і реалізовані в одному пристрої. Для визначення маршруту передачі або недопущення зациклення трафіку пристрої, зазвичай, «спілкуються між собою», для чого розроблено безліч протоколів, таких як OSPF, BGP і Spanning Tree, але при цьому кожен з них функціонує досить автономно.

Згідно з концепцією SDN, вся логіка управління вноситься в SDN-контролери, які здатні відслідковувати роботу всієї мережі. Не можна сказати, що це революційно нова ідея, але вона дозволить:

- Полегшити управління мережею (OPEX).
- Зменшити витрати на обладнання (CAPEX).
- Збільшити гнучкість в програмуваності, відкритості, інновації.

В архітектурі SDN можна виділити три рівні (рис. 1):

- інфраструктурний рівень, що надає набір мережевих пристроїв (комутаторів і каналів передачі даних);
- рівень управління, що включає в себе мережеву операційну систему, яка забезпечує додаткам мережеві сервіси та програмний інтерфейс для управління мережевими пристроями і мережею;
- рівень мережевих додатків для гнучкого й ефективного управління мережею.



Рис. 1 Структура

Звісно, з перевагою централізованого рівня управління з'являється багато недоліків обмеження масштабованості. Наприклад, на сьогодні дата центрам необхідно підтримувати 150 мільйонів потоків даних на секунду, а контролер SDN може обробити лише частину [2]. Одним з можливих способів вирішення проблеми масштабування є використання архітектури Kandoo (використання локальних контролерів та одного центрального)

#### Аналіз протоколів маршрутизації в традиційних мережах

Відомі протоколи маршрутизації в традиційних мережах неефективні для використання в SDN. Однак, табличні протоколи маршрутизації мають явну перевагу, яке полягає в тому, що коли вузлу потрібно відправити пакет,

потрібний маршрут вже побудований в таблицях маршрутизації.

Цей факт є істотним для скорочення часу формування маршруту передачі даних, тому такі алгоритми можуть бути використані в якості базових алгоритмів для розробки протоколів маршрутизації для SDN. До таких табличних протоколів маршрутизації відносяться відомі протоколи, які широко використовуються в сучасних провідних мережах: протокол RIP (Routing Information Protocol) [3], заснований на дистанційно-векторному алгоритмі, протокол OSPF (Open Shortest Path First) [4], заснований на алгоритмі станом каналу.

Найбільш використовуваним протоколом внутрішньої маршрутизації IGP (*Interior Gateway Protocol*) є протокол RIP, але він поступово витісняється протоколом OSPF. До нових протоколів зовнішньої маршрутизації EGP (*Exterior Gateway Protocol*) відноситься протокол BGP і його нова версія - протокол міждоменої маршрутизації IGRP (*Enhanced Internet Gateway Routing Protocol*), модифікований для використання з OSI адресами замість IP.

Протокол RIP призначений для порівняно невеликих і однорідних мереж (алгоритм Белмана-Форда), що дозволяє маршрутизаторам динамічно оновлювати маршрутну інформацію, отримуючи її від сусідніх маршрутизаторів. До основних недоліків даного алгоритму можна віднести [4]:

- Потрібно багато часу на оновлення маршрутної інформації.
- Велика ймовірність появи петель маршрутизації.
- Обмежує розмір мережі.

Для вирішення завдання маршрутизації в традиційних мережах, крім OSPF, також використовуються протоколи IS-IS, BGP, які лавиноподібно поширюють інформацію для кожного маршрутизатора, що складається з повної інформації про топології мережі, метрики мережі по всіх каналах, з метою забезпечення обчислення найкоротшого шляху до адресата. До протоколу OSPF включені

нові типи оголошень для поширення мережею інформації про номінальну і не зарезервовану пропускну здатність кожного каналу зв'язку. Завдяки цьому маршрутизатор, відповідно до QoS, визначає шлях передачі трафіку.

У OSPF використовується мінімальний обсяг службової інформації для швидкого реагування на помилки за допомогою обходу точки збою. Крім того, для розрахунку будь-якого маршруту застосовується один показник (наприклад, число транзитних ділянок, час затримки).

### Модифікований протокол багатошляхової маршрутизації в SDN

В протоколах маршрутизації, що використовуються в класичній архітектурі, маршрутизатори мають інформацію про вузол тільки з сусідами. Тому для знаходження маршруту від вузла  $S$  до вузла  $F$ , кожен з протоколів починає алгоритм пошуку найкоротшого шляху з вузла  $S$  та закінчує при досягненні вузла  $F$ . Так як, розробляється протокол для SDN мережі, де контролер має повну інформацію про вузли, запропоновано починати алгоритм з обох вузлів  $S$ ,  $F$  одночасно.

Розглянемо контролер, що має топологію мережі у вигляді графу:

$$G=(V, E, B, T),$$

де  $V=\{v_1, v_2, \dots, v_n\}$  – множина вершин;

$E=\{e_1, e_2, \dots, e_n\}$  – множина дуг графу;

$B=\{b_1, b_2, \dots, b_n\}$  – множина значень пропускну здатності вершин;

$T=\{t_1, t_2, \dots, t_n\}$  – множина затримки каналів.

Для запропонованого протоколу

маршрутизації буде використовуватись метрика пропускну здатності каналу та його затримки. Для пошуку множини найкоротших шляхів використаємо модифікований алгоритм Дейкстри [5], що починається зі стартової та кінцевої вершини. Тоді алгоритм наступний:

1. Ініціалізація. Запам'ятовуються масив шляхів: найкоротші шляхи що починаються з вершини  $S$ . Відстань до всіх вершин графа  $\infty$ .

1.1. Запам'ятовуються масив шляхів: найкоротші шляхи що починаються з вершини

$F$ . Відстань до всіх вершин графа  $\infty$ .

2. Знаходимо таку вершину (із ще не оброблених), де поточна найкоротша відстань є мінімальною. Та запам'ятовуємо масив вершин з відстанню, для побудови не тільки одного шляху.

2.1. Повторюємо крок 2, але використовуючи масиви вершин що починаються з вершини  $F$ .

3. Обходимо всіх сусідів поточної вершин, і якщо шлях в сусідню вершину через іншу вершину менше за поточний для вершини, запам'ятовуємо новий коротший шлях до сусіда. Зробимо це для всіх вершин що є в масиву вершин найкоротших шляхів.

3.1. Повторюємо крок 3, але використовуючи масиви вершин що починаються з вершини  $F$ .

4. Порівнюємо поточні вершини з кроків 3 та 3.1. Якщо вершини не співпадають, повторюємо кроки починаючи з 2. Якщо вершини співпадають – крок 5.

5. Створюється кінцевий масив найкоротших маршрутів, об'єднуючи маршрути побудовані з вершини  $S$  та  $F$  (рис.2).

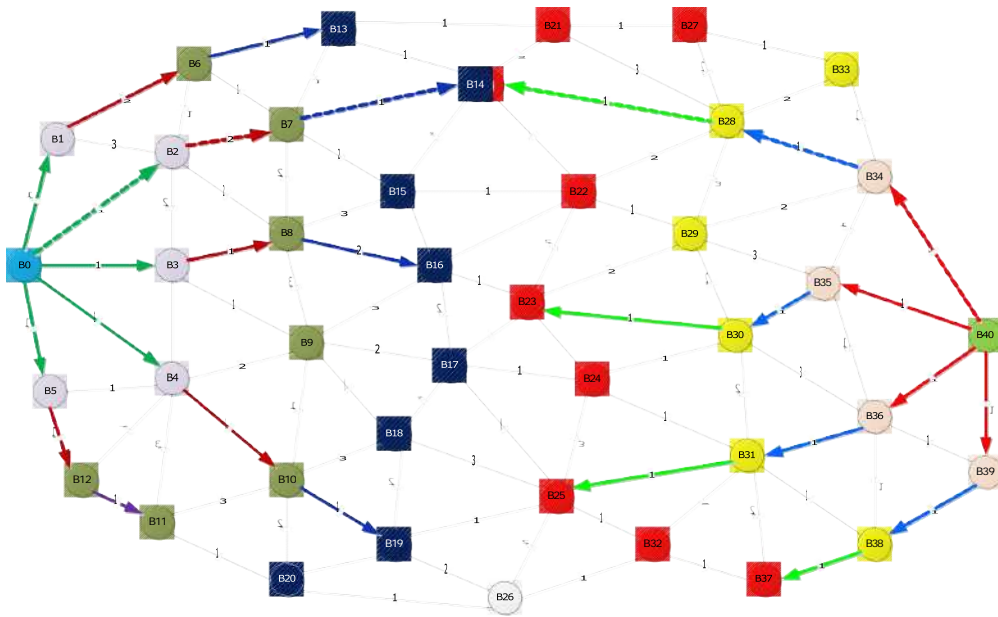


Рис. 2 Формування маршрутів

У таблицях представлені покрокове формування шляхів з двох вершин одночасно (Табл. 1,2)

Табл. 1. Формування шляхів від початкової вершини

$t_1$	$v_0-v_1$	$v_0-v_2$	$v_0-v_3$	$v_0-v_4$	$v_0-v_5$
$t_2$	$v_0-v_1-v_6$	$v_0-v_2-v_7$	$v_0-v_3-v_8$	$v_0-v_4-v_{10}$	$v_0-v_5-v_{12}$
$t_3$	$v_0-v_1-v_6-v_{13}$	$v_0-v_2-v_7-v_{14}$	$v_0-v_3-v_6-v_{16}$	$v_0-v_4-v_{10}-v_{19}$	$v_0-v_5-v_{12}-v_{11}$
$t_4$	$v_0-v_1-v_6-v_{13}-v_{21}$	-	$v_0-v_3-v_6-v_{16}-v_{23}$	$v_0-v_4-v_{10}-v_{19}-v_{25}$	$v_0-v_5-v_{12}-v_{11}-v_{20}$
$t_5$	$v_0-v_1-v_6-v_{13}-v_{21}-v_{27}$	-	-	-	$v_0-v_5-v_{12}-v_{11}-v_{20}-v_{26}$
$t_6$	$v_0-v_1-v_6-v_{13}-v_{21}-v_{27}-v_{33}$	-	-	-	-
$t_7$	Путь не сформирован!	-	-	-	-

Табл. 2. Формування шляхів від кінцевої вершини

$t_1$	$v_{40}-v_{39}$	$v_{40}-v_{36}$	$v_{40}-v_{35}$	$v_{40}-v_{34}$
$t_2$	$v_{40}-v_{39}-v_{38}$	$v_{40}-v_{36}-v_{31}$	$v_{40}-v_{35}-v_{30}$	$v_{40}-v_{34}-v_{28}$
$t_3$	$v_{40}-v_{39}-v_{38}-v_{37}$	$v_{40}-v_{36}-v_{31}-v_{25}$	$v_{40}-v_{35}-v_{30}-v_{23}$	$v_{40}-v_{34}-v_{28}-v_{14}$
$t_4$	$v_{40}-v_{39}-v_{38}-v_{37}-v_{32}$	-	-	-
$t_5$	$v_{40}-v_{39}-v_{38}-v_{37}-v_{32}-v_{26}$	-	-	-



## Висновки

Ефективність використання існуючих способів і алгоритмів маршрутизації в комп'ютерних мережах в більшості випадків залежить від структури і розмірності мережі. У зв'язку з великою розмірністю комп'ютерних мобільних мереж централізовані системи маршрутизації не забезпечують максимальної ефективності процедури маршрутизації. У сучасних комп'ютерних мережах технологія SDN дозволить полегшити та пришвидшити процес управління мережею, за рахунок розділення функцій передачі трафіку від функцій управління. Під час процедури реконфігурації мережі, з використанням технології SDN, при формуванні шляху обходу «проблемної» зони, маршрути будуть перебудовуватися з місця виникнення проблеми, а не з самого початку. Також в роботі запропонований модифікований спосіб формування множини найкоротших шляхів, що не перетинаються. В поєднанні з даною технологією, процедура формування маршрутів значно пришвидшиться.

## Список використаної літератури

1. [https://www.cisco.com/c/ru\\_ua/about/press/2016/11-17c.html](https://www.cisco.com/c/ru_ua/about/press/2016/11-17c.html)
2. V. Yazici, M. O. Sunay, and A. O. Ercan. Controlling a software-defined network via distributed controllers.
3. Столингс В. Современные компьютерные сети / Вильям Столингс. – СПб. : Питер, 2003. – 783 с. – (Серия «Классика computer science»).
4. Томас М. Структура и реализация сетей на основе протокола OSPF / М. Томас. – М. : Вильямс, 2004. – 816 с. – (Серия “Cisco Press”).
5. Кулаков Ю. А. Формирование множества непересекающихся путей в компьютерных сетях с применением алгоритма «обратной волны» / Кулаков Ю. А., Коган А. В., Диброва М. А., Чхаидзе Д. М. // Проблемы інформатизації та управління: збірник наукових праць. – К.: НАУ, 2015. – Вип. 4 (52). – С.68-73.

## **ДЕТЕКТУВАННЯ РУХІВ ЛЮДИНИ НА ВІДЕО ПОСЛІДОВНОСТІ В РЕАЛЬНОМУ ЧАСІ**

Представлено комплексний алгоритм розпізнавання рухів людини на відеопослідовності в реальному часі, який може працювати в режимі автономного розпізнавання і в режимі інтерактивного розпізнавання рухів людини. Запропоновано дворівневу архітектуру для комплексного алгоритму розпізнавання рухів людини, що містить на першому рівні кроки отримання послідовних кадрів з відеокамери, перед обробки отриманих кадрів, і виявлення людини на відеокадрі.

**КЛЮЧОВІ СЛОВА:** Детектування, розпізнавання рухів, відео послідовність, алгоритм, відеокамера, алгоритм Джонса-Віоли, ознаки Хаара, нейронна мережа.

A complex algorithm for recognizing human movements on a video sequence in real time is presented, which can work in the mode of autonomous recognition and in the mode of interactive recognition of human movements. A two-level architecture for a complex algorithm for recognizing human movements is proposed, which contains, at the first level, steps to obtain successive frames from the video camera, preliminary processing of the received frames, and identification of a person on the video frame.

**KEYWORDS:** Detection, motion recognition, video sequence, algorithm, video camera, Jones Viola algorithm, Haar signs, neural network.

### **1. Вступ та постановка проблеми дослідження**

В історії розвитку персонального комп'ютера спостерігалася еволюція користувальницького інтерфейсу в людино-машинній взаємодії (ЛМВ) від текстового командного інтерфейсу до графічного інтерфейсу, від простої клавіатури до миші, джойстика, електронної ручки, відео камери, і т.д. [1,] Пристрої ЛМВ стали більш зручними і природними для користувача. В даний час, з введенням нових понять, таких як «Віртуальне середовище – ВС», «людино-машинна інтелектуальна інтеракція – ЛМІ», «перцепційний призначений для користувача інтерфейс – ППІ » і т.д. потрібна розробка більш потужних і зручних способів взаємодії людини з комп'ютерною системою.

В якості одного із способів забезпечення комфортної взаємодії з комп'ютером, людська рука може бути використана в якості інтерфейсу введення [1, 2]. Жести є потужним каналом зв'язку, який формує основну частину передачі інформації в нашому повсякденному житті.

У порівнянні з традиційними пристроями ЛМВ, жести є менш нав'язливим, простим, більш зручними і природним способом взаємодії для користувачів. Проте, виразність жестів все ще залишається недостатньо вивченою для вирішення проблеми людино-машинної взаємодії.

В останні роки з'явилася і почала швидко розвиватися тенденція використання жестів, особливо жестів руки, як способу взаємодії з комп'ютерною системою. Розпізнавання жестів, таким чином, стало найважливішою частиною в ЛМІ і почало залучати безліч дослідників. Крім того, розроблені в ЛМІ технології також знаходять застосування в інших областях, таких як управління роботами, телеконференції, переклад мови жестів (для глухих), управління комп'ютерними іграми, і т.д.

### **2. Аналіз останніх досліджень і публікацій.**

На сьогодні, кількість наукових розробок, щодо теми дослідження є суттєвою. А.І. Томіліна та С.А. Савельєв [1] виконали порівняльний аналіз детекторів точкових особливостей для детектування і визначення

напрямку руху на відеопослідовності. Здійснено тестування швидкодії і якості реалізованих алгоритмів, проаналізовано вплив попередньої обробки на результат роботи алгоритмів.

К.В. Шепелев [2] розкрив точність детектування рухомих об'єктів на основі програмного забезпечення з відкритим кодом, що не перевищує 60%, це обумовлено існуючою проблемою, перший аспект полягає у відсутності алгоритмів і програмного забезпечення для імпортозаміщення, другий аспект полягає у високій вартості створення окремої системи відеоспостереження з функціями детектування і класифікації рухомих об'єктів.

Виявлення рухів об'єктів в послідовності зображень розкрили на сторінках своєї праці [3] Нгуєн В.Н. та Нгуєн Т.Т.

Із зарубіжних авторів варто наголосити на роботах Wu Y., Liu Q., and Huang T. [4], Yang J., Xu Y., Chen C. [5].

Однак, незважаючи на масштабність наукових досліджень, питання детектування рухів на відео послідовності залишається відкритим та потребує детального опрацювання.

### 3. Виклад основного матеріалу

Для опису динамічного компонента жестів потрібно створити відповідний класифікатор руху, що забезпечує можливість розпізнавання різних форм руху в реальному часі. Класифікатор руху повинен сприяти розпізнаванню сильно деформованих форм траєкторії (у порівнянні зі стандартними навчальними формами), тому що реальні рухи руки, що виконуються користувачами, можуть змінюватися і істотно відрізнятися від стандартних форм. Інші труднощі при розпізнаванні руху руки полягають в тому, що точки траєкторії, які спостерігаються при виконанні жесту дуже обмежені, особливо коли жест виконується з високою швидкістю за короткий час. У розглянутих роботах немає рішення проблеми розпізнавання руху. Алгоритм реалізації системи детектування рухів на відео послідовності, що пропонується у рамках даного дослідження наведено на рис. 1.

Перший рівень призначений для виявлення присутності руки в області видимості відеокамери і для ініціалізації роботи алгоритмів розпізнавання і трекінгу другого рівня. Звичайна відеокамера записує в одну секунду від 15 до 30 кадрів розміру 640x480 пікселів, і в кожен момент часу відеосистема доставляє тільки один кадр для обробки.

Система розпізнавання жестів в реальному часі повинна обробляти відеокадри в міру їх надходження з відеосистеми. Таким чином, час обробки кожного кадру має бути не більше 66 мілісекунд, щоб не було затримки в роботі системи.

Для скорочення часу обробки, розмір кадрів зменшується до 320x240 пікселів на кроці перед обробки. На цьому кроці також відбувається перетворення отриманих кольорових кадрів у напівтонові зображення, тому що надалі алгоритми обробки працюють тільки з напівтоновими зображеннями.

Детектор руки виконує пошук присутності руки на кожному відеокадрі. Якщо на відеокадрі не виявляється рука, то кадр скасовується і в обробку запускається наступний відеокадр з відеосистеми. Якщо рука присутня на кадрі, то детектор руки відключається. Метою кроку виявлення руки є ініціалізація роботи алгоритмів трекінгу другого рівня шляхом визначення прямокутної області кадру, в якій знаходиться рука. Дана область кадру буде використовуватися як модель об'єкта для алгоритму трекінгу.

Для розпізнавання автономних жестів, на другому рівні застосовується розпізнавання пози і пов'язаного з нею глобального руху. Після того як становище руки визначається за допомогою детектора руки (на першому рівні), прямокутна область, де знаходиться рука, зберігається і передається в алгоритм трекінгу, а сам детектор відключається. Алгоритм трекінгу аналізує цю область для створення моделі об'єкта і починає процес трекінгу. вступники відеокадри з відеосистеми потім передаються відразу на другий рівень.

Після попередньої обробки (зменшення розміру і перетворення кадру в півтонове зображення), механізм трекінгу запускається для пошуку місця розташування руки на новому відеокадрі. Алгоритм розпізнавання поз потім працює тільки з областю кадру, де знаходиться рука, а не з цілим кадром, і, таким чином, може забезпечити високу швидкість розпізнавання, незалежно від реального розміру відеокадру. Якщо жест не завершений, новий кадр пропускається на обробку, інакше алгоритм розпізнавання

руху запускається для розпізнавання отриманого жесту. Умовою завершення жесту приймається відсутність руки на кадрі, наприклад, коли рука рухається за межею області видимості камери і алгоритм трекінгу не може визначити місцеположення руки на кадрі. Розпізнаний жест потім може перетворюватися в команди для роботи з комп'ютерною системою. Даний режим може використовуватися в системі взаємодії на основі жестів для виконання команд, функції яких аналогічні гарячим клавішам.

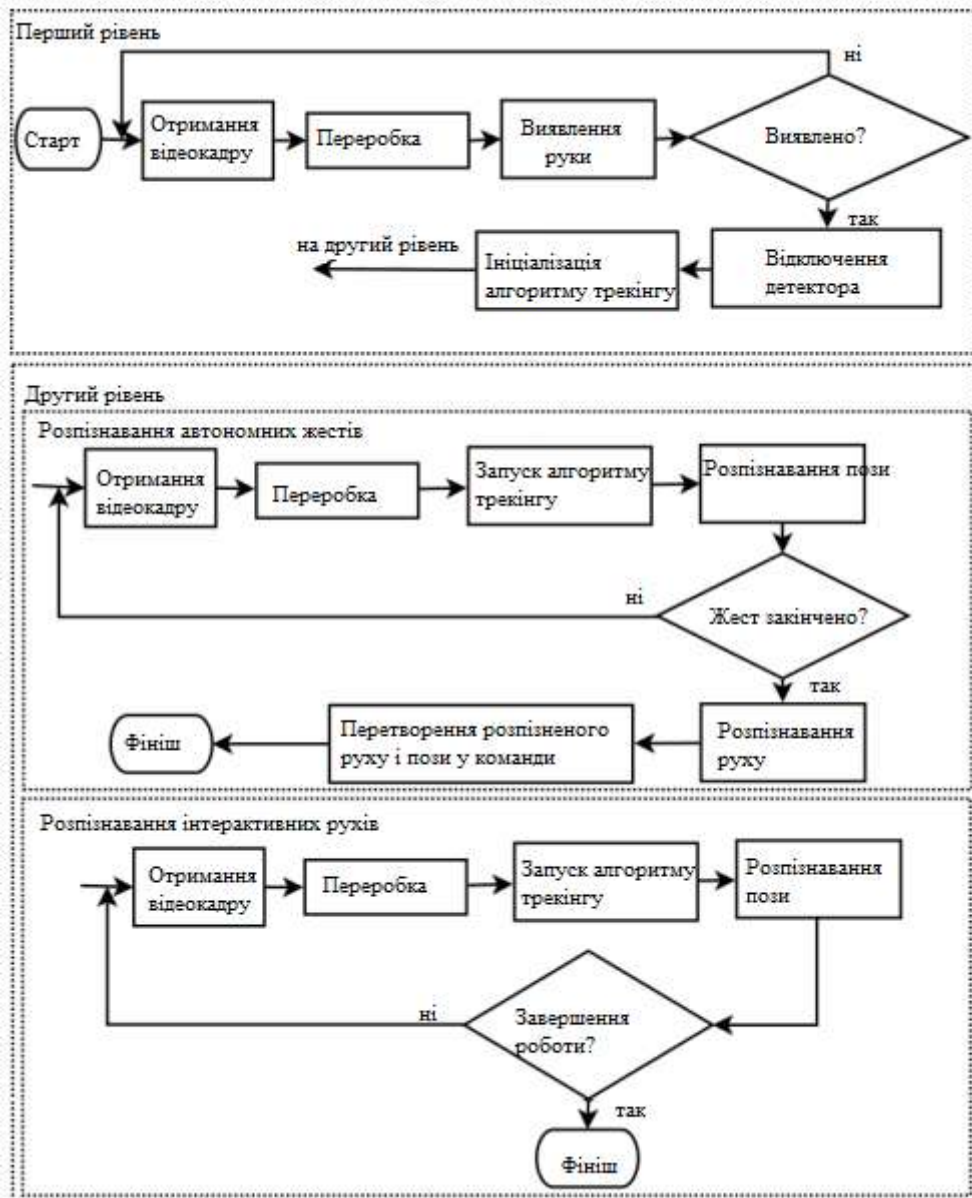


Рис. 1. Схема комплексного алгоритму розпізнавання жестів

Для розпізнавання інтерактивних жестів для прямої маніпуляції, на другому рівні застосовується розпізнавання пози і алгоритм трекінгу. При цьому трекінг також

відповідає за спостереження за становищем руки на кожного з них.

Розпізнавання пози дозволяє виконати команди, такі як клацання миші.

Даний режим розпізнавання призначається для безперервної роботи з комп'ютерною системою, такий як віддалене управління курсором миші, малювання за допомогою жестів, і т.д. Умовою завершення роботи є відсутність руки на кадрі, коли рука рухається за межею області видимості камери.

Виявлення руки на зображенні є первинним завданням системи розпізнавання жестів. В даний час, метод Джонса-Віоли з використанням ознак Хаара і каскадного AdaBoost класифікатора [6] вважається одним з кращих алгоритмів по співвідношенню ефективності розпізнавання до часу обробки для таких завдань [2, 4]. Метод Джонса-Віоли спочатку був розроблений для завдання виявлення людських обличчя на зображенні.

Детектор Джонса-Віоли також володіє вкрай низькою ймовірністю помилкового виявлення об'єкта і добре розпізнає об'єкт під невеликим кутом, приблизно до 15 градусів для руки і 30 градусів для особи. Таким чином, застосування методу Джонса-Віоли дає можливість виявлення руки на відеопослідовності в реальному часі.

Основні принципи, на яких базується метод, полягають в наступному:

- представлення зображення в «інтегральному» вигляді, що дозволяє швидко обчислювати ознаки Хаара;
- пошук потрібного об'єкта на основі ознак Хаара;
- використання каскадної класифікації AdaBoost, що дозволяє швидко відкидати вікна, де не знайдений об'єкт, і концентрувати обчислення для областей з найбільшою ймовірністю знаходження об'єкта.

Основна ідея запропонованого алгоритму розпізнавання рухів на відео послідовності складається в тому, що зображення розглядається як текстовий документ, в якому візуальні ознаки (характерні точки і області) зображення враховуються як слова, що утворюють даний документ. Клас документа визначається шляхом обчислення частоти появи деяких «ключових слів». Дана ідея була взята з популярної в обробці текстових документів моделі "bag-of-words" (BOW) [7].

Для застосування цієї ідеї в розпізнаванні об'єкта, характерні ознаки виділяються з набору зображень (бази даних для навчання), і потім поділяються на групи. У кожній групі вибирається один «представник» для всіх ознак групи. Обрані представники будуть служити «ключовими словами», з яких побудується так званий «словник ключових слів» (Безліч ключових слів). При зіставленні виділених ознак вхідного зображення з ключовими словами з словника ознак виходить гістограма ключових слів для даного зображення. Гістограма, що генерується служитиме дескриптором для розпізнавання у нейромережевому класифікаторі.

Алгоритм розпізнавання форми руки складається з наступних етапів:

1. Навчання класифікатора
  - а. Генерація словника візуальних ознак:
    - Виділення ознак методом SURF;
    - Кластеризація методом K-means;
    - Генерація словника з кластерів.
  - б. Створення дескрипторів і навчання нейронної мережі:
    - Зіставлення виділених ознак зі словником;
    - Навчання класифікатора (нейронної мережі).
2. Розпізнавання
  - Виділення ознак методом SURF;
  - Зіставлення виділених ознак зі словником;
  - Розпізнавання в нейронної мережі.

Початкове положення людини на відеокадрі визначається на етапі виявлення за допомогою детектора Джонса-Віоли. В процесі трекінгу, траєкторія руху записується для подальшої обробки в алгоритмі розпізнавання руху. Для вирішення цього завдання був розроблений швидкий алгоритм розпізнавання траєкторії руху руки на основі нейронної мережі.

Алгоритм розпізнавання траєкторії руху руки складається з наступних етапів:

Спрощення і згладжування траєкторії руху. В процесі трекінгу, траєкторія руху записується у вигляді масиву точок. Однак, при переміщенні людини, людина часто робить випадкові рухи малої амплітуди, що призводить до не "гладкої" траєкторії руху.

Таким чином, в масиві даних, отриманому після закінчення руху, завжди присутній якийсь шум. Крім того, кількість точок для кожного руху досить велика, що ускладнює обробку. Для вирішення цієї проблеми, застосовується алгоритм Рамер-Дугласа-Пекера (Ramer-Douglas-Peucker), що дозволяє "Згладити" траєкторію.

Передискретизація і перетворення траєкторії. Кількість точок в траєкторії руху людини не фіксована, а змінюється в залежності від швидкості переміщення і від швидкості обробки даних програмою. Механізм розпізнавання завжди вимагає, щоб число входів було фіксовано.

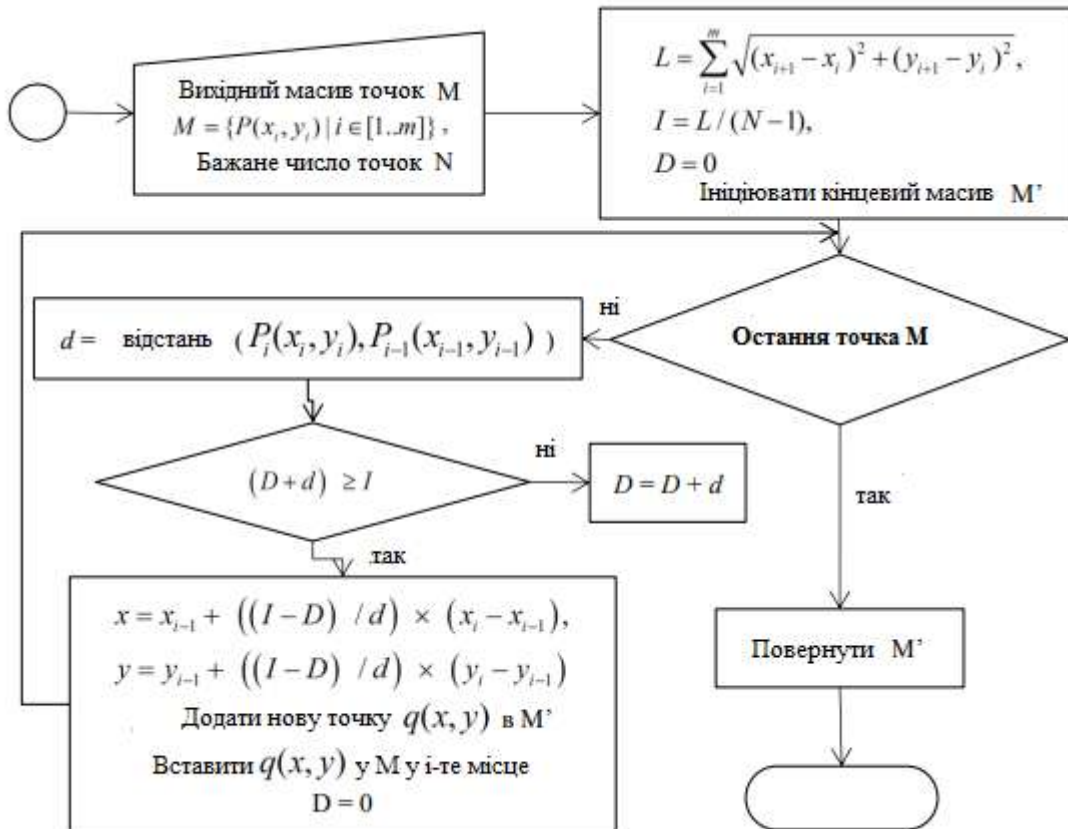


Рис. 2. Блок-схема процесу передискретизації і перетворення траєкторії

Повторна дискретизація (resampling) використовується для фіксування кількості точок траєкторії [8]. На даному етапі також виконується масштабування і переміщення траєкторії для підвищення ефективності дескрипторів, обчислених на наступному кроці.

Обчислення дескриптора. Масив точок траєкторії перетворюється в масив векторів нахилу і виконується обчислення синуса або

косинуса кутів нахилу. Масив синусів (або косинусів) надалі буде служити входом для класифікатора.

Навчання та розпізнавання в нейронній мережі. Для розпізнавання форми траєкторії руху людини в якості класифікатора застосовується багат шарова нейронна мережа зі зворотнім поширенням помилок.

#### 4. Висновки та перспективи подальших досліджень.

Представлений новий комплексний алгоритм розпізнавання рухів людини на відеопослідовності в реальному часі, який може працювати в режимі автономного розпізнавання і в режимі інтерактивного розпізнавання рухів людини. Запропоновано дворівневу архітектуру для комплексного алгоритму розпізнавання жестів, що містить на першому рівні кроки отримання послідовних кадрів з відеокамери, перед обробки отриманих

кадрів, і виявлення руки на відеокадрі. На другому рівні виконується стеження за рукою в часі, розпізнавання пози і розпізнавання глобального руху.

Запропоновано застосування алгоритму Джонса-Віоли для виявлення руки у відеопотоці з можливістю функціонування в реальному часі.

Алгоритм працює на основі ознак Хаара, інтегрального зображення, і каскадного AdaBoost класифікатора.

### *Література*

1. Томилина А.И., Савельев С.А. Анализ методов детектирования движения на видеопоследовательности на основе точечных особенностей // Актуальные проблемы авиации и космонавтики. 2016. – № 12. – С. 658-660
2. Шепелев К.В. Детектирование и классификация движущихся объектов в видеопоследовательности // Известия высших учебных заведений. Поволжский регион. Технические науки. 2017. – № 12. – С. 658-660
3. Нгуен В.Н., Нгуен Т.Т. Обнаружение движений объектов в последовательности изображений // Известия ТулГУ. Технические науки. 2017. – № 2. – С. 133-138
4. Wu Y., Liu Q., and Huang T. An adaptive self-organizing color segmentation algorithm with application to robust real-time human hand localization // Proceedings of the International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems (RATFG-RTS '99). – Washington, DC: IEEE Computer Society, 1999. –P. 161-166.
5. Yang J., Xu Y., Chen C. Gesture Interface: Modeling and Learning // IEEE International Conference on Robotics and Automation (San Diego, CA, 1994). – IEEE Computer Society, 1994. –V. 2. –P. 1747-1752.
6. Ryataeva A.V., Bandedev O. E. Video based flame detection algorithm // Сибирский журнал науки и технологий. 2017. – № 4. – С. 796-803.
7. Теплицкий Э.Г., Захаров С.М., Митрохин М.А. Метод обнаружения движущихся объектов на сложном динамическом фоне в оптическом диапазоне // Известия ВУЗов. Поволжский регион. Технические науки. 2016. – № 4 (40). – С. 16-26.
8. Афанасьев, Г.И. Особенности применения видеодетекторов движения в современных охранных системах видеонаблюдения [Текст] / Г.И. Афанасьев, Ю.В. Костенкова, С.А. Тоноян // Территория инноваций. – Энгельс, 2018. – № 4 (20). – С. 11-22

УДК 004.04

ЧЕРНУШЕВИЧ Д.О.,  
ОЛІЙНИК Ю.О.

## ПІДХІД ДО АНАЛІЗУ ПОВЕДІНКИ КОРИСТУВАЧІВ ВЕБ-САЙТІВ

Запропоновано розробку та реалізацію системи аналізу поведінки людини в веб-середовищі, що ґрунтується на методах й алгоритмах збору даних. Створене програмне рішення дозволить визначити основні тенденції в інтересах інтернет-користувачів, аналізувати їх на обраних проміжках часу та робити висновки.

The development and implementation of the system of human behavior analysis in a web environment based on data collection methods and algorithms is proposed. The created software solution will determine the main trends in the interests of Internet users, analyze them at selected intervals and draw conclusions.

Ключові слова: ВЕБ-АНАЛІТИКА, ВЕБ-СЕРЕДОВИЩЕ, КОЕФІЦІЄНТ ВЗАЄМОДІЇ, ТРАФІК, КОРИСТУВАЧ

### 1. Вступ

Веб-аналітика - це пряме відображення поведінки відвідувача на сайті. Отримуючи різні показники веб-аналітики, такі як кількість відвідувань та відвідувачів (окремих осіб), а також тривалість відвідування, можна створити основні показники ефективності (КРІ) - універсальну аналітичну модель, яка дозволяє визначати основні тенденції у запитах відвідувачів. Ця інформація дозволяє комерційним компаніям коригувати свої веб-сайти та бізнес-цілі, виявляючи області для вдосконалення, знаходити нові інструменти онлайн-просування сайту та тестування нових функціональних можливостей, тощо.

Представлений проект охоплює вивчення найпоширеніших показників, різні методи їх збору, аналіз ключових показників ефективності, а також створення, у вигляді програмного рішення, що засноване на математичній моделі, сучасного інструмента для веб-аналітики.

### 2. Вимоги до програмного забезпечення

Метою програмною реалізації є розробка системи аналізу поведінки людини в веб-середовищі, яка ґрунтується на методах й алгоритмах збору даних з веб-середовища.

В ході створення програмного продукту мають бути задоволені наступні вимоги:

- Візуалізувати обсяг трафіку і поведінкові показники по днях;
- Виводити класифікацію відвідувачів у формі звіту;

- Відобразити сегментацію відвідувачів в формі діаграм;
- Аналізувати цінність нового контенту;
- Відобразити зміни поведінкових показників;
- Класифікувати користувачів за допомогою логістичної регресії, пробіт-регресії або ROC-кривої;
- Прогнозувати фактично будь-які важливі показники;
- Задовольняти вироблені критерії оцінки результатів аналізу поведінки відвідувача веб-сайту;
- Порівнювати показники окремих сторінок одного сайту.
- Керувати рішеннями, які знижують ймовірність несприятливого результату та мінімізують можливі втрати бізнесу.

### 3. Опис алгоритму

Попередні дії

- Спроекувати структуру бази даних на MySQL з усією інформацією, яку необхідно аналізувати..
- Розробити серверний додаток в якості прокладки між клієнтською частиною та спроектованою базою даних.

Крок 1: Сегментувати трафік за активністю, залученістю і типу відвідувача. За зацікавленістю:

- незацікавлені (сеанси тривалістю до 60 секунд);
- цікавляться (сеанси тривалістю 60-180 секунд);



- зацікавлені (сеанси тривалістю понад 180 секунд).

З цією метою використовуємо модель Бокса-Дженкінса (рис. 1).

Крок 2: Налаштувати візуалізацію даних з можливістю різної фільтрації за допомогою графіків, що пов'язаний з базою MySQL. Всі звіти в даному файлі налаштовані на автоматичне оновлення раз на добу.

Особливістю програмної реалізації є те, що часто дії відвідувача є випадковими та нелогічними, і єдиним шляхом, який може надати точні дані про поведінку відвідувача, є шлях від однієї сторінки до сторінки, що безпосередньо слідує за нею. Іншими словами, єдиною сторінкою, яка впливає на поведінку відвідувачів на веб-сайті, є та, яку вони переглядають зараз. Цей метод аналізу стає все більш актуальним, тому що компаніям легше аналізувати дані шляху в контексті, не маючи посилання на весь сайт, щоб вивчити поведінку відвідувачі.

#### 4. Оцінка результатів та KPI

Для оцінки роботи програмного продукту як інструменту веб-аналітики та його відповідності вимогам, - було визначено розподіл Стюдента. Він ґрунтується на t-критерії Стюдента, який використовується з

метою обґрунтування середнього значення нормального розподілу і так само наближається до нормального розподілу в міру збільшення параметра.

Якщо розміри вибірок відрізняються незначно, застосовується спрощена формула наближених розрахунків:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

де  $\bar{X}_1$  та  $\bar{X}_2$  випадкові величини,  $n_1$  і  $n_2$  - кількість елементів вибірки, а

$$s^2 = \frac{\sum_{t=1}^n (X_t - \bar{X})^2}{n - 1}$$

Кількість ступенів свободи розраховується як:

$$df = \frac{(s_1^2/n_1 + s_2^2/n_2)^2}{(s_1^2/n_1)^2/(n_1 - 1) + (s_2^2/n_2)^2/(n_2 - 1)}$$

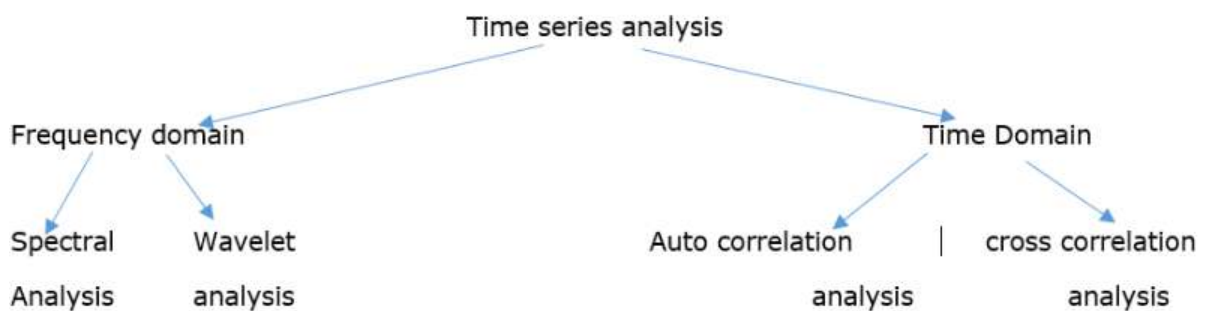


Рис. 1. Аналіз часових рядів за допомогою моделі Бокса-Дженкінса

Основні переваги t-тесту в тому, що він:

- Пристосований до роботи великими вибірками (немає обмеження на обсяг даних);
- Враховує розподіл і величину вибірки;
- Підходить для вимірювання різних параметрів (можна порівнювати і кількісні показники)

Визначення коефіцієнтів значущості передбачає:

- визначення рівня значимості параметра шляхом присвоєння різних рангів;
- перевірку придатності експертних оцінок для подальшого використання;
- визначення оцінки попарного пріоритету параметрів;

- обробку результатів та визначення коефіцієнту значимості.

При вступі користувачів в комунікацію в певної структури інтернет-ресурсу виділяється тематична інтернет-спільнота, де всім учасникам призначається унікальний ідентифікаційний номер. Таким чином, якщо розділити всю можливу активність користувачів по етапах, то надається можливим розробити спеціальний рейтинг активності користувачів інтернет-ресурсу.

Розглянемо етапи активності аудиторії:

Етап 0 - пасивний стан ( $r_0$ ): користувачі складаються в інтернет спільноті, але ніякої активності не проявляють, причому можливо повне ігнорування зовнішнього впливу на них;

Етап 1 - перегляд профілю ( $r_1$ ): активність користувачів обмежується тільки переглядом головної сторінки інтернет-ресурсу підприємства;

Етап 2 - споживання контенту ( $r_2$ ): користувачі здійснюють перегляд контенту інтернет-ресурсу, але не роблять більш складних дій;

Етап 3 - взаємодія з контентом ( $r_3$ ): користувачі здійснюють перегляд контенту, а так само користуються функціями односторонньої комунікації (коментарі та позначки про збіг інтересів);

Етап 4 - вчинення комунікації ( $r_4$ ):

спілкування з представниками промислового підприємства або з іншими користувачами інтернет-ресурсу;

Етап 5 - вчинення комерційної дії ( $r_5$ ).

У підсумку, користувачеві призначається той етап активності, який охарактеризований наявністю максимального типу активності.

Наприклад, якщо користувач споживав контент, мав комунікацію з представниками підприємства і зробив покупку продукції, то йому призначається 5 етап активності. Таким чином, сума часток етапів 4 ( $r_4$ ) та 5 ( $r_5$ ) буде характеризуватися як загальний показник активності аудиторії інтернет-ресурсу, що можна представити в математичному вигляді:

$$K_{\text{взаємод.}} = \frac{r_4 + r_5}{q_{\text{рес.}}},$$

де  $q$  – загальна кількість користувачів інтернет-ресурсу;

Підсумковий показник активності аудиторії являє собою коефіцієнт, який об'єктивно показує наскільки успішно йде залучення користувачів в вивчення контенту інтернет-ресурсу і в комунікацію з його представниками.

## 5. Висновки

Головною метою розробки алгоритму стало створення сучасної системи аналізу поведінки людини в веб-середовищі, яка ґрунтується на методах й алгоритмах збору даних з мережі Інтернет.

Математична модель використана для реалізації проекту ґрунтується методах статистики. За її допомогою ми визначаємо чи виходить отримана величина за межі області допустимих значень даної метрики (так званий довірчий інтервал), чи являється зміна критичною.

Створене програмне рішення дозволяє визначати основні тенденції в інтересах інтернет-користувачів, аналізувати їх на обраних проміжках часу та робити висновки про пріоритетність у соціальному житті населення (в т.ч. по регіонах), а для комерційних представників - розробляти бізнес-стратегії для просування сайтів та охоплення нової аудиторії, базуючись безпосередньо на числових даних отриманих у ході дослідження.

### Список використаних джерел

1. Clifton, B. (2012). *Advanced Web Metrics with Google Analytics* (3rd ed.). Indianapolis, IN: John Wiley & Sons., p. 11-16.
2. Lovett, J. (2009). *US Web Analytics Forecast, 2008 To 2014*. Cambridge, MA: Forrester Research.
3. Hu, X., & Cercone, N. (2004). A Data Warehouse/Online Analytic Processing Framework for Web Usage Mining and Business Intelligence Reporting. *International Journal of Intelligent Systems*, 19(7), p. 585–606.

УДК 004.92+911

*ДРЕГАЛО Л. В.  
ТЄЛИШЕВА Т. О.*

## **ОГЛЯД МОДЕЛЕЙ АНАЛІЗУ ТРАНСПОРТНОЇ ДОСТУПНОСТІ РАЙОНІВ МІСТА**

У цій статі розглянуто визначення транспортної доступності, на основі якого описані декілька моделей для її аналізу. Продемонстроване наочне порівняння оцінки транспортної доступності за допомогою моделі на основі відстаней та ізохронної моделі. Визначені їх переваги та недоліки з точки зору практичного використання.

### **ДОСТУПНІСТЬ, ІЗОХРОН, ТРАНСПОРТ, МОДЕЛЮВАННЯ**

This article addresses the definition of transport accessibility and describes models for its analysis based upon this definition. Distance-based and isochrones-based models are overviewed. The illustrated comparison of these models with their pros and cons is evaluated from a practical point of view.

### **ACCESSIBILITY, ISOCHRON, TRANSPORT, MODELLING**

#### **1. Вступ**

У цій статі розглянуто різні моделі аналізу транспортної доступності районів міста які використовують в сучасних інформаційних технологіях. Застосування сучасних інструментів таких як Open Street Map, Google Maps Traffic дозволяє нам побудувати складні моделі пересування людей в просторі із урахуванням багатьох факторів, таких як розклад громадського транспорту чи обсяг трафіку на дорогах.

#### **2. Визначення транспортної доступності**

Транспортна доступність – важливий показник для містобудівників. Він показує, наскільки жителі мікрорайону, району, міста охоплені різноманітними послугами, робочими та навчальними місцями, можливостями для відпочинку та розвитку. Доступність визначається атрибутами як моделей поведінки, так і транспортної системи в цілому. Просторовий розподіл видів діяльності визначається моделями розподілу земельних ресурсів, їх якостями та атрибутами, та напряду впливає на доступність. Часові та грошові витрати на переміщення також є важливими складовими доступності. Хоча більшість дослідників погоджуються на таких загальних властивостях доступності, вони розробили широкий спектр способів її вимірювання. [1]

Найпростіші способи вимірювання доступності - кількісний підрахунок

можливостей. Такі способи підраховують загальну кількість можливостей (тобто робочих місць, магазинів, розважальних закладів тощо), які знаходяться на певній відстані від району. Ця міра дає змогу оцінити, скільки різноманітних можливостей є у жителів певного району. Що більше таких можливостей, то більшою є транспортна доступність району. Гравітаційні способи вимірювання ставлять різним місцям призначення різні ваги в залежності від типу зайнятості у тому місці, а також вартості, часу або відстані що треба подолати, щоб туди дістатись. Загалом ці, та інші варіанти вимірювання пропонують різні компроміси між складністю побудови транспортної моделі та легкістю сприйняття кінцевого результату. Складніші моделі потребують більш складних і повних наборів даних.

Метою будь-якої транспортної системи є не процес переміщення як такий, а доступ до різноманітних об'єктів інфраструктури. Попит на транспорт - це попит, що походить від потреби подорожуючих на товари і послуги. Отже, для оцінки транспортної системи потрібно ввести міру її ефективності в доставці людей до об'єктів інфраструктури, якими вони бажають користуватись – міру транспортної доступності. У публікації [2] автор наводить таке визначення доступності: "Доступність - це легкість, з якою людина може дістатись до будь-якого заняття яким вона може займатись у місті, використовуючи певну транспортну

систему". Прикладом доступності може бути легкість подорожей на роботу у жителя спального району.

Той же автор пропонує найбільш загальне визначення доступності  $A_i$  для точки  $i$ :

$$A_i = \sum_j O_j f(C_{ij}), \quad (2.1)$$

де  $O_j$  – множина можливостей (або об'єктів інфраструктури), доступних у точці  $j$ ;  $C_{ij}$  – вартість подорожі з точки  $i$  в точку  $j$  (може виражатись як грошова вартість, час, відстань тощо);  $f(C_{ij})$  – якась функція від вартості. Зазвичай функцію  $f$  обирають такою, щоб вона спадала при зростанні вартості  $C_{ij}$ . Це відповідає меншій доступності більш далеких об'єктів інфраструктури. [3]

### 3. Модель транспортної доступності, заснована на відстані

Одним із варіантів оцінки транспортної доступності місцевості є використання відстані в якості міри віддаленості між двома точками:

$$C_{ij} = d_{ij}, \quad (3.1)$$

де  $d_{ij}$  – Евклідова відстань між точками  $i$  та  $j$ .

У загальному випадку функцію  $f(C_{ij})$  для моделі транспортної доступності, що заснована на відстані, можна виразити так:

$$f(C_{ij}) = \begin{cases} 0 & | C_{ij} > d, \\ 1 & | C_{ij} \leq d. \end{cases} \quad (3.2)$$

де  $d$  – максимально допустима відстань між точками  $i$  та  $j$ .

Таким чином для заданої точки  $i$  можна визначити множину точок  $j$ , які знаходяться на певній відстані або ближче, ніж усі інші точки на площині. Спроектуювши цю множину на карту місцевості, отримаємо карту доступності певного району засновану на відстані.

Показником транспортної доступності у цій моделі буде кількість корисних точок  $O$ , що знаходяться на заданій або меншій відстані від вказаної точки.

### 4. Ізохронна модель транспортної доступності

Ізохронна модель транспортної доступності заснована на окремому випадку рівняння (2.1), наведеного вище, де параметр  $C_{ij}$  визначається так:

$$C_{ij} = t_{ij}, \quad (4.1)$$

де  $t_{ij}$  – час, необхідний щоб перейти з точки  $i$  в точку  $j$  (виражений в одиницях часу, наприклад секундах чи хвилинах).

Функція  $f(C_{ij})$  в цій моделі приймає такий вигляд:

$$f(C_{ij}) = \begin{cases} 0 & | C_{ij} > t, \\ 1 & | C_{ij} \leq t. \end{cases} \quad (4.2)$$

де  $t$  – граничний час подорожі з точки  $i$  в точку  $j$ .

Час переходу між двома точками можна вирахувати за допомогою багатьох чинників, таких як дороги на місцевості, обмеження швидкості, доступність громадського транспорту тощо. Застосувавши таку модель і знайшовши відповідно множину точок  $j$ , до яких з точки  $i$  можна дістатись за визначений час, можна спроектувати цю множину точок на карту місцевості, отримавши таким чином ізохронну карту транспортної доступності для певного району місцевості.

Ізохронна лінія – це лінія, що з'єднує усі точки на місцевості з однаковим часом подорожі до них із визначеної іншої точки. Під ізохронною областю слід розуміти множину точок, до яких із початкової точки можна дістатись за визначений час або швидше. [4]

Показником транспортної доступності у цій моделі буде кількість корисних точок, які можна досягнути з вихідної точки за визначений час.

### 5. Порівняння моделей транспортної доступності

За допомогою інформаційної системи, побудованої на основі відкритих платформ Open Street Map та Open Trip Planner [5], був проведений аналіз транспортної доступності. За вихідну точку для аналізу було прийнято місцезнаходження кампусу НТУУ «КПІ». Як видно із зображених на рис. 1 карт, створених на основі розглянутих моделей для режиму пішої доступності, ізохронна модель краще враховує особливості транспортної інфраструктури міста – у даному випадку, пішохідні дороги. Як видно із цих карт, точки, що мають рівну відстань на мапі часто реально не є однаково доступними.



Рис. 1. Карта рівних відстаней (а) та ізохронна карта (б)

### Висновки

З визначення і побудови карт транспортної доступності районів міста для конкретного прикладу ми виявили, що ізохронна модель транспортної доступності більш точно показує доступність різних точок на карті. Ми показали, що використання ізохронної моделі має значні переваги порівняно з моделлю, побудованою на відстаннях. Найбільша перевага – можливість поєднання з ізохронною моделлю складних моделей визначення часу подорожі між двома точками.

В майбутньому необхідно встановити можливість використання відкритих даних для побудови складніших і точніших ізохронних моделей, що враховують такі аспекти як громадський та особистий транспорт, час доби для здійснення сполучення, завантаженість доріг. Це дозволить створити корисну модель для аналізу транспортної доступності районів міста.

### Список літератури

1. Пашинська Н. М. Сучасні особливості територіальної організації транспортної системи //Проблеми суспільної географії. – 2010.
2. Handy S. L., Clifton K. J. Evaluating neighborhood accessibility: Possibilities and practicalities //Journal of transportation and statistics. – 2001. – Т. 4. – №. 2/3. – С. 67-78.
3. Koenig J. G. Indicators of urban accessibility: theory and application //Transportation. – 1980. – Т. 9. – №. 2. – С. 145-172.
4. O'Sullivan D., Morrison A., Shearer J. Using desktop GIS for the investigation of accessibility by public transport: an isochrone approach //International Journal of Geographical Information Science. – 2000. – Т. 14. – №. 1. – С. 85-104.
5. Tilahun N., Yin S., Li M. The Metropolitan Chicago Accessibility Explorer.

## **ПІДСИСТЕМА АВТОПІЛОТУВАННЯ ТА МОДЕЛЮВАННЯ РОБОТИ БЕЗПІЛОТНОГО АВІАЦІЙНОГО КОМПЛЕКСУ У СКЛАДІ СИСТЕМИ МОНІТОРИНГУ СТАНУ ПОСІВІВ**

В даній статті розглянуто застосування новітньої системи для ринку аграрного бізнесу дії якої направлені на підвищення якості врожаю та його кількості. Описані складові системи та принцип роботи основних задач - прокладання автоматичного маршруту польоту, введення ручних даних для польоту виходячи з погодних умов, а також поданий детальний опис процесу моделювання.

**КЛЮЧОВІ СЛОВА:** АВТОПІЛОТУВАННЯ, БЕЗПІЛОТНИЙ АВІАЦІЙНИЙ КОМПЛЕКС, МОДЕЛЮВАННЯ ПОЛЬОТУ, ПОГОДНІ УМОВИ, ГАМІЛЬТОНІВ ГРАФ, БЕКТРЕКІНГ.

This article discusses the application of the latest system for the agrarian business market, aimed at improving the quality of the crop and its quantity. Described components of the system and the principle of the main tasks - the laying of an automatic flight route, the introduction of manual data for flight based on weather conditions, as well as a detailed description of the simulation process.

**KEYWORDS:** AUTOPILOTING, UNMANNED AVIATION COMPLEX, FLEET MODELING, WEATHER CONDITIONS, HAMILTON'S GRAPH, BACKTRACKING.

### **1. Вступ**

В наші часи з розвитком сучасних технологій все більше з'являється ідей і бізнесів на межах перетину декількох ніш. Мовою бізнесу такі перетини ще називають «голубим океаном». Реальні сфери перетину ніш не просто знайти і зовсім не просто стати першим хто зможе запустити роботу в новому напрямку, але такі «океани» мають набагато більший фінансовий потенціал ніж звичайні ніші. Одним з таких напрямів на сьогоднішній день є системи моніторингу стану посівів з застосуванням комплексів безпілотної авіації.

Зараз в Україні великі аграрні компанії залучають найрізноманітніші методи відслідковування стану врожаю від отримання фото з супутників до залучення мотодельтапланів з спеціальним оснащенням. Найрозвиненішим способом відслідковування якості врожаю на сьогоднішній день є мотодельтаплани через відносно дешеву експлуатацію та можливість довгий час перебувати в повітрі. Незважаючи на це, в даному підході є і значні мінуси. До основних можна віднести: небезпеку для життя пілота, надмалу корисну вагу мотодельтаплану, неможливість польоту в поганих погодних умовах.

Виходячи з цього застосування безпілотників стає максимально актуальним у наш час. Та треба зазначити, що в даному

підході також є певні мінуси і найбільший з них велика вартість запуску проекту. Система має складатись з наступних базових блоків (підсистем): «підсистема автопілотування безпілотним авіаційним комплексом», «підсистема моделювання плану польоту», «підсистема управління поворотними антенами», «наземна станція управління» та «підсистема мультиспектрального аналізу аерофото-знімків».

Нижче більш детально будуть описані підсистеми автопілотування та моделювання польоту безпілотної авіаційного комплексу.

### **2. Підсистема автопілотування безпілотним авіаційним комплексом**

Як було сказано вище - існує декілька підходів для отримання даних про реальний стан посіву. На даний момент розвитку систем спостереження основними є знімки з супутників, аерофотозйомка за допомогою маломоторної авіації, мотодельтапланів, та безпілотних літальних комплексів. Найпоширенішим на сьогоднішній день є спосіб отримання даних за допомогою залучення мотодельтапланів, але, як було зазначено вище, він має значні недоліки.

У випадку залучення безпілотної авіаційного комплексу вирішуються дві задачі. Перша – при наявності в замовника більше двох окремих полів для аналізу

вирахувати черговість обльоту для складання оптимального маршруту обльоту кожного окремого регіону. Друга – після обчислення черговості обльоту регіонів ми знаємо координати точок входу в регіон і параметри камери, яка застосовується для вирішення задачі моніторингу стану посіву, тим самим постає задача вирахувати оптимальний маршрут обльоту кожного регіону з середини.

Базовими вхідними параметрами є координати полів з посівами які запитуються в клієнта і вносяться в програму оператором безпілотного авіаційного комплексу.

### 3. Постановка задачі прокладання оптимального маршруту

При наявності більше ніж двох регіонів інтересу (тобто, більше ніж двох окремих полів) стає актуальною задача по прокладанню оптимального маршруту обльоту заданих регіонів, що ускладнюється необхідністю побудови оптимального маршруту ще й в середині кожного з регіонів. При розв'язанні цієї задачі необхідно враховувати обмеження, що виходять з характеристик камери та параметрів борта.

Для вирішення задачі пошуку оптимального шляху обльоту регіонів доцільно застосовувати алгоритми пошуку гамільтонового циклу, а вирішення задачі розрахунку найкоротшого обльоту регіону зсередини витікає з обмежень, заданих параметрами безпілотника та технічними обмеженнями камери.

Дамо визначення гамільтоновому графу. У випадку коли граф має простий цикл, в якому містяться кожна з вершин графу по одному разу, цей цикл вважається гамільтоновим циклом, а граф називається гамільтоновим. [1]

Відповісти на питання, чи має граф гамільтонів цикл, зазвичай, дуже важко. Вивчення достатніх умов наявності в графі гамільтонового циклу – один із важливих напрямків у теорії графів. Інтуїтивно зрозуміло, що граф із багатьма ребрами, достатньо рівномірно розподіленими, з великою ймовірністю має гамільтонів цикл. Алгоритм розв'язання задачі базується на теоремі «Якщо для кожної вершини  $v$  зв'язаного простого графа з  $n \geq 3$  вершинами виконується нерівність  $d(v) \geq n/2$ , то цей граф має гамільтонів цикл.» [2].

### 4. Алгоритм знаходження гамільтонового циклу на основі бектрекінгу

Існує алгоритм побудови гамільтонового циклу в графі (якщо він існує) на основі бектрекінгу. Алгоритм полягає в наступному. Починаємо з довільної вершини. Будуємо шлях без повторення вершин, доки це можливо. Якщо вдалося пройти всі вершини, то перевіряємо, чи існує ребро, що з'єднує останню й початкову вершини цього шляху. Якщо описаний процес у певний момент неможливо продовжити, то повертаємося на одну вершину назад і намагаємося продовжити побудову шляху (без повторення вершин) іншим способом.

Цей алгоритм значно зменшує кількість ітерацій в порівнянні з повним перебором.

Далі на рисунку 2 наведено псевдокод алгоритму пошуку оптимального маршруту [6].

```

begin
  activeset := {0};
  bestval := NULL;
  currentbest := NULL;
  while activeset is not empty do
    choose a branching node, node k ∈ activeset;
    remove node k from activeset;
    generate the children of node k, child i, i=1, ..., nk,
    and corresponding optimistic bounds obi;
    for i=1 to nk do
      if obi worse than bestval then kill child i;
      else if child is a complete solution then
        bestval := obi, currentbest := child i;
      else add child i to activeset
    end for
  end while
end

```

Рис. 2. Псевдокод роботи алгоритму

### 5. Алгоритм побудови маршруту всередині регіону

Треба сказати щодо методології побудови маршруту в середині регіону інтересу те, що він базується на обмеженнях які вибираються оператором безпілотного літального апарату виходячи з потреб які закладаються на кожен окремих політ.

Основними параметрами які будуть мати вплив на побудову маршрута всередині регіону є вибір камери Mapir Survey 3 чи 2 покоління, та радіуси розвороту безпілотного літального апарату. Оскільки для подальшого аналізу аерофотознімків ми маємо отримати фото в двох спектрах червоному та інфрачервоному на літальному апараті

застосовується спеціальний вид камер, які мають певні обмеження щодо висоти польоту в зоні зйомки, повздожнього та поперечного перекриття. Виходячи базових обмежень і координат точок входу в регіон - складається маршрут в зоні інтересу. На рис.4 зображено побудований маршрут обльоту регіону.

### 7. Проведення моделювання польоту та погодні обмеження

Після проведення всіх попередніх налаштувань (відповідна форма для задання параметрів налаштувань представлена на рис.5) і введення в систему координат полів, наданих замовником, система готова до проведення імітації польоту з урахуванням параметрів безпілота. Проміжний результат проведення моделювання поданий на рис.6. Даний функціонал доданий в систему для спрощення роботи оператора комплексу. Після проведення повного моделювання польоту виходячи від базових параметрів безпілотної літальної апаратури, ми отримуємо звіт про можливість чи неможливість виконання даного маршруту у зручній для оператора формі. На основі отриманої інформації є можливість заздалегідь розрахувати кількість стартів, час який потрібно залучити на проведення польотів та кількість витратних матеріалів. Це в свою чергу використовується в розрахунку вартості виїзду команди, та формування цін для замовника. Форма базового звіту подана на рис.3.

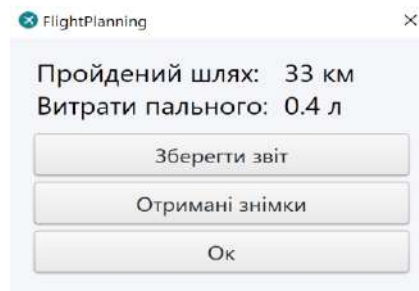


Рис. 3. Звіт після проведення моделювання польоту

Слід зазначити, що знайдений маршрут польоту не завжди буде найкращим розв'язком. Це пов'язане з тим, що алгоритмом розв'язання задачі не враховуються поточні погодні умови в зоні польоту (а саме напрям та силу вітру). З урахуванням цього, в програмному продукті є режими ручного введення алгоритмів польоту виходячи від реальних погодних умов. Для індивідуальних налаштувань розроблене меню регіонів інтересу в якому оператор має можливість ввести всі дані для маршруту польоту безпілотної авіаційної апаратури в ручному режимі. Тим самим корегувати напрям польоту та висоту всередині кожного окремого регіону та щільність перекриття зйомки в зоні.

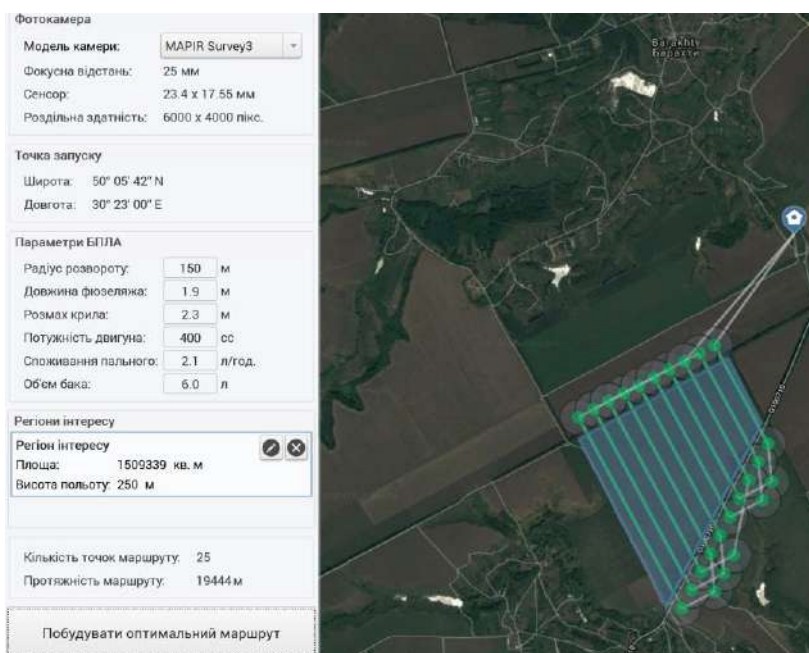


Рис. 4. Маршрут всередині регіону.



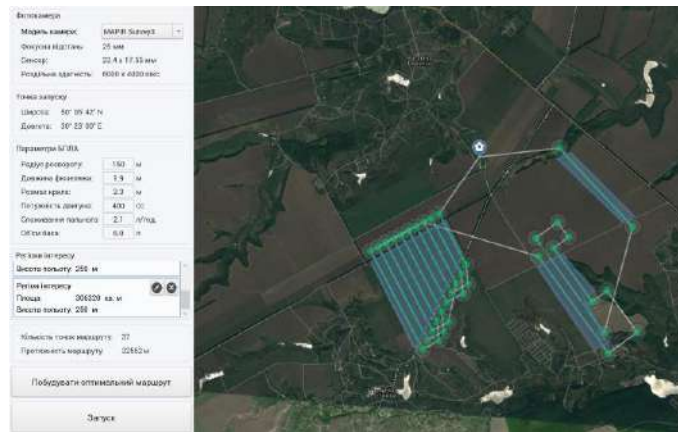


Рис. 5. Готовий план маршруту до проведення моделювання



Рис. 6. Проміжні результати проведення моделювання.

## Висновки

В рамках системи представлена реалізована ідея використання безпілотного літального комплексу для вирішення проблем моніторингу стану посівів для аграрних компаній, середніх та малих фермерських господарств. В процесі роботи над системою було розглянуто основні підходи для складання плану польоту та його моделювання. Описаний процес взаємодії з функціоналом прокладання маршрутів виходячи від погодних умов та алгоритми згідно яких реалізована робота.

## Список літератури

1. Андерсон Дж. Дискретна математика і комбінаторика. — М.: Вільямс, 2006. — 960 с
2. Кемені Дж., Снелл Дж., Томпсон Дж. Введення в кінцеву математику. — М., 1963.,
3. Bollobás, V. Graph Theory: An Introductory Course. New York: Springer-Verlag, 1979.
4. Chartrand, G. Introductory Graph Theory. New York: Dover, 1985.
5. Слюсар, В.И. Система исследований НАТО по развитию нелетального оружия.. Зб. матеріалів VI міжнародної науково-практичної конференції “Проблеми координації військово-технічної та оборонно-промислової політики в Україні. Перспективи розвитку озброєння та військової техніки”. – Київ. С. 306 - 309. (2018).
6. R. Bellman: On a Routing Problem // Quarterly of Applied Mathematics. 1958. Vol 16, No. 1. С. 87-90, 1958.

УДК 004.056.57

НАФІЕВ А.Е.

ГРИША О.В.

## БЕЗПЕКА ВІРТУАЛІЗАЦІЇ: МОНІТОРИНГ ТА ІНТРОСПЕКЦІЯ ВІРТУАЛЬНОЇ МАШИНИ

Все частіше найважливіші критичні системи піддаються віртуалізації в цілях економії коштів на обладнання. Одночасно з цим зростають вимоги стандартів і правових, нормативних зобов'язань в цій сфері. У цій статті розглядається, то яким чином комп'ютерна безпека буде адаптуватися до контролю за віртуалізованими середовищами.

**КЛЮЧОВІ СЛОВА:** ВІРТУАЛІЗАЦІЯ, ІНТРОСПЕКЦІЯ ВІРТУАЛЬНОЇ МАШИНИ, ГІПЕРВІЗОР, СИСТЕМА ЗАХИСТУ ВІД ВТОРГНЕНЬ, КІБЕРАТАКА, ШКІДЛИВЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

Increasingly, critical critical systems are being virtualized to save money on hardware. At the same time, the requirements of standards and legal and regulatory obligations in this area are growing. This article discusses how computer security will adapt to the control of virtualized environments.

**KEYWORDS:** VIRTUALIZATION, VIRTUAL MACHINE INTROSPECTION, VIRTUAL MACHINE MONITOR, INTRUSION PROTECTION SYSTEM, CYBER ATTACK, MALWARE.

### 1. Вступ

Сьогодні віртуалізація найчастіше використовується в якості опису розбиття фізичної системи на кілька віртуальних екземплярів (ОС), кожен з яких працює окремо один від одного. Все частіше критичні системи віртуалізуються з метою економії коштів на обладнання. У той же час зростають вимоги стандартів і правових, нормативних зобов'язань, які змушують підприємства дотримуватися їх. Для того щоб забезпечувати конфіденційність, цілісність і доступність даних в віртуалізованих середовищах потрібно більш глибоке розуміння рішень з безпеки віртуалізаційних середовищ. Існування таких механізмів як виявлення вторгнень і профілактичні системи дозволяють простежити і не допустити вторгнення в систему віртуалізації. Ці системи мають потужні можливості моніторингу і успішно використовуються в традиційних обчисленнях. Але той факт що ці технології спочатку були розроблені без урахування застосування їх в системах віртуалізації робить їх менш ефективними.

Нові технології захисту, такі як інтроспекція віртуальних машин (Virtual machine introspection VMI) з'явилися порівняно недавно. Технологія VMI спеціально створена і адаптована для

моніторингу віртуалізованих середовищ. Незважаючи на те що вона підвищує рівень пропонованої захисту, ці системи мають власні обмеження і недоліки. У цій статті розглядаються технології віртуалізації, що виникають загрози і доступні механізми моніторингу віртуалізованих середовищ. Так само ми розглянемо деякі критичні обмеження здібностей VMI забезпечувати безпечну платформу і коротко опишемо як за допомогою деяких прийомів ми можемо направити ці уразливості проти атакуючого, забезпечуючи ще один додатковий рівень захисту.

### 2. Ризики безпеки

З огляду на недавній інтерес до віртуалізації, у неї є потенціал, щоб змінити підхід людей до своїх обчислювальних вимогам на фундаментальному рівні. На жаль, віртуалізація успадковує багато загрози безпеки, пов'язані з традиційними обчислювальними проблемами, і додатково вводить нові проблеми безпеки, яких немає в традиційних обчислювальних середовищах.

*Шкідливе ПО.* Сьогодні шкідливе ПО являє собою проблему, яка може загрожувати кожній комп'ютеризованій середовищі. Підприємства стикаються з величезними

фінансовими або репутаційні втрати через зараження шкідливими програмами.

Віртуалізація не може захистити від таких атак і, по суті, може погіршити ситуацію, оскільки вона надає додаткові шляхи для зараження системи. Сучасне шкідливе ПЗ може виявляти віртуальне середовище і відповідати шляхом відключення або зараження критичних компонентів таких як VMM. Бувають випадки, коли складні і стійкі шкідливі програми, зокрема руткіти, які самі використовують віртуалізацію. Після зараження системи вони розміщують її на віртуальній машині і використовують технології віртуалізації для перехоплення дій власника.

*Мережа.* Через складності віртуального середовища, конфігурація мережі стає ще складніше, коли віртуальні машини вводяться в мережу. На фізичному комп'ютері може розміщуватися кілька віртуальних машин, і завдання налаштування з'єднання між віртуальними машинами і іншою частиною мережі не є тривіальним завданням. Традиційні апаратні мережеві комутатори, які відповідають за підключення мережевих сегментів, тепер замінюються програмним забезпеченням (vSwitches) в віртуалізованих середовищах. vSwitches схильні до традиційних мережевих атак рівня-2 і обмежена видимість цих компонентів використовується тільки для оперування мережею і збереження конфігурацій.

### 3. Виявлення та запобігання вторгнень

Програмне забезпечення управління є адекватним для адміністрування і контролю над віртуальною інфраструктурою, але практично неефективно, коли справа стосується безпеки. Ця роль належить системам захисту від вторгнень Intrusion protection system (IPS). Ці системи засновані за допомогою апаратних, або програмних реалізацій, які стратегічно розміщують програмні компоненти (датчики) в захищених системах. Ці датчики забезпечують безпеку, орієнтовану на інспекцію, і спрацьовують в разі порушень системи.

Існують два різних підходи до розробки і використання IPS. Перший підхід встановлює IPS всередині віртуальної машини - це система захисту від вторгнень на основі хоста Host-based intrusion protection system (HIPS). Інший підхід встановлює IPS в мережі периметра і дозволяє IPS контролювати повний сегмент мережі (домен) з декількома віртуальними машинами - це мережева система захисту від вторгнень Network-based intrusion protection system (NIPS). На малюнку 1 показано, як два описаних вище підходу реалізовані в віртуалізованому середовищі. Кожен з цих підходів має свої переваги і недоліки (які ми коротко обговоримо). Спосіб розміщення IPS всередині інфраструктури є основним фактором у визначенні гарантій безпеки, які він пропонує.

Основною перевагою HIPS є глибокий захист від загроз. Оскільки вони встановлені всередині віртуальної машини, вони мають повну видимість запущених процесів і здатні захищати роботу внутрішньої віртуальної машини від потенційних небезпек. Це означає, що раніше невідомі загрози (наприклад, нульовий день загрози) потенційно можуть бути виявлені і нейтралізовані через їх «ненормального» поведінки.

З іншого боку, основною перевагою NIPS є можливість контролювати всі комунікації між віртуальними машинами і VMM (і, в кінцевому рахунку, світом). Пакети, відправлені між світом і віртуальними машинами, можуть бути перевірені на наявність шкідливого трафіку, і будь-які атаки, що містяться в цих пакетах, можуть бути запобігати до того, як почнуться нападу.

Нарешті, той факт, що NIPS знаходиться за межами віртуальних машин, забезпечує певну незалежність від операційних систем і розгортання віртуальних машин.

Ми обговорили і чітко вказали обмеження систем запобігання вторгнень на основі хоста і мережі. За своєю природою механізми, засновані на хостах, уразливі для складних атак шкідливого ПО, які відключають HIPS всередині віртуальної машини. Навіть у відсутності атак необхідність окремо налаштовувати кожен

HIPS може витрачати адміністративний час, і нерідко HIPS стикаються з труднощами при оновленні основних компонентів віртуальної машини. Нарешті, виділена захист для кожної віртуальної машини буде споживати великі обсяги ресурсів процесора і пам'яті хоста.

Всі ці обмеження можна було б уникнути, використовуючи підхід NIPS, але, на жаль він також має свої недоліки. Мережева захист означає, що IPS не може контролювати дії, що відбуваються всередині віртуальних машин. Захист периметра також означає, що атаки, які виходять від однієї віртуальної машини і призначені для іншої віртуальної машини в одному домені, не можуть бути запобігати. У цьому контексті NIPS може переглядати домен як єдину сутність і перевіряти трафік між цим доменом і гіпервізором.

Зв'язок між віртуальними машинами в одному домені протікає через vSwitch без перевірки (див. Рис. 1). Коли використовуються NIPS, потрібно більше уваги, щоб гарантувати, що віртуальні машини, що знаходяться в одному домені, мають схожий профіль. Було б нерозумно, щоб віртуальні машини з різними вимогами безпеки і ризиками були захищені одним NIPS в рамках однієї і тієї ж політики безпеки.

#### 4. Інтроспекція віртуальної машини

Інтроспекція віртуальної машини (VMI) - це гібридний підхід до моніторингу віртуалізації і розміщенні захисту. Він спрямований на об'єднання функціональних можливостей двох систем захисту (HIPS і NIPS), що обговорювалися раніше, з метою досягнення оптимального результату.

Архітектура VMI заснована на віртуальній машині, яка відповідає за захист і управління всіх інших віртуальних машин. Захист VM включає в себе безпеку політик і конкретних правил, які повинні застосовуватися на гостьових віртуальних машинах. Ця спеціальна віртуальна машина взаємодіє з VMM (гіпервізор), перехоплює і перевіряє всю інформацію низького рівня (переривання, звернення до пам'яті і т. д.), Яка передається між VMM і гостьовими віртуальними машинами. Оскільки всі операції безпеки

виконуються за допомогою VMM, процес моніторингу вимагає невеликі накладні витрати. VM відповідає за безпеку ізолювана від інших гостьових віртуальних машин і працює з більш високими привілеями, ніж гостьові віртуальні машини. Це додає додатковий рівень захисту від шкідливих атак, які виникають в непривілейованих віртуальних машинах. На малюнку 2 показана топологія інфраструктури VMI.

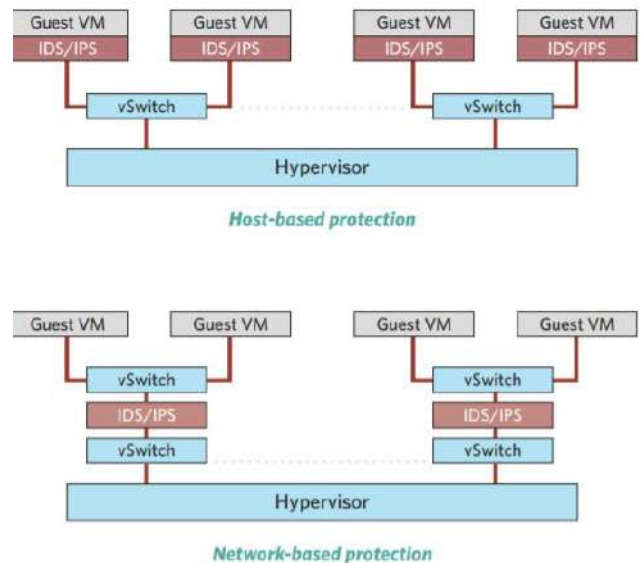


Рис. 1. Архітектура HIPS та NIPS

Результатом є механізм захисту з усіма перевагами традиційних технологій моніторингу: надійна, ефективна і централізована система безпеки, яка може перевіряти всі комунікації VM і отримувати деяку інформацію про їх внутрішню роботу. Більшість сучасних загроз і атак шкідливих програм можна уникнути за допомогою технології VMI.

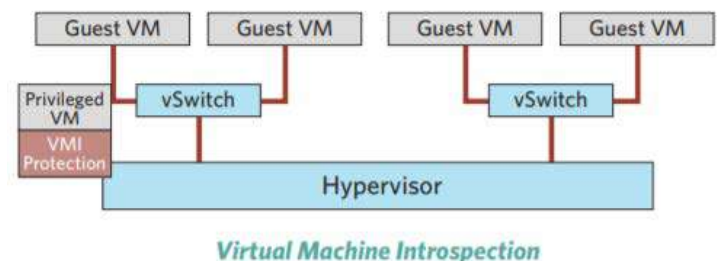


Рис. 2. Архітектура VMI

#### 5. Обмеження VMI

Жодна технологія безпеки не ідеальна, і технологія VMI не є винятком. Її обмеження

тісно пов'язані з тим, що віртуальна машина відповідає за безпеку ізолювана від віртуальних машин, які вона захищає. У віртуальній машині безпеки є велика кількість низкоуровневої інформації про внутрішні процеси гостьових віртуальних машин (наприклад, всі запити доступу до пам'яті VM). Однак ця інформація сама по собі марна через складність відновлення фактичного стану роботи гостьової віртуальної машини з низкоуровневої інформації. Навіть незважаючи на те, що віртуальна машина безпеки може збирати багато інформації про дії гостьової віртуальної машини, вона все ще не може зрозуміти сенс цих дій. Це питання відомий як семантичний розрив. Без необхідної семантичної обізнаності неможливо визначити фактичну мету вмісту пам'яті віртуальної машини.

## 6. Атаки

Нові атаки низького рівня, звані атаками які маніпулюють безпосередньо структурами ядра Direct kernel structure manipulation (DKSM). Ці атаки маніпулюють даними віртуальної машини, які використовують систему VMI для створення семантичної обізнаності. Операційна інформація віртуальної машини (наприклад, список процесів) копіюється в нове місце пам'яті по шкідливому коду, а зараженої віртуальній машині надається вказівка продовжити роботу з використання інформації, що зберігається в нових місцях. Якщо додаток VMI контролює вихідні осередки пам'яті для збору інформації про безпеку, то їх вміст буде виглядати доброякісним, так як ця пам'ять інфікованої віртуальною машиною більше не використовується. Діаграма на малюнку 3 показує приклад, коли Віртуальна машина будучи заражена троянським додатком, невидима для додатка VMI. Троянській додаток представлено в VMI як законний додаток (програвач Windows Media). Хоча додаток VMI може перевіряти розташування зі скопійованої інформацією, їх вміст просто розглядається як біти пам'яті без певної семантики. Ці атаки навіть дозволяють Приховати дані від самої віртуальної машини.

Використовуючі приклад на малюнку 3, все, що потрібно для зловмісника - це змінити файл trojan.exe на той, що виглядає законно, наприклад, svchost.exe. Ефективний метод, додаток VMI бачить wmplayer.exe, а механізм захисту всередині VM бачить svchost.exe, який дозволяє безперешкодно запускати фактично вміст пам'яті (шкідливої програми). Ці атаки не обмежуються просто маніпулюванням іменами процесів, але можуть використовуватися для зміни будь-якої системної інформації. Потенційний результат віклікає занепокоєння: невловимий зловмісник з можливістю таємного захоплення віртуальних машин. Крім того, жоден з існуючих продуктів VMI не має можливості виявляти або запобігати цим атакам.

## 7. Заключення

Ці атаки здаються руйнівними, але можливо, що від них може бути щось корисне. Ми могли б використовувати їх в захисних цілях.

Метою було б доручити віртуальній машині автоматично приховувати свою конфіденційну інформацію в разі атаки. Ми коротко викладемо, як може працювати такий захисний механізм – операційна інформація віртуальної машини повинна бути скопійована в місце в пам'яті і частково змінена. Нова частина коду стає посередником між усіма додатками ОС, які запитують дані VM (наприклад, для процесів лістингу) та фактичні дані. Тригери / перехоплювачі поміщаються в віртуальну машину для виявлення можливих спроб атаки.

Поки система сприймається як безпечна, новий код захисту повертає істинне значення запитом. Якщо був запущений перехоплювач (тобто виявлена потенційна атака), то посередник повертає фальсифіковане уявлення запитаного значення. На малюнку 4 показано поняття цієї концепції, коли ми хочемо приховати файл firewall.exe від зловмісника і представити його як calc.exe. Як і при атаці, це дозволяє нам спотворювати системну інформацію для додатка VM, за винятком того, що в цьому випадку шкідлива програма обманюється.

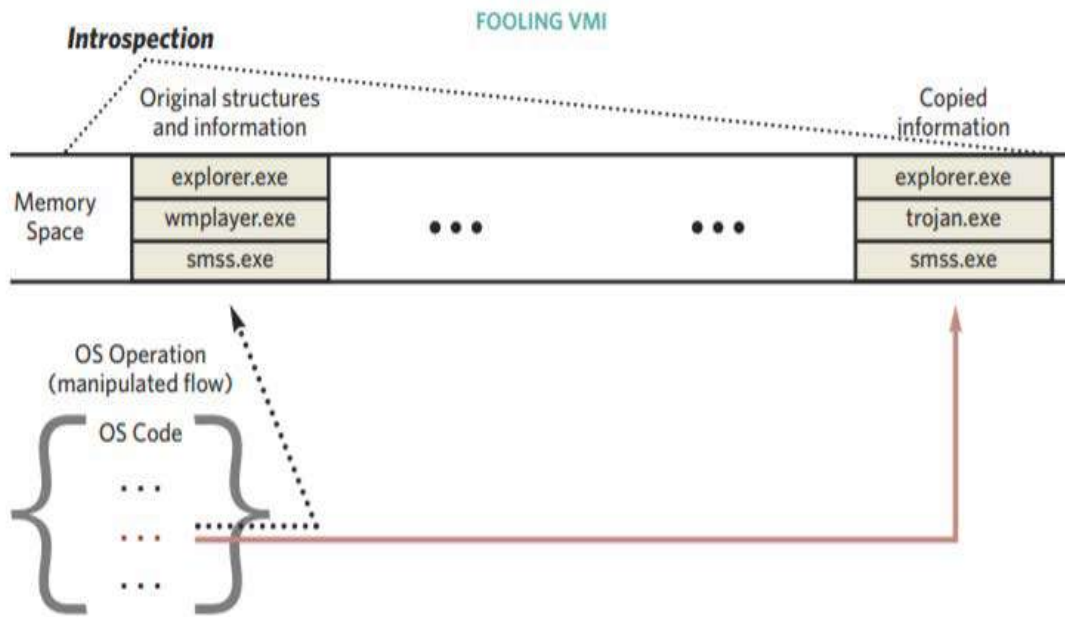


Рис. 3. Зараження троянським вірусом VM, невидиме системою VMI

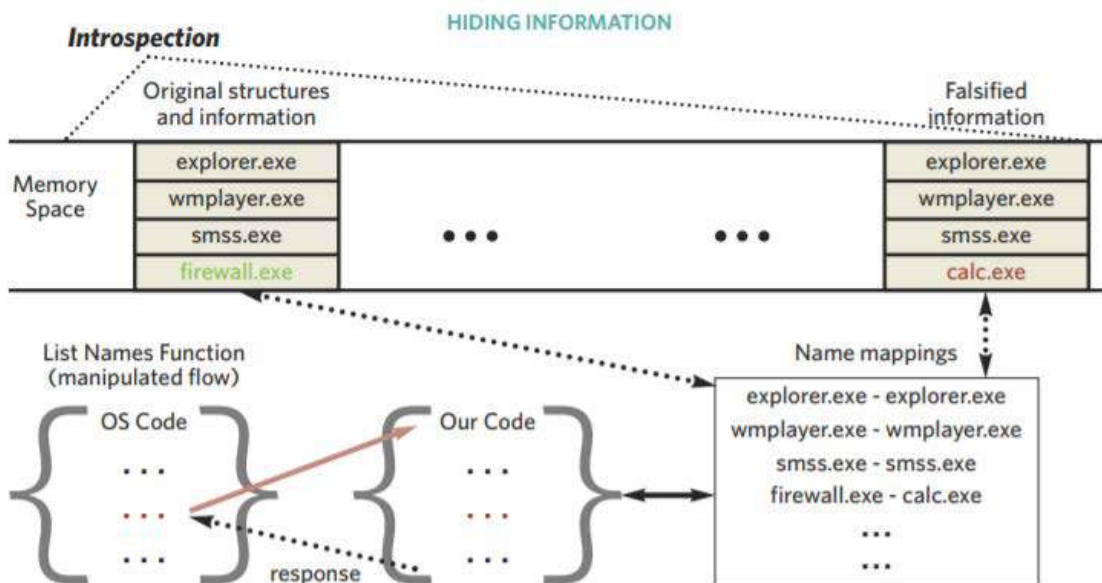


Рис. 4. Зміна імені важливого процесу системи з метою безпеки та ухилення від шкідливого ПО

## Висновки

Добре відомо, що традиційні системи моніторингу і захисту недостатні для задоволення потреб забезпечення віртуалізації. Існують серйозні обмеження цих систем захисту, починаючи від обмежених гарантій безпеки і закінчуючи їх впливом на продуктивність.

Технологія інтроспекції віртуальної машини (VMI) є важливим кроком для моніторингу віртуалізованих середовищ. Однак технологія VMI не настільки безпечна, як широко поширена. VMI забезпечує захист від найбільш поширених атак на сьогоднішній день, але її надійність недостатня для захисту від виникаючих атак. Хоча атаки, згадані в цій статті, на

даний час не розглянуті докладно, це не означає, що користувачі віртуалізації можуть бути впевнені у своїй безпеці.

Семантичний розрив як і раніше залишається головною темою дослідження. Дослідження з аналізу системи і пам'яті робляться в спробі подолати це обмеження. В даний час широко поширена технологія VMI не може вважатися цілком безпечною і надійною до тих пір, поки семантичний розрив не буде повністю перекритий.

#### **Список літератури**

1. Payne, B. D. Simplifying virtual machine introspection using LibVMI. Sandia report (2012).
2. Payne, B. D., De Carbone, M., and Lee, W. Secure and flexible monitoring of virtual machines. In Computer Security Applications Conference, 2007. ACSAC 2007. Twenty-Third Annual (2007), IEEE, pp. 385–397.
3. Poisel, R., Malzer, E., and Tjoa, S. Evidence and cloud computing: The virtual machine introspection approach. *JoWUA* 4, 1 (2013), 135–152.
4. Garfinkel, T., Rosenblum, M., et al. A virtual machine introspection-based architecture for intrusion detection. In NDSS (2003), vol. 3, pp. 191–206.
5. Sgandurra, D., and Lupu, E. Evolution of attacks, threat models, and solutions for virtualized systems. *ACM Comput. Surv.* 48, 3 (Feb. 2016), 46:1–46: