

# БИОНИКА ИНТЕЛЛЕКТА

ИНФОРМАЦИЯ, ЯЗЫК, ИНТЕЛЛЕКТ

---

№ 2 (95)

2020

---

НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Основан в октябре 1967 г.

Учредитель и издатель

*Харьковский национальный университет радиэлектроники*

Периодичность издания – 2 раза в год



Научно-технический журнал  
«БИОНИКА ИНТЕЛЛЕКТА»

ISSN 0555-2656

Основан Харьковским национальным университетом  
радиоэлектроники в 1967 году

Реферирование и индексирование:

Google Scholar



INDEX  COPERNICUS  
I N T E R N A T I O N A L



---

Журнал включен в список научных специализированных изданий Украины  
по техническим и физико-математическим наукам  
согласно приказа Министерства образования и науки Украины № 820 от 11.07.2016



**В. А. Широков**

Український мовно-інформаційний фонд НАН України, Київ  
vshirokov48@gmail.com

**М. В. Надутенко**

Український мовно-інформаційний фонд НАН України, Київ  
maxkrb@gmail.com

**О. Є. Стрижак**

Національний центр «Мала академія наук України», Київ  
sac953@gmail.com

**С. С. Ющенко**

Український мовно-інформаційний фонд НАН України, Київ  
Yuschenko\_Svetlana@i.ua

## ТЕХНОЛОГІЧНІ ЗАСАДИ ЛОГІКО-ЛІНГВІСТИЧНИХ ДОСЛІДЖЕНЬ ЗАКОНОДАВСТВА

У статті обґрунтовано застосування інтелектуальних когнітивних комп'ютерних технологій в логіко-лінгвістичних дослідженнях законодавства України на прикладі проведеного авторами комплексного логіко-лінгвістичного дослідження Закону України «Про особливу процедуру усунення Президента України з посту (імпічмент)» № 39-ІХ від 10 вересня 2019 року на авторських програмних інструментах – Системі аналізу текстів та Когнітивній ІТ-платформі «ПОЛІЕДР».

КОГНІТИВНА ІТ-ПЛАТФОРМА «ПОЛІЕДР», СИСТЕМА АНАЛІЗУ ТЕКСТІВ, ЛОГІКО-ЛІНГВІСТИЧНЕ ДОСЛІДЖЕННЯ, ОНТОЛОГІЯ, КОНЦЕПТОГРАФІЧНИЙ АНАЛІЗ, ЛОГІКО-ЛІНГВІСТИЧНІ ДЕФЕКТИ

**V. A. Shyrovkov, M. V. Nadutenko, O. Ye. Stryzhak, S. S. Yuschenko. Technological fundamentals of logical and linguistic research of legislation.** The article substantiates and describes the use of the author's software products – the Text Analysis System and the Cognitive IT Platform «POLYHEDRON» in the prospective logical-linguistic studies of Ukrainian legislation. Authors use the complex logical and linguistic research of the Law of Ukraine "On special procedure for the removal of the President of Ukraine from office (impeachment)" No. 39-IX of September 10, 2019 as an example.

COGNITIVE IT PLATFORM «POLYHEDRON», TEXT ANALYSIS SYSTEM, LOGICAL AND LINGUISTIC RESEARCH, ONTOLOGIES, CONCEPTOGRAPHICAL ANALYSIS

### ВСТУП

Удосконалення національного законодавства є першочерговим завданням уряду будь-якої країни, оскільки від дієвості законів та інших нормативно-правових актів залежить ефективність їх правового впливу на політичні, економічні, соціальні та інші процеси, що відбуваються в країні. Проте жодне національне законодавство не можна назвати досконалим. Так, недоліком законодавства США визнається архаїчність, Російської Федерації – неузгодженість нормативно-правових актів федерального та місцевого рівнів, Республіки Білорусь – постійне коригування законів для досягнення відповідності з указами Президента; Грузії – істотний вплив національних традицій та культурних звичаїв, що ускладнює імплементацію міжнародних нормативно-правових норм до грузинського законодавства; Великобританії – велика кількість формально діючих законів, що фактично втратили актуальність; Індії – відсутність правових наслідків за вчинення соціальних, правових та інших переслідувань на підставі расової, кастової та релігійної належності; Італії – істотний вплив консервативного національного менталітету. Приклади такого типу можна продовжувати і продовжувати. Однею з наріжних проблем законодавства України є

недієвість у вітчизняних реаліях правових норм, скопійованих із актів інших країн та таких, що дісталися у спадок від часів СРСР<sup>1</sup>.

Зазначені недоліки стосуються юридичної сфери, але не менш важливою для забезпечення дієвості законодавства є лінгвістична сфера, оскільки субстратом права, формою, в якій втілено правові акти є природна мова. Закон – це результат законотворчого процесу, етапами якого є розробка концепції, організація робочої групи, формування тексту законопроекту, проведення експертиз та обговорень, врахування зауважень, редагування і, власне, процедура прийняття. Відома серед правників метафора «кола законодавчого пекла»<sup>2</sup> відображає складність перебігу законодавчого процесу та почасти пояснює недостатність уваги до лінгвістичного оформлення законопроектів «зацикленістю» учасників законотворчості на проходженні «кіл законодавчого пекла»,

<sup>1</sup> Огляд недоліків національних законодавств зроблено авторами статті виключно з ілюстративною метою на основі аналітичних веб-публікацій, отриманих на запити «недоліки законів <назва країни>». Автори не претендують на вичерпність наведеного переліку.

<sup>2</sup> Див. напр. Ромовська З. В. Українське цивільне право. Загальна частина: академічний курс: Підручник. – 3-те вид., допов. – К.: «ВД «Дакор», 2013. – С. 112.

яке до того ж зазвичай відбувається в умовах сильного цейтноту. На нашу думку, наслідком такого стану справ, обтяженим імпліцитністю семантичних та логічних дефектів у текстах законопроектів, авральним режимом їх формування, великою завантаженістю нормопроектувальників та іншими об'єктивними чинниками, є прийняття слабких у логічному та лінгвістичному аспекті законів із руйнівним потенціалом.

На авторів цієї статті, обов'язковість законодавчих вимог нівелюється недосконалим термінологічним апаратом, застосуванням лексичних та синтаксичних конструкцій з нечіткою семантикою, смисловими прогалинами (лакунами), інтертекстуальною неузгодженістю правових приписів.

Результати проведених в Українському мовно-інформаційному фонді НАН України (далі – Фонд) логіко-лінгвістичних досліджень вітчизняного законодавства переконливо свідчать, що юридична наука, декларуючи законодавство як систему, досі немає жодної її функціональної мовно-когнітивної моделі – лексикографічної, концептографічної, онтологічної або іншої. Унаслідок цього в текстах невизначеної кількості законів (навіть у Конституції України), попри їх ретельне опрацювання висококваліфікованими фахівцями, мають місце логіко-лінгвістичні вади (дефекти; далі – ЛЛД), що проявляються на всіх рівнях функціонування тексту.

Зрозуміло, що для лінгвістичного вдосконалення законодавства недостатньо чітко визначити терміни та поняття в одному чи декількох законах або переформулювати неоднозначні приписи – це потрібно зробити в усій законодавчій базі й виключно засобами інтелектуальних когнітивних комп'ютерних технологій, оскільки обсяг нормативно-правової бази України<sup>3</sup>, з урахуванням динаміки її модифікації, виключає будь-який інший підхід до її аналізу, спроможний забезпечити її смислову та логічну коректність.

Крім великого обсягу, іншою рисою законодавства є слабка структурованість та різноплановість текстів, що його утворюють (див. виноску 3). Створення оптимальних умов для опрацювання різнопланових інформаційних масивів визначає технологію проведення логіко-лінгвістичних досліджень законодавства як таку, що відноситься до концепцій

<sup>3</sup> Станом на 24 грудня 2020 року згідно з інформацією, наведеною на офіційному сайті [rada.gov.ua](http://rada.gov.ua) до бази даних «Законодавство України» входять: 6860 законів України; 46 Кодексів; 89314 постанов; 34604 наказів; 6801 рішення; 29823 указів; 45579 розпоряджень; 83 декрети; 1356 актів; 840 вимог; 566 висновків; 107 вказівок; 334 директиви; 9 доктрин; 39 законопроектів; 3879 інструкцій; 779 конвенцій; 13010 міжнародних документів; 1762 правила та десятки інших позицій (режим доступу: <https://zakon.rada.gov.ua/laws/main/a/stru4#Typ>).

Big Data («великі дані»). Тобто комп'ютерні інструменти, орієнтовані на дослідження законодавства, мають реалізовувати когнітивні функції з урахуванням гіпервластивостей Big Data, основними з яких є рефлексія (здатність до саморозвитку), рекурсія (здатність до самовідтворення) та редукція (здатність до спрощення).

Інтерпретація зазначених гіпервластивостей зумовлює використання таких підходів у дослідженні законодавства: 1) структурний аналіз інформаційних ресурсів; 2) визначення форми взаємодії з інформаційними ресурсами; 3) визначення критеріїв вибору потрібних для дослідження контекстів.

Фонд є однією з небагатьох вітчизняних наукових установ, що володіють технологічним (повним) циклом розробки когнітивних інформаційно-аналітичних інструментів роботи з масивами природномовних текстів (далі – ПТ) відповідно до концепцій Big Data – від наукових теорій до промислових зразків. Метою цієї статті є огляд двох програмних інструментів, розроблених у Фонді для проведення різного роду лінгвістичних досліджень, а саме: Системи аналізу текстів та Когнітивної ІТ-платформи «ПОЛІЕДР», а також демонстрація їхніх функціональних можливостей на прикладі проведеного авторами статті логіко-лінгвістичного дослідження Закону України «Про імпічмент Президента України».

### СИСТЕМА АНАЛІЗУ ТЕКСТІВ

Система аналізу текстів (САТ) – комп'ютерний інструмент, розроблений на програмній платформі ULISS<sup>4</sup>, створеній на базі інформаційно-лінгвістичних теорій [1 – 5]. Операційною основою САТ є метод  $N$ -грам, згідно з яким ПТ моделюється ланцюжками слів<sup>5</sup> довжини  $n$ , що інтерпретуються як певні випадкові величини.

САТ пройшла достатню апробацію на текстах реальної складності, що дає нам право стверджувати її практичну значимість як інструменту лінгвоекспертних досліджень. Зокрема, з її допомогою було проаналізовано текст Конституції України, кілька редакцій Податкового кодексу України, статuti та передвиборчі програми українських політичних партій, низку лексикографічних праць (зокрема, Словник української мови в 20 томах) та ін.

Універсальність САТ забезпечує роботу з текстами практично будь-якої структури, обсягу та складності. Якщо об'єктом дослідження виступає законопроект, САТ дозволяє оперативно оцінити його смислову відповідність загальносистемним вимогам вітчизняної правової системи (узгодженість із Конституцією

<sup>4</sup> ULISS – Universal Lingua-Information Semantic Service.

<sup>5</sup> Терміном «слово» в цьому контексті позначено лінійну послідовність (ланцюжок) певних символів з деякого скінченного алфавіту. Виділення в таких ланцюжках слів, як *лексичних одиниць певної мови*, становить окреме завдання.

України, національним законодавством України, міжнародними угодами, згоду на обов'язковість яких надано Верховною Радою України та ін.) та вимогам законодавчої техніки до текстуального оформлення законопроектів (послідовний, стислий, логічний виклад нормативного матеріалу, чіткість формулювань, уніфікованість правової, офіційно-ділової та галузевої термінології та ін.) [6, 7].

Користувач САТ має можливість:

- прямого доступу до контекстів семантично навантажених лексичних та лексико-синтаксичних одиниць тексту (слова, словосполучення, сполучення слів, колокації тощо) в усіх граматичних значеннях із заданою лексичною відстанню між компонентами;

- обчислення смислової узгодженості (близькості) тексту аналізованого законопроекту із текстами нормативно-правових актів, відповідність яким у ньому декларується;

- обчислення смислової близькості альтернативних законопроектів, підготовлених різними суб'єктами законодавчої ініціативи для розгляду Верховною Радою України;

- дослідження законопроектів на можливе авторство (співавторство), запозичення та імовірні інформаційно-текстові джерела;

- обчислення обсягу та проведення змістового аналізу зроблених у тексті законопроекту модифікацій на різних етапах законодавчого процесу;

- локалізації та інтерпретації логіко-лінгвістичних дефектів в аналізованому документі.

Проілюструємо викладене результатами, отриманими авторами цієї статті за допомогою САТ при комплексному логіко-лінгвістичному дослідженні Закону України «Про особливу процедуру усунення Президента України з посту (імпічмент)» № 39-IX від 10 вересня 2019 року (далі – «Закон № 39-IX»), проект якого вніс Президент України В. О. Зеленський. Значимість цього закону полягає в тому, що він стосується не лише Президента України або згаданих у ньому посадових осіб, а й усього українського суспільства – як свідчення демократичного шляху розвитку держави.

Логіко-лінгвістичне дослідження Закону № 39-IX передбачало обчислення його семантичної близькості з: 1) проектом Закону України «Про імпічмент Президента України», внесеним Президентом України В. О. Зеленським; 2) альтернативним законопроектом «Про імпічмент Президента України», внесеним народним депутатом України І.А. Артюшенком, 3) Конституцією України (у редакції від 21 лютого 2019 року), 4) Регламентом Верховної Ради України № 1861-VI (у редакціях він 16 липня 2019 року та 25 серпня 2019 року). САТ обчислює семантичну близькість текстів за формальними ранговими критеріями, сутність яких розкрито нижче.

Зауважимо, що відношення семантичної близькості ПТ є симетричним лише для релевантних (об'єктивно рівнозначних) текстів, що було враховано при розробленні САТ. Наприклад, статут будь-якої політичної партії, зареєстрованої в Україні, повинен відповідати Закону про політичні партії, а не навпаки; локальні нормативні акти (правила внутрішнього розпорядку, інструкції, методичні рекомендації тощо) повинні відповідати кодифікованим актам, а не навпаки.

Нехай є два ПТ, один з них (Т1) – еталонний, інший (Т2) – аналізований. Згідно з теорією слабкої синонімії текстів [3], підтвердженої численними статистичними експериментами на текстах реальної складності, текст Т2 є семантично еквівалентним («слабко синонімічним») тексту Т1, у разі виконання хоча би однієї з трьох умов: 1) у тексті Т2 наявні 50 % та більше уніграм (окремі лексеми у тих самих граматичних формах, тобто словоформи однієї лексеми, що утворюють її словозмінну парадигму) тексту Т1 (*ранговий критерій*  $Q_1$ ); 2) у тексті Т2 наявні більше 30% біграм (дві поруч розташовані словоформи) тексту Т1 (*ранговий критерій*  $Q_2$ ); 3) у тексті Т2 наявні 10% триграм (три поруч розташовані словоформи) тексту Т1 (*ранговий критерій*  $Q_3$ ) [5, 8].

Чим вище значення будь-якого рангового критерію, тим більш близьким за змістом є аналізований текст еталонному. Значення «одиниця» будь-якого рангового критерію означає текстуальну, а отже – й смислову тотожність аналізованого тексту еталонному.

Результат обчислення САТ рангових критеріїв виводиться на екран у вигляді матриці аналізу (рис. 1).

Порівняння висновків, зроблених за результатами аналізу отриманих за допомогою САТ даних із неофіційною інформацією з компетентних джерел про хід законодавчого процесу, результатом якого стало формування остаточної версії тексту Закону № 39-IX, підтвердило достовірність наших висновків.

Так, високі (наближені до одиниці) значення рангових критеріїв  $Q_1$ ,  $Q_2$ ,  $Q_3$ , обчислені Системою для Закону № 39-IX та Регламенту Верховної Ради України № 1861-VI в редакції від 16 липня 2019 року ( $Q_1 = 0,8023$ ;  $Q_2 = 0,7357$ ;  $Q_3 = 0,6875$ ) та низькі – обчислені для Закону № 39-IX та Регламенту Верховної Ради України № 1861-VI в редакції от 25 вересня 2019 року ( $Q_1 = 0,1453$ ;  $Q_2 = 0,0764$ ;  $Q_3 = 0,0650$ ) означають, що джерелом тексту Закону № 39-IX послугував текст Регламенту, конкретно – Глава 30 і стаття 133. Справді, для формування тексту Закону № 39-IX з Регламенту було вилучено 17 статей Глави 30 – з 171-ї до 188-ї.

Низькі значення рангових критеріїв  $Q_1$ ,  $Q_2$ ,  $Q_3$  для законопроекту «Про імпічмент Президента України», внесеного народним депутатом України І.А. Артюшенком та Глави 30 Регламенту Верховної

Корпус документів		Матриця аналізу			
N	Опис документу	1. E:\TEMP\IMPICHMENT \ZU_Pryjnyatyj.txt	4. E:\TEMP\IMPICHMENT \CONSTITUTION 85 105 111 151.txt	5. E:\TEMP\IMPICHMENT \ZU REGLAMENT (do zmin) Glava 30.txt	6. E:\TEMP\IMPICHMENT \ZU REGLAMENT (pislja zmin) Glava 30.txt
1	E:\TEMP\IMPICHMENT\ZU_Pryjnyatyj.txt				
2	E:\TEMP\IMPICHMENT \Project_10340.txt	61,85	25,31	52,21	20,09
3	E:\TEMP\IMPICHMENT \Project_10340-1.txt				
4	E:\TEMP\IMPICHMENT\CONSTITUTION 85 105 111 151.txt	92,10	19,75	80,23	14,53
5	E:\TEMP\IMPICHMENT\ZU REGLAMENT (do zmin) Glava 30.txt				
6	E:\TEMP\IMPICHMENT\ZU REGLAMENT (pislja zmin) Glava 30.txt	70,47	50,47	73,80	100

Кількість документів 6 Виділено 1

**Рис. 1. Сформована САТ матриця аналізу з результатами попарного обчислення рангового критерію  $Q_1$  для текстів, завантажених до САТ для проведення комплексного логіко-лінгвістичного дослідження Закону № 39-IX (фрагмент)**

Ради України в редакції від 16 липня 2019 року ( $Q_1 = 0,2000$ ;  $Q_2 = 0,0760$ ;  $Q_3 = 0,0471$ ), указують на те, що цей законопроект базувався не на Регламенті, а є самостійним текстом (хоча й не виключається інше текстуальне джерело).

Рангові критерії менше одиниці для законопроект, внесеного Президентом України В.О. Зеленським та Закону України № 39-IX, в який втілюється цей законопроект після прийняття ( $Q_1 = 0,9210$ ;  $Q_2 = 0,8421$ ;  $Q_3 = 0,7982$ ) свідчить про наявність поправок, внесених до законопроект в ході розгляду, обговорення, доопрацювання та узгодження.

Низькі значення рангових критеріїв для законопроект І.А. Артюшенка та Закону № 39-IX свідчать про істотні смислові відмінності цих двох документів – фактично лексика перетинається на рівні семантичного ядра юридичної мови та не відображає специфіку змісту предметної галузі саме цього Закону.

Процедура імпичменту Президента України згадується у статтях 85, 105, 111, 151 Конституції України, проте обсягу їх тексту недостатньо для статистично імовірних (таких, що відображають реальні закономірності досліджуваної проблеми) висновків про ступінь відповідності аналізованих текстів Конституції України: обидва документи – Закон № 39-IX та законопроект І. А. Артюшенка мають низькі значення рангових критеріїв із зазначеними статтями Конституції України (Закон № 39-IX:  $Q_1 = 0,1976$ ;  $Q_2 = 0,0617$ ;  $Q_3 = 0,0306$ ; законопроект І. А. Артюшенка:  $Q_1 = 0,2531$ ;  $Q_2 = 0,0893$ ;  $Q_3 = 0,0394$ ).

Для зручності користувачів екранну форма САТ поділено на три поля (рис. 2). У лівому – список ланцюжків словоформ, що збігаються (на рис. 2 обрано збіги за трійками словоформ), у центральному полі – аналізований текст, у правому – еталонний текст, із яким порівнюється аналізований.

Ланцюжки словоформ, що збігаються, у центральному та правому полях акцентовано шрифтовими виділеннями. У спливаючому вікні «Статистика» наведено результати обчислення рангових критеріїв для аналізованого та еталонного текстів за трьома N-грамами – довжиною в одну словоформу (значення  $Q_1$ ), дві словоформи (значення  $Q_2$ ), три словоформи (значення  $Q_3$ ).

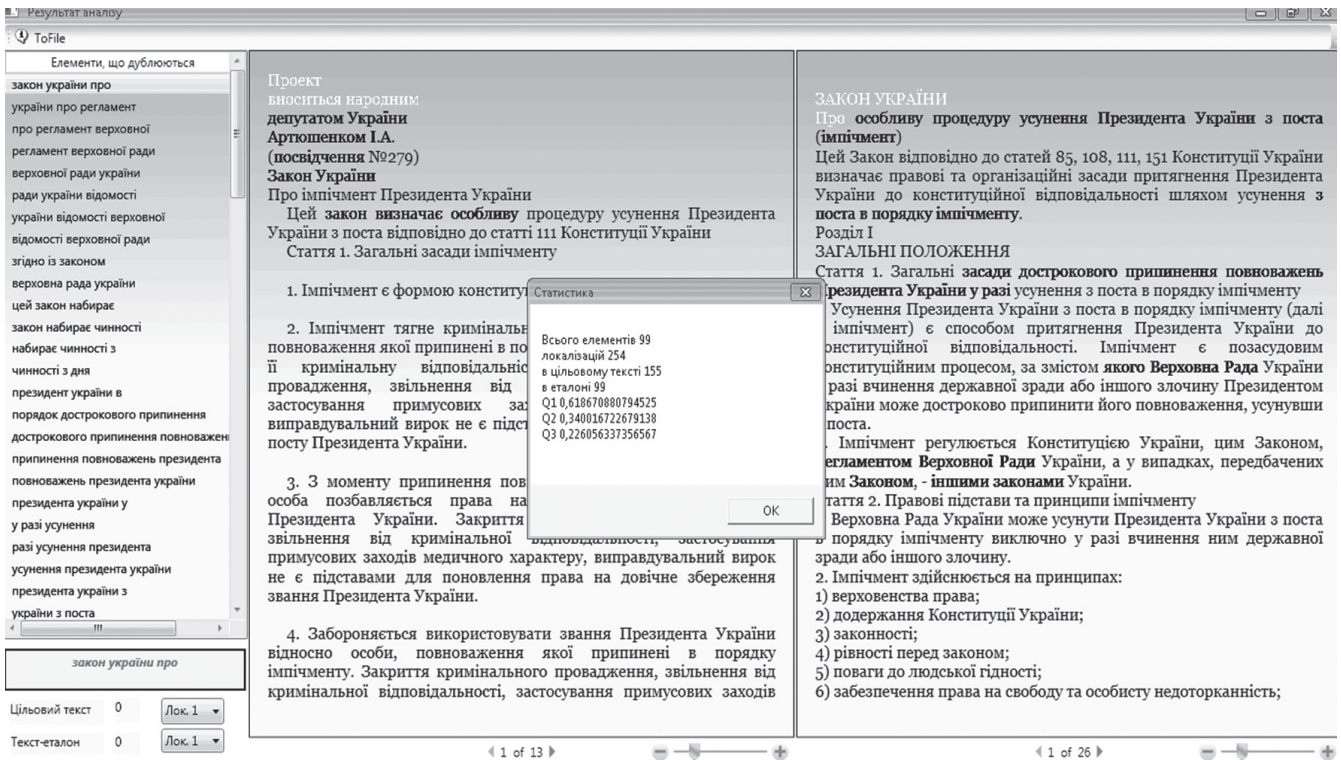
Висока швидкість обробки великих текстових масивів, простота інтерпретації, наукова надійність та наочність представлення результатів роблять САТ незамінним помічником дослідника.

### КОГНІТИВНА ІТ-ПЛАТФОРМА «ПОЛІЕДР»

Когнітивну ІТ-платформу «ПОЛІЕДР»<sup>6</sup>, орієнтовану на автоматичну обробку просторово-розподіленої неструктурованої текстової інформації великого обсягу (Big Data) та підтримку процесів раціонального вибору та прогнозування з наступним формуванням інформаційно-аналітичних WEB-орієнтованих рішень, створено в рамках проекту з розвитку системи національних лінгвістичних ресурсів та національної лінгвістичної інфраструктури («Всеукраїнський лінгвістичний діалог»).

Когнітивна ІТ-платформа «ПОЛІЕДР» забезпечує реалізацію таких інформаційних процесів, як: лінгвосемантичний аналіз мережевих інформаційних ресурсів, створених на основі різних технологій і стандартів; трансдисциплінарна інтеграція з іншими інформаційними та автоматизованими системами; таксономізація документів з урахуванням міжконтекстних зв'язків; формування онтологічних інтерактивних документів; встановлення імпліцитної інформації

<sup>6</sup> Свідоцтво про авторське право на твір. № 96078. Комп'ютерна програма «Когнітивна ІТ платформа ПОЛІЕДР» (KIT ПОЛІЕДР) (англ. «POLYHEDRON»).



**Рис. 2.** Екранна форма Системи аналізу текстів із результатами обчислення рангових критеріїв для законопроекту «Про імпічмент Президента України», внесеного Президентом України В. О. Зеленським та Закону України «Про особливу процедуру усунення Президента України з поста (імпічмент)» № 39-ІХ від 10 вересня 2019 року, в який втілюється законопроект після прийняття

в інформаційних ресурсах; глибинне та машинне навчання (Deep Learning, Machine Learning); використання форматів та протоколів Semantic Web та ін.

Когнітивна ІТ-платформа «ПОЛІЕДР» має в арсеналі інструменти, що дозволяють виконувати контекстно-семантичний, лінгвоструктурний та концептографічний аналіз ПТ, будувати таксономію документів, класифікувати та генерувати онтології предметних галузей та онтології вибору; конструювати трансдисциплінарні онтології; надавати інформаційно-аналітичну підтримку при прийнятті рішень, забезпечувати багатокритеріальний аналіз та рейтингування; здійснювати семантичний пошук лексичних структур у розподілених мережевих текстових масивах; будувати лінгвістичні кластери – цифрові лінгвістичні корпуси із засобами асоціативного пошуку семантично зв'язаних інформаційних масивів.

Усі зазначені сервіси Когнітивної ІТ-платформи «ПОЛІЕДР» розподіляються на дві функціонально пов'язані групи – онтолого-семантичну та онтолого-аналітичну.

Онтолого-семантична група включає сервіси лінгвoseмантичної обробки інформаційних ресурсів. Зокрема, до цієї групи належать трансдисциплінарні сервіси, орієнтовані на обробку неструктурованої текстової інформації, а саме:

1) лінгвoseмантичний аналіз текстових документів українською, російською та англійською мовами, що передбачає: формалізоване представлення

синтактико-семантичної структури речень у XML; автоматичне виділення багатослівних термінів та контекстів, що їх містять; встановлення семантичних відношень багатослівних термінів за заданими шаблонами їх описів;

2) таксономізація – процедура структурування текстових масивів на основі системологічного представлення їх термінологічної системи в ієрархічному вигляді, що має результатом експлікування структури аналізованих текстів у вигляді графа, вершинами якого є терміни, їх семантичні описи та контексти, в яких терміни фігурують;

3) рекурсивна редукція – процедура багатоетапного послідовного перетворення первинної структури тексту в онтологічний вигляд на основі виділення первинних шаблонів, що включає лексичний та синтаксичний аналіз та забезпечує автоматичну побудову таксономії, тезаурусу і онтології аналізованих текстових масивів;

4) систематизація документальних мережевих ресурсів (big data sources) – процедура, результатом якої є утворення цифрових колекцій з масиву ПТ за визначеними користувачем параметрами (мовними, понятійними, прагматичними, часовими, стильовими, функціональними, інтенціональними та ін.);

5) індексна розмітка інформаційних ресурсів – процедура, що ґрунтується на корпусних технологіях та теорії семантичних станів. Її результатом є маркування термінів з урахуванням їхньої ваги, на основі

якої здійснюється семантико-синтаксичний аналіз трансдисциплінарних ПТ.

б) встановлення міжконтекстних зв'язків – процедура знаходження термінів та словосполучень у контекстах документів, що обробляються та визначення рівня їх семантичної еквівалентності та системологічної належності; результатом застосування цієї процедури є відображення динамічного каталогу пов'язаних таксономічних структур, у контекстах яких містяться зазначені терміни та словосполучення;

7) утворення полімовної синонімічної зони – процедура, що передбачає розвиток та використання Електронного перекладного словника синонімів, який на сьогодні охоплює основну лексику української, російської, англійської, німецької, французької та італійської мов;

8) інтерактивація онтологічних документів – процедура, що забезпечує інтерактивну взаємодію з термінами та словосполученнями текстових масивів, що мають множину контекстних дефініцій, на основі їх таксономічного та онтологічного представлення. Інтерактивація забезпечує динамічну зміну порядку відображення контекстів терміносистеми тексту при взаємодії з іншими документами та інформаційними системами. Результатом інтерактивації є формування динамічного каталогу контекстів термінів, що використовуються у процесі взаємодії.

Онтолого-аналітична група сервісів включає сервіси, що забезпечують трансдисциплінарний аналіз та підтримку прийняття оптимальних організаційно-управлінських рішень в умовах складності та різноманіття ситуацій вибору. До них належать:

1) багатокритеріальний порівняльний аналіз;

2) ранжування альтернатив – визначення конкурентності альтернатив (реальних або абстрактних сутностей з притаманними їм станом, поведінкою та індивідуальністю) на основі їх статистичних характеристик;

3) конкурентна нормалізація критеріїв – відображення варіантів концептів із функціональними властивостями (об'єкти, стратегії, шляхи розвитку, прогнози та ін.) за обраною користувачем сукупністю критеріїв для прийняття оптимального рішення;

4) обернене ранжування альтернатив – пост-аналіз результатів ранжування альтернатив з метою покращення отриманих результатів та доповнення вихідної онтологічної моделі предметної області новими властивостями об'єктів дослідження;

5) експертне оцінювання – аналіз проблеми, стану об'єкта чи процесу експертним методом;

б) раціональний вибір – встановлення лінійного порядку на множині альтернатив з метою вибору найкращих за сукупністю критеріїв;

7) онтологічний інтерфейс – засіб взаємодії користувача з інформаційно-аналітичними системами,

побудованими на різних комп'ютерних платформах;

8) трансдисциплінарна категоризація контекстів – системологічне динамічне формування класів контекстних описів на основі стійких бінарних сполучень між визначеними користувачем об'єктами (терміни, фрази, словоформи тощо);

9) прогностичне оцінювання станів розвитку процесів – прогнозування перебігу процесу на основі інформації щодо його перебігу в контрольований період часу;

10) система індикаторів – формування на основі трансдисциплінарного аналізу стану інноваційних рішень, формування рейтингів об'єктів дослідження і пов'язаних з ними процесів, створення та модифікація об'єктів з урахуванням сукупності факторів впливу;

11) керуюча онтологія – операціональне розширення інтерактивного документа для вирішення завдань, що виходять за рамки онтології, шляхом підключення додаткових модулів. Цільова функція інтерактивного документа є комбінацією цільових функцій підмножини підключених модулів.

Отже, зазначені когнітивні трансдисциплінарні сервіси реалізують процеси лінгвосемантичного аналізу просторово-розподіленої неструктурованої інформації великого обсягу, її структуризацію, встановлення контекстних зв'язків між документами, прогнозування та підтримку процесів оптимального вибору з наступним формуванням інтелектуальних інформаційно-аналітичних WEB-орієнтованих рішень, забезпечують аналітичні сервіси структурування даних та відповідають критеріям сучасного інформаційно-аналітичного середовища. Більш того, використання названих сервісів дозволяє інтегрувати інформаційні ресурси в єдиний інформаційний простір.

Конфігурацію когнітивних сервісів трансдисциплінарного інформаційно-аналітичного середовища наведено на рис. 3.

Більш детальну інформацію про науково-технічні засади Когнітивної ІТ-платформи «ПОЛІЕДР» можливо дізнатися з робіт [9-15], наше завдання в контексті цієї статті – продемонструвати її застосування в лінгвоекспертній діяльності на прикладі комплексного логіко-лінгвістичного дослідження Закону України № 39-ІХ.

#### **КОМПЛЕКСНЕ ЛОГІКО-ЛІНГВІСТИЧНЕ ДОСЛІДЖЕННЯ Закону України № 39-ІХ**

За результатами дослідження встановлено, що Закон України № 39-ІХ текстуально погрунтовано на Регламенті Верховної Ради України № 1861-ІV в редакції від 10 лютого 2010 року. Автори статті встановили, що Закон України № 1861-ІV «спадкував»

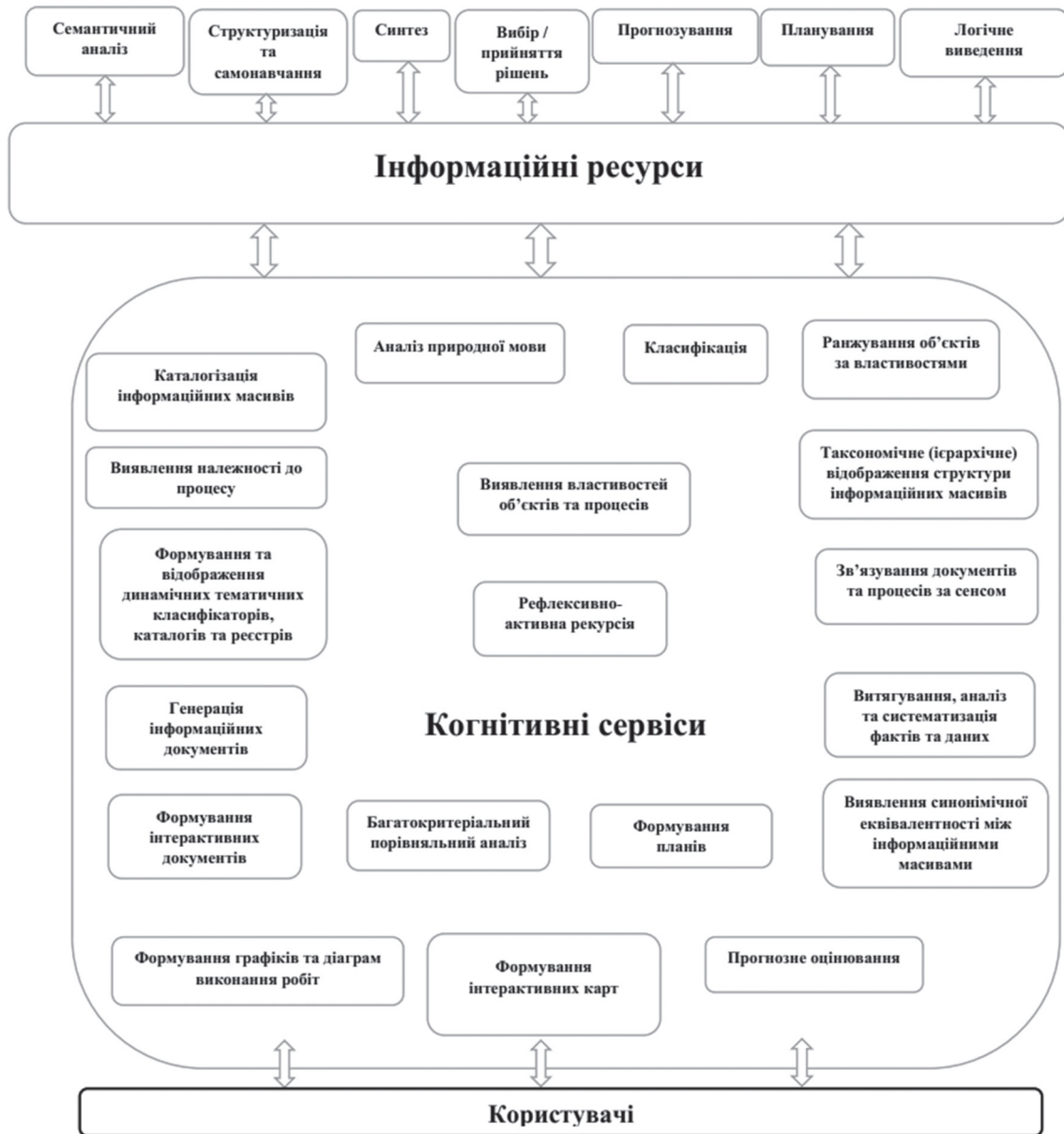


Рис. 3. Когнітивні сервіси трансдисциплінарного інформаційно-аналітичного середовища

усі ЛЛД тексту-«донора», слабкого не лише в лінгвістичному аспекті, але, як показали події 2014 року в Україні, в юридичному, оскільки процедура імпичменту, описану в Регламенті, не змогли застосувати навіть до В. Ф. Януковича, що самоусунувся від здійснення конституційних повноважень Президента України, за що був оголошений в міжнародний розшук новою українською владою 26 лютого 2014 року.

Всупереч вимогам законодавчої техніки щодо суворого дотримання норм літературної української мови при формуванні тексту нового закону «Про імпичмент Президента України», цьому ключовому показнику якості законодавчого тексту не було приділено достатньо уваги.

Недосконалість формулювань тексту Закону України № 39-IX простежується на різних рівнях мови – лексичному, синтаксичному, стилістичному, семантичному, логічному. Відсутність офіційної

інтерпретації неоднозначних виловлювань та тлумачень ключових понять (особливо – поняття «державна зрада», яке, власне, й вмотивовує процедуру імпичменту), неточне слововживання, смислова неповнота, підміна понять, хибна синонімія, порушення логічних законів (несуперечливості, достатньої підстави) робить Закон України № 39-IX уразливим не лише в лінгвістичному або логічному сенсі, але й також у юридичному.

ЛЛД представляється у вигляді загальної лексикографічної моделі [2], реєстровою одиницею якої виступає проблемний контекст із зазначенням його локалізації (позиції в тексті Закону – стаття, частина статті), інтерпретаційною частиною – поля «Тип ЛЛД» та «Підтип ЛЛД». Поле «Коментар», що зв'язує проблемний контекст із типом та підтипом встановленого в ньому ЛЛД, заповнюється користувачем вручну. Автоматизоване заповнення цього поля

можливе для орфографічних, пунктуаційних та повторюваних ЛЛД, у разі яких у полі «Коментар» буде імпортовано наявні в базі коментарі або відповідні пункти «Українського правопису».

Приклади опису встановлених ЛЛД в Законі України № 39-ІХ:

<b>Контекст:</b> Імпічмент регулюється Конституцією України, цим Законом, Регламентом Верховної Ради України, а у випадках, передбачених цим Законом, – іншими законами України.
<b>Локалізація:</b> стаття 1 частина 2
<b>Тип ЛЛД:</b> семантичний
<b>Підтип ЛЛД:</b> помилка у слововживанні (слово без урахування значення)
<b>Коментар:</b> в Конституції України та Регламенті Верховної Ради України імпічмент згадується; але процедуру імпічменту вони не регулюють. РЕГУЛЮВАТИСЯ, юється, недок. Пас. до регулювати. РЕГУЛЮВАТИ, юю, юєш, недок., перех. 1. Впорядковувати що-небудь, керувати чимось, підкоряючи його відповідним правилам, певній системі. Див. Регламент Верховної Ради України, стаття 171: «Порядок дострокового припинення повноважень Президента України у разі усунення Президента України з поста в порядку імпічменту.
1. Порядок дострокового припинення повноважень Президента України у разі усунення Президента України з поста в порядку імпічменту <b>визначається</b> Конституцією України, Законом України «Про особливу процедуру усунення Президента України з поста (імпічмент)», іншими законами України та цим Регламентом». <i>Статті 171 – 188 виключено на підставі Закону № 39-ІХ від 10 вересня 2019 року.</i>

<b>Контекст:</b> Кандидат на посаду спеціального прокурора чи спеціального слідчого подає до Верховної Ради України особову картку, декларацію особи, уповноваженої на виконання функцій держави або місцевого самоврядування, за минулий рік в порядку, визначеному Законом України «Про запобігання корупції».
<b>Локалізація:</b> стаття 4 частина 6
<b>Тип ЛЛД:</b> семантичний
<b>Підтип ЛЛД:</b> смислова неповнота
<b>Коментар:</b> відсутнє тлумачення поняття «минулий рік», що рівнозначно може розумітися як «календарний рік, що передував року, в якому розпочато процедуру імпічменту» або «рік, точкою зворотного відліку якого є подання кандидатами на посади спеціального прокурора чи спеціальних слідчих документів до Верховної Ради».

Зазначимо, що Рубрикатор ЛЛД, розроблений у Фонді, є результатом багаторічної роботи науковців, які збирають та описують колекцію ЛЛД та розвивають технологію багаторівневого маркування ЛЛД в ПТ [16].

За результатами пошуку та опису ЛЛД Закону № 39-ІХ було побудовано онтологію встановлених у ньому ЛЛД на Когнітивній ІТ-платформі «ПОЛІЕДР» (рис. 4).

Кореневими вершинами цієї онтології є найменші формально марковані структурні одиниці аналізованого Закону, що мають смислову завершеність – позначені абзацами частини його статей, в яких встановлено той чи інший ЛЛД.

Вершинами онтології другого та третього ярусів є підтипи та типи ЛЛД відповідно. Натискання на

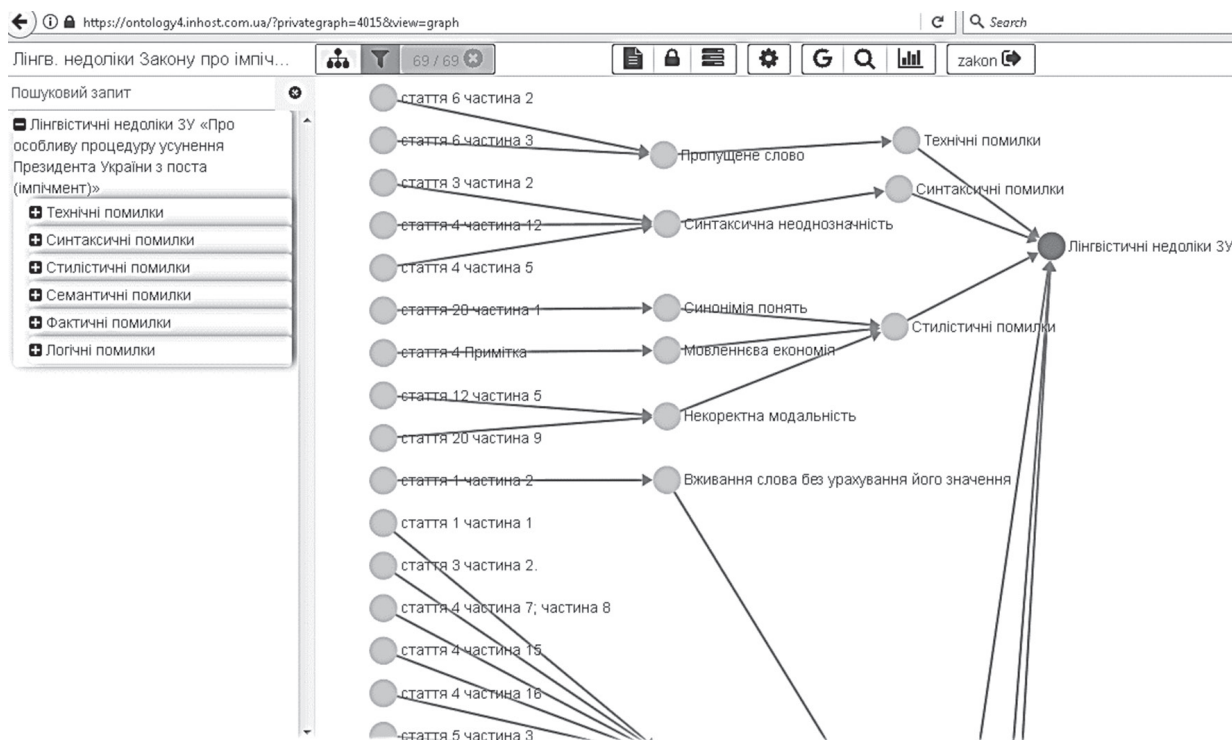


Рис. 4. Онтологія логіко-лінгвістичних дефектів Закону України «Про особливу процедуру усунення Президента України з поста (імпічмент)» № 39-ІХ від 10 вересня 2019 року, побудована на Когнітивній ІТ-платформі «ПОЛІЕДР» за результатами логіко-лінгвістичного дослідження (фрагмент)

вершини другого ярусу відкриває контекст, в якому дослідник встановив ЛЛД, натискання на вершини третього ярусу – всі контексти в документі з ЛЛД відповідного типу. Структурні елементи аналізованого тексту, в яких ЛЛД не встановлено, в онтології не фігурують. При додаванні або видаленні вершин онтологія автоматично перебудовується так, щоби уникнути перетину зв'язків (ребер).

Термінальна вершина – «Лінгвістичні недоліки ЗУ № 39-IX» об'єднує граф в одне ціле.

На нашу думку, представлення ЛЛД у вигляді онтології, завдяки своїй наочності, є надзвичайно зручним для користувачів, оскільки дозволяє *одним поглядом* оцінити лінгвістичну якість *усього* документу. Зрозуміло, як складно це зробити в традиційний спосіб – гортаючи тисячі сторінок із підкресленими редакторами помилками.

Онтологічне представлення ЛЛД також зручне у разі необхідності оцінити якість роботи кожного автора (авторського колективу), що працював над окремою структурною одиницею (глава, розділ, частина) великого тексту (регламент, національна доповідь, кодекс та ін.), оскільки візуалізує зв'язок помилки з автором проблемного контексту.

На Когнітивній ІТ-платформі «ПОЛІЕДР» також можливо проводити концептографічні дослідження ПТ (інша назва – глибокий лінгвoseмантичний аналіз), що полягають у встановленні концептів (понять,

що визначають смисл тексту), їхніх властивостей та зв'язків між ними на основі їх контекстних описів.

Екранну форму глибокого лінгвoseмантичного аналізу ПТ (у нашому прикладі – Закону України № 39-IX) наведено на рис. 5. Для зручності користувачів її поділено на три частини.

У лівій частині (позначено цифрою 1) виводиться список термінів Закону України № 39-IX зі зв'язаними термінами (верхнє вікно «Дерево термінів») та контексти, в яких зустрічаються вибрані користувачем терміни в Законі України № 39-IX (нижнє вікно «Речення тексту»). При активації опції «підкреслити терміни» у вікні контекстів обрані користувачем терміни підкреслюються.

У центральній частині (позначено цифрою 2) виводиться перелік усіх термінів із зазначенням кількості входжень в аналізований текст та формою пошуку терміну за типом, що задається в умовах пошуку вибором потрібного типу зі списку.

У правій частині екранної форми (позначено цифрою 3) знаходиться зведена таблиця класів термінів кожної зі статей Закону України № 39-IX (рис. 6), сформована за такою схемою:

(клас термінів A-Z), (термін A), (термін B), (термін C), ..., (термін Z)

(термін C – клас термінів C<sub>1-n</sub>), (термін C<sub>1</sub>), (термін C<sub>2</sub>), (термін C<sub>3</sub>), ..., (термін C<sub>n</sub>)

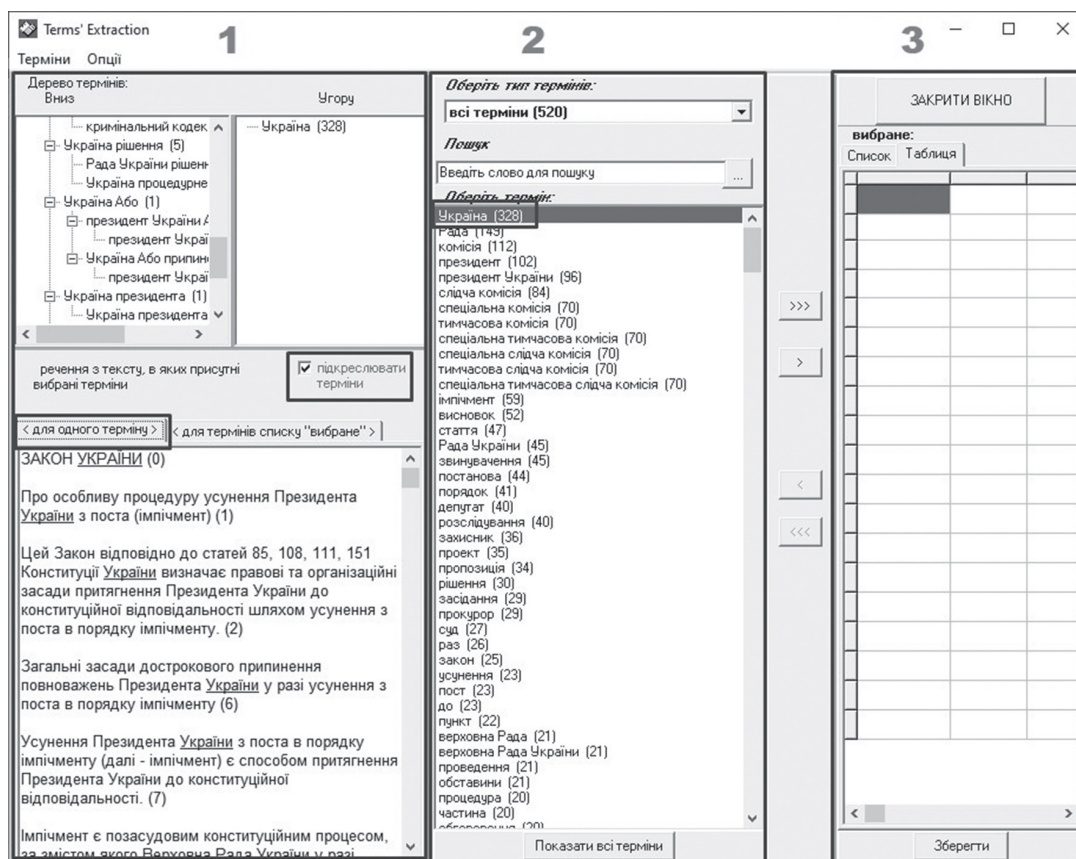


Рис. 5. Екранна форма глибокого лінгвoseмантичного аналізу Закону України № 39-IX, виконаного на Когнітивній ІТ-платформі «ПОЛІЕДР»

1	Закон України Про імпичмент	Закон України Про імпичмент	Україна	комісія	імпичмент	висновок	постанова	звинувачення	захисник	порядок
2	Україна	Україна	Президент України	Верховна Рада України	Конституція України	народний депутат України	громадянин України	Закон України	Конституційний Суд України	Кримінальний кодекс України
3	Президент України	Президент України	усунення Президента України	припинення повноважень Президента України	Президентом України державної зради	захисник прав Президента України	звинувачення Президента України	запитання Президента України	зауваження Президента України	пояснення Президента України
4	усунення Президента України	усунення Президента України	процедура усунення Президента України	прийняття рішення про усунення Президента України з поста в порядку імпичменту	неприйняття рішення про усунення Президента України з поста в порядку імпичменту					

Рис. 6. Права частина екранної форми глибокого лінгвoseмантичного аналізу Закону України № 39-IX презентує класи термінів кожної зі статей Закону в табличному вигляді

(термін  $C_2$  — клас термінів  $C_{21-n}$ ), (термін  $C_{21}$ ), (термін  $C_{22}$ ), (термін  $C_{23}$ ),..., (термін  $C_{2n}$ ).

За результатами концептографічного дослідження Закону України № 39-IX за допомогою Конструктора онтологій Когнітивної ІТ-платформи «ПОЛІЕДР» було побудовано онтологію – універсальну модель представлення знань про предметну галузь «імпичмент Президента України», вилучених із цього Закону, що дозволяє проводити будь-які операції над цими знаннями, включаючи їх транслявання до іншого програмного середовища (рис. 7).

Побудована за результатами концептографічного дослідження Закону України № 39-IX онтологія дозволила виявити сім невизначеностей (колізій) у його тексті.

Кожна невизначеність (колізія) представляється програмній системі, що не розуміє смислу тексту, як неповнота або надлишковість зв'язків між вершинами онтології. Оскільки зазначені відхилення безпосередньо обумовлюються змістово-логічною неповнотою або надлишковістю аналізованого тексту, встановлені Когнітивною ІТ-платформою

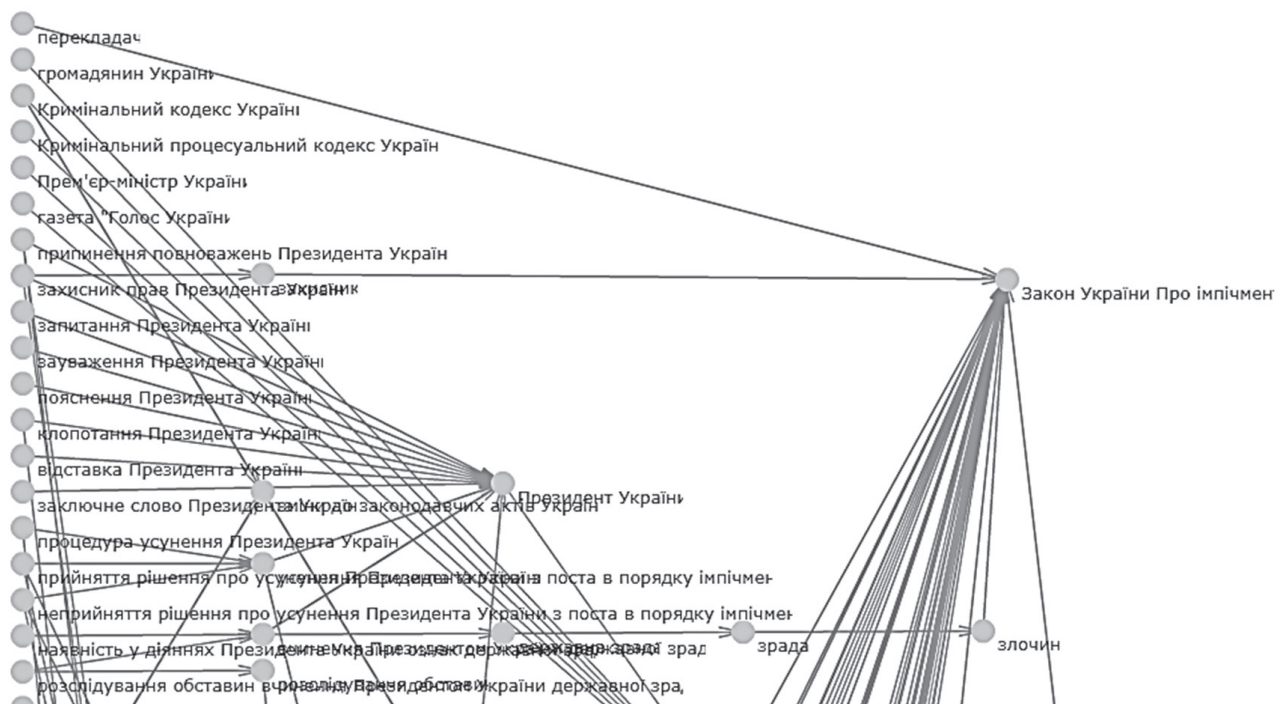


Рис. 7. Онтологія Закону України «Про особливу процедуру усунення Президента України з поста (імпичмент)» № 39-IX от 10 вересня 2019 року, побудована на Когнітивній ІТ-платформі «ПОЛІЕДР» (фрагмент)

«ПОЛІЕДР» невизначеності (колізії) можливо вважати об'єктивними фактами.

Змістовно-логічна неповнота та надлишковість тексту закону однаково призводять до неоднозначної інтерпретації нормативних приписів, що, як свідчить практика, завжди закінчується конфліктом інтересів у правозастосуванні.

Деталізовані фрагменти онтології Закону України № 39-ІХ, наведені нижче, демонструють колізії, встановлені Когнітивною ІТ-платформою «ПОЛІЕДР» на основі неповноти зв'язків між вершинами онтології (рис. 8) та надлишковості зв'язків між вершинами онтології (рис. 9). Колізії позначено символом \$.

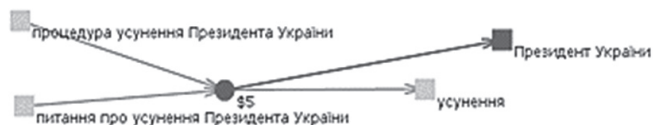


Рис. 8. Фрагмент онтології Закону № 39-ІХ з експлікованою смисловою лакуною, зумовленою відсутністю у тексті жодних вказівок стосовно подальших дій Президента України, усуненого з посту в порядку імпічменту

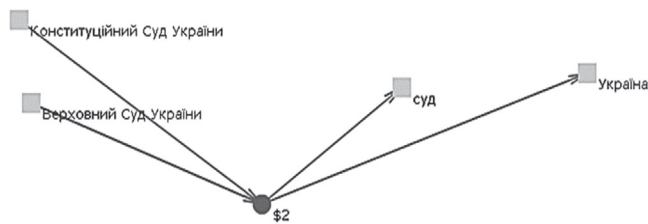


Рис. 9. Фрагмент онтології Закону № 39-ІХ, що вказує на потенційний конфлікт інтересів вищих судових органів України при прийнятті рішень у справі про імпічмент Президента України

Когнітивна ІТ-платформа «ПОЛІЕДР» надає користувачам нові інструменти і засоби дослідження законодавства, що дозволяють експлікувати потенційні джерела конфліктів у правозастосуванні – неоднозначні положення, терміни та поняття з некоректним або відсутнім офіційним тлумаченням.

Слід також зазначити, що сервіси Когнітивної ІТ-платформи «ПОЛІЕДР», що забезпечують концептографічне дослідження ПТ, реалізують відображення зв'язності його концептів у форматі нарративного дискурсу [11]. Іншими словами, реалізуються когнітивно-комунікативні акти взаємодії між контекстами кожного концепту онтології аналізованих текстів. Це дозволяє виявляти семантичну зв'язність між контекстами концептів, з якою пов'язується логічна зв'язність.

Сервіси Когнітивної ІТ-платформи «ПОЛІЕДР» забезпечують агрегування всіх контекстів досліджуваних концептів, відображення класів еквівалентності законодавчих актів за різними ознаками (смыслова близькість концептів / понять / норм; гіпо-гіперонімічні ланцюжки концептів та ін.), відображення та відстеження логіки взаємовпливу процесів, що

регулюються аналізованими нормативно-правовими актами, побудова онтології аналізованих нормативно-правових актів із можливістю деталізації фрагментів.

## ВИСНОВКИ

Проведене в Українському мовно-інформаційному фонді НАН України комплексне логіко-лінгвістичне дослідження Закону України «Про особливу процедуру усунення Президента України з посту (імпічмент)» № 39-ІХ от 10 вересня 2019 року із застосуванням авторських програмних продуктів – Системи аналізу текстів та Когнітивної ІТ-платформи «ПОЛІЕДР», що реалізують корпусні та концептографічні технології обробки та дослідження ПТ, а також технології онтологічного моделювання ПТ, показало, що попри політичну та соціальну значимість цього закону, в ньому мають місце логіко-лінгвістичні дефекти, що проявляються на різних рівнях функціонування тексту – синтаксичному, семантичному, стилістичному, логічному, концептологічному.

Звідси випливає цілком очевидний висновок: усунення недоліків вітчизняного законодавства є наріжною проблемою, розв'язання якої вимагає комплексного підходу. Крім розробки та впровадження у сферу законотворчості сучасних когнітивно-інформаційних комп'ютерних інструментів реалізація цього підходу потребує ретельного аналізу та, можливо, навіть перегляду структури, модусу, змісту та інших характеристик законотворчого процесу. Аналогічні твердження справедливі й щодо до правозастосовної практики.

## Список літератури:

- [1] Теорія лексикографічних систем: Широков В. А. Інформаційна теорія лексикографічних систем. – К.: Довіра, 1998. – 331 с. В.А.Широков та ін. Лінгвістично-інформаційні студії: Праці Українського мовно-інформаційного фонду НАН України: у 5 т. Том 1. Наукова парадигма та основні мовно-інформаційні структури. Київ, Український мовно-інформаційний фонд НАН України. 2018. 271 с. ISBN 978-966-02-8683-2. ISBN 978-966-02-8684-9 (Т. 1). DOI 10.33190/978-966-02-8683-2/8684-9. [https://movoznavstvo.org.ua/files/tom\\_1\\_B5\\_print.pdf](https://movoznavstvo.org.ua/files/tom_1_B5_print.pdf).
- [2] Теорія семантичних станів: Широков В. А. Феноменологія лексикографічних систем. – К.: Наук. думка, 2004. – 327 с.; В.А.Широков та ін. Лінгвістично-інформаційні студії: Праці Українського мовно-інформаційного фонду НАН України: у 5 т. Том 1. Наукова парадигма та основні мовно-інформаційні структури. Київ, Український мовно-інформаційний фонд НАН України. 2018. 271 с. ISBN 978-966-02-8683-2. ISBN 978-966-02-8684-9 (Т. 1). DOI 10.33190/978-966-02-8683-2/8684-9. [https://movoznavstvo.org.ua/files/tom\\_1\\_B5\\_print.pdf](https://movoznavstvo.org.ua/files/tom_1_B5_print.pdf).
- [3] Широков В. А. Язык. Информация. Система. Palmarium Academic Publishing, 2017. ISBN 978-3-659-72403-9. 280 с.;

- [4] Широков В. А. . Феноменологические основания для грамматического описания языка. В кн. «Когнитивно-семиотические аспекты моделирования в гуманитарной сфере». Под. ред. В. Л. Стефанюка, Э. А. Тайсиной. Институт прикладной семиотики Академии наук Татарстана. ISBN 978-5-9690-0386-6. Изд-во АН Татарстана. 2017, С. 98 – 144.
- [5] Теорія слабкої синонімії: В. А. Широков та ін. Корпусна лінгвістика. – К.: Довіра, 2005, С. 390 – 393.
- [6] Правила оформления проектов законов и основные требования законодательной техники. Методические рекомендации Аппарата Верховной Рады Украины. Издание четвертое, исправленное и дополненное. Киев. 2014. – 39 с. Режим доступа: [http://static.rada.gov.ua/site/bills/info/zak\\_rules.pdf](http://static.rada.gov.ua/site/bills/info/zak_rules.pdf).
- [7] Лінгвістично-інформаційні студії: праці Українського мовно-інформаційного фонду НАН України: у 5 т. / В. А. Широков, М. В. Надутенко, С. С. Ющенко та ін. Т. 4.: Корпусна та когнітивна лінгвістика. – Київ: Український мовно-інформаційний фонд НАН України, 2018. – 280 с.
- [8] Крыгин М., Широков В. Корпусные технологии при проведении лингвистических экспертиз. Прикладна лінгвістика та лінгвістичні технології. Зб. наук. праць. НАН України, Укр. мовно-інформ. фонд [та ін.] / редкол.: Л. П. Крисін [та ін.]. Київ: Довіра, 2010. С. 130-140.
- [9] Oleksandr Y. Stryzhak, Hrihorii M. Potapov, Vitalii V. Prychodniuk, Roman I. Chepков. Еволюція управління – від ситуаційного до трансдисциплінарного. В зб. наук. праць «Екологічна безпека та природокористування». Том 30, № 2 (2019).
- [10] Приходнюк В. В, Стрижак О. Є, Гайко С. І., Чепков Р. І. Інформаційно-аналітичний комплекс підтримки процесів трансдисциплінарних досліджень. В зб. наук. праць «Екологічна безпека та природокористування», 2018.
- [11] Dovgyi S., Stryzhak O. (2020) Transdisciplinary Fundamentals of Information-Analytical Activity. In: Ilchenko M., Uryvsky L., Globa L. (eds) Advances in Information and Communication Technology and Systems. MCT 2019. Lecture Notes in Networks and Systems, vol 152. Springer, Cham. [https://doi.org/10.1007/978-3-030-58359-0\\_7](https://doi.org/10.1007/978-3-030-58359-0_7).
- [12] Онтологические методы и средства обработки предметных знаний: монография / А. В. Палагин, С. Л. Крывый, Н. Г. Петренко. – Луганск: изд-во ВНУ им. В. Даля, 2012. – 324 с.
- [13] Volodymyr Shyrokov, Iryna Ostapova, Maksym Nadutenko, Yulia Verbynenko. Ontologized lexicographical systems in modern terminography // Cognitive Studies | Etudes cognitives. – Warsaw, 2016. – P. 90–99.
- [14] Надутенко Максим, Надутенко Маргарита. Специфікація основних системних компонентів лінгвостатистичних засобів концептографічного маркування інформаційних ресурсів // Методологія та історіографія мовознавства: матеріали VI науково-практичної Інтернет-конференції (Слов'янськ, 23–24 жовтня 2019 р.). – Слов'янськ : ДДПУ, 2019. – С. 79–82.
- [15] Надутенко М. В. Загальний огляд та перспективи використання національних лінгвістичних цифрових ресурсів Українського мовно-інформаційного фонду НАН України // Зб. праць конфер. Національний університет «Києво-Могилянська академія», 2020. – <http://ekmair.ukma.edu.ua/handle/123456789/17607>
- [16] Ющенко С. Концепція рубризатора мовних девицій в актах первинного законодавства України. Прикладна лінгвістика та лінгвістичні технології. Зб. наук. праць. Редкол.: Л. П. Крисін [та ін.]. НАН України, Укр. мовно-інформ. фонд [та ін.]. Київ: Довіра, 2010. С. 234–242. <http://megaling.ulif.org.ua/tezi-2013-rik/yuschenko-svetlana>.

Надійшла до редколегії 18.09.2020



Ivan Bozhko<sup>1</sup>, Grygoriy Chetverykov<sup>2</sup>, Oleksandr Karataiev<sup>3</sup>

<sup>1</sup> Kharkiv National University of Radio Electronics, Kharkiv, Ukraine,  
ivan.bozhko@nure.ua

<sup>2</sup> Kharkiv National University of Radio Electronics, Kharkiv, Ukraine,  
grigorij.chetverykov@nure.ua

<sup>3</sup> Kharkiv National University of Radio Electronics, Kharkiv, Ukraine,  
oleksandr.karataiev@nure.ua

## THE MODERN QUANTUM COMPUTING TOOLS INVESTIGATION

The current paper covers the investigation of the current state of the existing tools for quantum programming including QCL (Quantum Computation Language), quantum pseudocode, Q# programming language and Quipper. Since quantum computing is one of the main research areas today, the respective tools are being created quite often. They are aimed on simplifying the development of quantum programs, on the one hand, and provide some platform for testing and running them, on the other hand. So, the authors investigated the currently available tools and provided the results in the article.

QUANTUM COMPUTING, QUANTUM COMPUTER, Q#, QUANTUM COMPUTING TOOLS, MICROSOFT QUANTUM DEVELOPMENT KIT

**Божко І.К., Четвериков Г.Г., Каратаєв О.А. Дослідження сучасних засобів квантових обчислень.** Дана робота являє собою дослідження поточного стану існуючих інструментів квантових обчислень, особливо мови програмування Q# як найбільш розвинутого інструменту для цього в даний час. Оскільки квантові обчислення сьогодні є однією з основних областей досліджень, створюються відповідні інструменти. Вони покликані спростити розробку квантових програм, з одного боку, і надати платформу для тестування і запуску їх, з іншого боку. Тому автори дослідили наявні в даний час інструменти і представили результати в даній статті.

КВАНТОВІ ОБЧИСЛЕННЯ, КВАНТОВИЙ КОМП'ЮТЕР, Q#, ІНСТРУМЕНТИ КВАНТОВИХ ОБЧИСЛЕНЬ, MICROSOFT QUANTUM DEVELOPMENT KIT

**Божко И.К., Четвериков Г.Г., Каратаев А.А. Исследование современных средств квантовых вычислений.** Данная работа представляет исследование текущего состояния существующих инструментов квантовых вычислений, особенно языка программирования Q# как самого развитого инструмента для этого в настоящее время. Поскольку квантовые вычисления сегодня являются одной из основных областей исследований, создаются соответствующие инструменты. Они призваны упростить разработку квантовых программ, с одной стороны, и предоставить платформу для тестирования и запуска их, с другой стороны. Поэтому авторы исследовали имеющиеся в настоящее время инструменты и представили результаты в данной статье.

КВАНТОВЫЕ ВЫЧИСЛЕНИЯ, КВАНТОВЫЙ КОМПЮТЕР, Q#, ИНСТРУМЕНТЫ КВАНТОВЫХ ВЫЧИСЛЕНИЙ, MICROSOFT QUANTUM DEVELOPMENT KIT

### Introduction

The research of the quantum computer area started for the first time back in 1980 by the Soviet mathematician Yu.I. Manin [1]. However, more interest in this type of calculation arose only in 1982, after the American theoretical physicist Richard Feynman noticed that not all quantum-mechanical operations can be accurately transferred to a classical computer and more efficiently carried out by quantum operations.

An additional relevance to the quantum computing problem was added by mathematician Peter Shore, who in 1994 proposed an algorithm that allows the expansion of a  $n$ -valued number to simple multipliers with polynomial complexity. On classical computers, this task is much more complex and does not allow you to get the result for a satisfactory time.

Since this task is the basis of many popular cryptographic algorithms (for example, RSA) [2], the creation of quantum computers may influence the security of the data exchange in the network, and after the appearance of a real prototype of a quantum computer, it can become a global security issue.

Consequently, quantum computing is relevant not only for scientific problems of quantum processes modeling, but also relevant to the world of information technology. In order to popularize them not only in the scientific world, but also among developers, tools are created to simplify the work with quantum algorithms, such as the programming language Q#, which is considered in this paper.

### 1. Existing quantum computing tools

Over the past few years, with the growing popularity of quantum computing research, tools and emulators of quantum computers have begun to appear that allow you to try calculations to practice. Here are some popular tools and their descriptions.

#### 1.1. Microsoft Quantum Development Kit

Microsoft has released a preview version of Quantum Development Kit, which includes the new quantum programming language Q#, integration with the Visual Studio development environment, simulators that work with both the local system and their powerful Azure cloud platform, as well as libraries and code samples that can be used as constructive blocks.

## 1.2. IBM Quantum Experience

IBM has created an experimental quantum 5-qubit processor that is available to users through the Internet [2]. On the IBM Quantum Experience website, you can find a short tutorial that explains the basics of quantum computing and system usage instructions, the configuration of the queue access interfaces, a simulator that allows you to simulate their configuration before running it on the actual machine, and access to the machine itself that allows you to run the configuration and view the results.

## 1.3. Rigetti Forest

The Rigetti Forest package consists of the instructions based quantum language called Quil, an open Python library for building Quil programs called pyQuil, a quantum library called Grove, and a simulation environment called QVM (Quantum Virtual Machine). pyQuil and Grove are open source programs available on Github. Users can develop their applications using pyQuil and Grove on their own computer, and then transfer them to QVM for simulation through a web portal that is available to registered users.

## 1.4. ProjectQ

ProjectQ is an open source software for quantum computing, implemented in Python. This allows users to implement their quantum programs in Python using a powerful and intuitive syntax. ProjectQ can then broadcast these programs to any server part: a simulator that runs on a classic computer, or a quantum computer (for example, using IBM Quantum Experience). Other hardware platforms are currently not supported.

In addition to these tools, there are others such as Cirq, Quirk, QuTiP, but they are less powerful than those described above.

As can be seen from the description of existing solutions, most of them are complementary to existing programming languages (in particular, Python), but given the new paradigm of computations, this may cause difficulties in programming algorithms using these tools, therefore, a more flexible and powerful solution is a separate programming language, which currently only Microsoft offers.

Also, it is necessary to highlight the Intel solution with the existing experimental quantum computer, but it requires execution of queries to individual qubits through the API, which complicates its use.

The short comparison could be seen in the table 1.1.

**Table 1.1**

The comparison of the quantum computing tools

Feature\ Technology	IBM Q Exp.	Rigetti Forest	ProjectQ	Q#
Language	-	-	-	+
API Access	+	+	-	+
Integration with other languages	-	+	+	+
Availability for different platforms	+	-	-	-
Possibility to use real computer instead of emulators	+	-	-	-

## 2. The Q# language

As discussed above, one of the parts of Microsoft Quantum Development Kit is a Q# language specifically designed for quantum computing. In terms of software engineering, this solution is most interesting as it allows abstracting from the paradigm of classical computing and classical programming languages and describing a quantum algorithm using a special syntax.

Consider the language Q # in more detail.

### 2.1. Computing model

According to official Microsoft documentation [4], a natural model for quantum computation is to treat the quantum computer as a coprocessor, similar to that used for GPUs, FPGAs, and other adjunct processors. The primary control logic runs classical code on a classical "host" computer. When appropriate and necessary, the host program can invoke a sub-program that runs on the adjunct processor. When the sub-program completes, the host program gets access to the sub-program's results.

In this model, there are three levels of computation:

- Classical computation that reads input data, sets up the quantum computation, triggers the quantum computation, processes the results of the computation, and presents the results to the user.
- Quantum computation that happens directly in the quantum device and implements a quantum algorithm.
- Classical computation that is required by the quantum algorithm during its execution.

There is no intrinsic requirement that these three levels all be written in the same language. Indeed, quantum computation has somewhat different control structures and resource management needs than classical computation, so using a custom programming language allows common patterns in quantum algorithms to be expressed more naturally.

Keeping classical computations separate means that the quantum programming language may be very constrained. These constraints may allow better optimization or faster execution of the quantum algorithm.

Q# (Q-sharp) is a domain-specific programming language used for expressing quantum algorithms. It is to be used for writing sub-programs that execute on an adjunct quantum processor, under the control of a classical host program and computer.

Q# provides a small set of primitive types, along with two ways (arrays and tuples) for creating new, structured types. It supports a basic procedural model for writing programs, with loops and if/then statements. The top-level constructs in Q# are user defined types, operations, and functions.

### 2.2. Q# type system

The Q# language provides a small set of primitive types, as well as two methods (arrays and corrections) for creating new types of data (it means that the language have

a bit more constraints than classical languages, which was made for optimization purposes). In general, the language usually supports a procedural conditional programming model (if -this) and cycles.

Let us consider the primitive data types which other types consist of:

- The int type represents a 64-bit signed (two's complement) integer.
- The double type represents a double-precision floating-point number.
- The bool type represents a Boolean value, either true or false.
- The qubit type represents a quantum bit or qubit. They are opaque to the user; the only operation possible with them, other than passing them to another operation, is to test for identity (equality). Ultimately, actions on Qubits are implemented by calling operations in the Q# standard library.
- The Pauli type represents an element of the single-qubit Pauli group. This type is used to denote the base operation for rotations and to specify the basis of a measurement. This type is a discriminated union with four possible values: PauliI, PauliX, PauliY and PauliZ.
- The Result type represents the result of a measurement. This type is a discriminated union with two possible values: One and Zero. Zero indicates that the +1 eigenvalue was measured; One indicates the -1 eigenvalue.
- The Range type represents a sequence of integers.
- The String type is a sequence of Unicode characters that is opaque to the user once created. This type is used to report messages to a classical host.

It is also necessary to note that having the type system described above means a set of reserved keywords: true, false, PauliI, PauliX, PauliY, PauliZ, Zero and One.

Outside of the primitive types, there are also another types we will consider now in more detail.

Given any valid Q# type 'T there is a type that represents an array of values of type 'T. This array type is represented as 'T[] for example, Qubit[] or Int[][].

In the second example, note that this represents a potentially jagged array of arrays, and not a rectangular two-dimensional array. Q# does not include support for rectangular multi-dimensional arrays.

Given any valid Q# types 'T1, 'T2, 'T3, etc., there is a type that represents a tuple of values of types 'T1, 'T2, 'T3, etc., respectively. This tuple type is represented as ('T1, 'T2, 'T3, ...). Any number of types may be tupled together. The empty tuple, (), is equivalent to unit in F#.

It is possible to create arrays of tuples, tuples of arrays, tuples of sub-tuples, etc.

Tuple instances are immutable. Q# does not provide a mechanism to change the contents of a tuple once created.

It is also possible to create a singleton (single-element) tuple, ('T1), such as (5) or ([1;2;3]). However, Q# treats a singleton tuple as completely equivalent to a value of the enclosed type. That is, there is no difference between 5 and (5), or between 5 and (((5))), or between (5, (6)) and (5, 6).

This equivalence applies for all purposes, including assignment and expressions. It is just as valid to write (5)+3 as to write 5+3, and both expressions will evaluate to 8. We refer to this property as singleton tuple equivalence.

There is also a possibility for creating user-defined types. A Q# file may define a new named type based on a standard type. Any legal type may be used as the base for a user-defined type.

User-defined types may be used anywhere any other type may be used. In particular, it is possible to define an array of a user-defined type and to include a user-defined type as an element of a tuple type.

It is not possible to create recursive type structures. That is, the type that defines a user-defined type may not be a tuple type that includes an element of the user-defined type. More generally, user-defined types may not have cyclic dependencies on each other.

The mutability of instances of user-defined types is the same as the mutability of instances of the base type of the user-defined type. Specifically, instances of user-defined types based on tuples are immutable; instances of user-defined types based on arrays are potentially mutable.

Effectively, a user-defined type is a subtype of the base type. Thus, a value of a user-defined type may be used anywhere a value of the base type is expected. This is applied recursively.

For example, suppose type IntPair is a user-defined type with base type (Int, Int), and type IntPair2 is a user-defined type with base type IntPair. A value of type IntPair2 may be used anywhere a value of type IntPair2, IntPair, or (Int, Int) is expected. A value of type IntPair may be used anywhere a value of type IntPair or (Int, Int) is expected.

Different user-defined types based on the same base type are treated as distinct and unrelated types. In the previous example, if IntPair3 is also a user-defined type with base type (Int, Int), then IntPair and IntPair3 are unrelated and a value of one may not be used where a value of the other is expected.

A Q# operation is a quantum subroutine. That is, it is a callable routine that contains quantum operations.

A Q# function is a classical subroutine used within a quantum algorithm. It may contain classical code but no quantum operations. Functions may not allocate or borrow qubits, nor may they call operations. It is possible, however, to pass them operations or qubits for processing.

Together, operations and functions are known as callables.

All Q# callables are considered to take a single value as input and return a single value as output. Both the input

and output values may be tuples. Callables that have no result return the empty tuple, (); callables that have no input take the empty tuple as input.

The basic signature for any callable is written as ('Tinput => 'Tresult) or ('Tinput -> 'Tresult), where both 'Tinput and 'Tresult are type specifiers. The first form, with =>, is used for operations; the second form, with ->, for functions. For example, ((Qubit, Pauli) => Result) represents the signature for a possible single-qubit measurement operation.

Function types are completely specified by their signature. For example, a function that computes the sine of an angle would have type (Double -> Double).

Operations – but not functions – may allow the application of one or more functors. Functors are meta-operations that generate a variant of a base operation.

Operation types are specified by their signature and the list of functors they support. For example, the Pauli X operation has type (Qubit => ()) : Adjoint, Controlled). An operation type that does not support any functors is specified by its signature alone, with no trailing :

Callable signatures may contain type parameters. Type parameters are indicated by a symbol prefixed by a single quote; for example, 'A is a legal type parameter. Type-parameterized functions and operations are similar to generic functions in many programming languages, but Q# does not provide a full generic type/function capability.

A type parameter may appear more than once in a single signature. For example, a function that applies another function to each element of an array and returns the collected results would have signature (('A[], 'A->'A) -> 'A[]). Similarly, a function that returns the composition of two operations might have signature (('A=>'B), ('B=>'C)) -> ('A=>'C)).

When invoking a type-parameterized callable, all arguments that have the same type parameter must be of the same type, or be compatible with the same type; that is.

Q# does not provide a mechanism for constraining the possible types that might be substituted for a type parameter. Thus, type parameters are primarily useful for functions on arrays and for composing callables.

An operation with additional functors supported may be used anywhere an operation with fewer functors but the same signature is expected. For instance, an operation of type (Qubit=>():Adjoint) may be used anywhere an operation of type (Qubit=>()) is expected.

Q# is covariant with respect to callable return types: a callable that returns a type 'A is compatible with a callable with the same input type and a result type that 'A is compatible with.

Q# is contravariant with respect to input types: a callable that takes a type 'A as input is compatible with a callable with the same result type and an input type that is compatible with 'A.

A functor in Q# is a factory that defines a new operation from another operation. Functors have access to the implementation of the base operation when defining the implementation of the new operation. Thus, functors can perform more complex functions than traditional higher-level functions.

A functor is used by applying it to an operation, returning a new operation. For example, the operation that results from applying the Adjoint functor to the Y operation is written as (Adjoint Y). The new operation may then be invoked like any other operation. Thus, (Adjoint Y)(q1) applies the adjoint functor to the Y operation to generate a new operation, and applies that new operation to q1. Similarly, (Controlled X)(controls, target).

The two standard functors in Q# are Adjoint and Controlled.

In quantum computing, the adjoint of an operation is the complex conjugate transpose of the operation. For operations that implement a unitary operator, the adjoint is the inverse of the operation. For a simple operation that just invokes a sequence of other unitary operations on a set of qubits, the adjoint may be computed by applying the adjoints of the sub-operations on the same qubits, in the reverse sequence.

Given an operation expression, a new operation expression may be formed using the Adjoint functor, with the base operation expression enclosed in parentheses, ( and ). The new operation has the same signature and type as the base operation. In particular, the new operation also allows Adjoint, and will allow Controlled if and only if the base operation did.

For instance, (Adjoint QFT) designates the adjoint of the QFT operation.

The controlled version of an operation is a new operation that effectively applies the base operation only if all of the control qubits are in a specified state. If the control qubits are in superposition, then the base operation is applied coherently to the appropriate part of the superposition. Thus, controlled operations are often used to generate entanglement.

In Q#, controlled versions always take an array of control qubits, and the specified state is always for all of the control qubits to be in the computational (PauliZ) One state,  $|1\rangle|1\rangle$ . Controlling based on other states may be achieved by applying the appropriate Clifford operations to the control qubits before the controlled operation, and then applying the inverses of the Cliffords after the controlled operation. For example, applying an X operation to a control qubit before and after a controlled operation will cause the operation to control on the Zero state ( $|0\rangle|0\rangle$ ) for that qubit; applying an H operation will control on the PauliX Zero state  $|+\rangle:=(|0\rangle+|1\rangle)/\sqrt{2}$ ;  $|+\rangle:=(|0\rangle+|1\rangle)/2$  rather than the PauliZ Zero state.

Given an operation expression, a new operation expression may be formed using the Controlled functor,

with the base operation expression enclosed in parentheses, ( and ). The signature of the new operation is based on the signature of the base operation. The result type is the same, but the input type is a two-tuple with a qubit array that holds the control qubit(s) as the first element and the arguments of the base operation as the second element. If the base operation took no arguments, (), then the input type of the controlled version is just the array of control qubits. The new operation allows `Controlled`, and will allow `Adjoint` if and only if the base operation did.

If the base function took only a single argument, then singleton tuple equivalence will come into play here. For instance, `Controlled(X)` is the controlled version of the `X` operation. `X` has type `(Qubit => () : Adjoint, Controlled)`, so `Controlled(X)` has type `((Qubit[], (Qubit)) => () : Adjoint, Controlled)`; because of singleton tuple equivalence, this is the same as `((Qubit[], Qubit) => () : Adjoint, Controlled)`.

Similarly, `Controlled(Rz)` is the controlled version of the `Rz` operation. `Rz` has type `((Double, Qubit) => () : Adjoint, Controlled)`, so `Controlled(Rz)` has type `((Qubit[], (Double, Qubit)) => () : Adjoint, Controlled)`. For example, `((Controlled(Rz))(controls, (0.1, target))` would be a valid invocation of `Controlled(Rz)`.

As another example, `CNOT(control, target)` can be implemented as `(Controlled(X))([control], target)`. If a target should be controlled by 2 control qubits (`CCNOT`), we can use `(Controlled(X))([control1;control2], target)` statement.

### 2.3. Q# expressions

Expressions is an important part of any programming language and Q# is not an exception. Let us consider some of the important expressions.

Given any expression, that same expression enclosed in parentheses is an expression of the same type. For instance, `(7)` is an `Int` expression, `([1;2;3])` is an expression of type array of `Ints`, and `((1,2))` is an expression with type `(Int, Int)`.

The equivalence between simple values and single-element tuples removes the ambiguity between `(6)` as a group and `(6)` as a single-element tuple.

The name of a symbol bound or assigned to a value of type `T` is an expression of type `T`. For instance, if the symbol `count` is bound to the integer value 5, then `count` is an integer expression.

Numeric expressions are expressions of type `Int` or `Double`. That is, they are either integer or floating-point numbers.

`Int` literals in Q# are identical to integer literals in C#, except that no trailing "l" or "L" is required (or allowed). Hexadecimal integers are supported with a "0x" prefix.

`Double` literals in Q# are identical to double literals in C#, except that no trailing "d" or "D" is required (or allowed).

Given an array expression of any element type, an `Int` expression may be formed using the `Length` built-in

function, with the array expression enclosed in parentheses, ( and ). For instance, if `a` is bound to an array, then `Length(a)` is an integer expression. If `b` is an array of arrays of integers, `Int[][]`, then `Length(b)` is the number of sub-arrays in `b`, and `Length(b[1])` is the number of integers in the second sub-array in `b`.

Given two numeric expressions, the binary operators `+`, `-`, `*`, and `/` may be used to form a new numeric expression. The type of the new expression will be `Double` if both of the constituent expressions are `Double`, or will be an `Int` expression if both are integers.

Given two integer expressions, a new integer expression may be formed using the `%` (modulus), `^` (power), `&&&` (bitwise AND), `|||` (bitwise OR), `^^^` (bitwise XOR), `<<<` (arithmetic left shift), or `>>>` (arithmetic right shift) operations. The second parameter to either shift operation must be greater than or equal to zero. The behavior for shifting negative numbers is undefined.

Given any numeric expression, a new expression may be formed using the `-` unary operator. The new expression will be the same type as the constituent expression.

Given any integer expression, a new integer expression may be formed using the `~~~` (bitwise complement) unary operator.

The only `Qubit` expressions are symbols that are bound to `Qubit` values or array elements of `Qubit` arrays. There are no `Qubit` literals.

The four Pauli values, `PauliI`, `PauliX`, `PauliY`, and `PauliZ`, are all valid Pauli expressions.

Other than that, the only Pauli expressions are symbols that are bound to Pauli values or array elements of Pauli arrays.

The two `Result` values, `One` and `Zero`, are valid `Result` expressions.

Other than that, the only `Result` expressions are symbols that are bound to `Result` values or array elements of `Result` arrays. In particular, note that `One` is not the same as the integer 1, and there is no direct conversion between them. The same is true for `Zero` and 0.

Given any three `Int` expressions `start`, `step`, and `stop`, `start .. step .. stop` is a range expression whose first element is `start`, second element is `start+step`, third element is `start+step+step`, etc., until `stop` is passed. A range may be empty if, for instance, `step` is positive and `stop < start`. The last element of the range will be `stop` if the difference between `start` and `stop` is an integral multiple of `step`; that is, the range is inclusive at both ends.

Given any two `Int` expressions `start` and `stop`, `start .. stop` is a range expression that is equal to `start .. 1 .. stop`. Note that the implied `step` is `+1` even if `stop` is less than `start`; in such a case, the range is empty. For example, `1..3` is the range 1, 2, 3.

A callable literal is the name of an operation or function defined in the compilation scope. For instance, `X` is an operation literal that refers to the standard

library X operation, and Message is a function literal that refers to the standard library Message function.

If an operation supports the Adjoint functor, then (Adjoint op) is an operation expression. Similarly, if the operation supports the Controlled functor, then (Controlled op) is an operation expression.

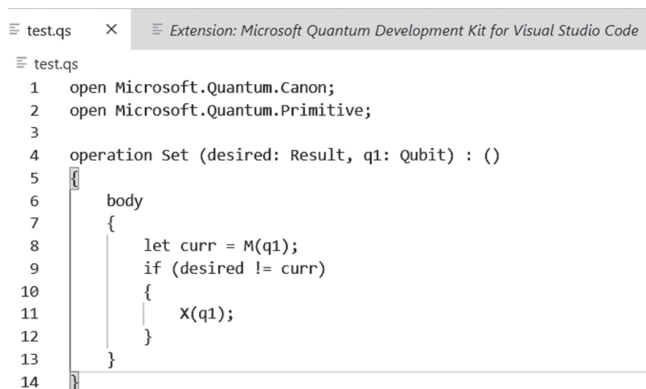
Q# callables are allowed to be directly or indirectly recursive. That is, an operation or function may call itself, or it may call another callable that directly or indirectly calls the callable operation.

### 3. A quantum program creating example (Q#)

In order to illustrate the possibilities of quantum computing via Q# language we will create a simple program in Visual Studio. As a precondition, Microsoft Quantum Development Kit should be installed on the computer.

Firstly, we should open Visual Studio 2017 and create a solution of the type “Q# Application”. Afterwards, optionally we should update the necessary NuGet packages. Now, it is possible to create a program.

Consider a simple program of setting a qubit state (see Fig. 1):



```

test.q#
1  open Microsoft.Quantum.Canon;
2  open Microsoft.Quantum.Primitive;
3
4  operation Set (desired: Result, q1: Qubit) : ()
5  {
6      body
7      {
8          let curr = M(q1);
9          if (desired != curr)
10         {
11             X(q1);
12         }
13     }
14 }

```

Fig. 1. The qubit state setting function

As it can be seen from the code, the standard tools allow us to do everything via the standard library. Thus, we include the necessary standard tools on the first and the second lines and create a Set function accepting two parameters of Result and Qubit types respectively. Then, we get the current qubit state via M() function and compare the current state with the desired one. If they are not equal, we set the necessary state.

It should be clear from this example that having such a dedicated tool for a special computing paradigm as a language simplifies programming dramatically, since using other tools (e.g. calling an API) would require much more code.

### Conclusion

In scope of the article the basic tools of quantum computing have been investigated. It included different languages (imperative (QCL), functional (Quipper), multi-paradigm (Q#)), quantum pseudocode and frameworks used with the existing languages (e.g. ProjectQ for Python).

According to the retrieved data, the accurate performance cannot be determined using the given tools because of the absence of a fully functional quantum computer prototype (the results obtained for emulators cannot be considered precise). The tools cover most of the basic quantum computing units including qubits and operations with them, but differ in terms of the used syntax and the standard library.

### References

- [1] Nielsen, M. A., Chuang, I. L. Quantum Computation and Quantum Information. Quantum Computation and Quantum Information. UK: Cambridge University Press. 2010.
- [2] Simon J Gay. Quantum programming languages: Survey and bibliography. Mathematical Structures in Computer Science. 2006. № 16 (04). P. 581-600.
- [3] C. C. Moran. Quintuple: A tool for introducing quantum computing into the classroom. / Christine C. Moran – Frontiers in Physics 6(JUL). – p. 69
- [4] Bernhard Ömer. Quantum Programming in QCL. Master thesis computing science. TU Vienna. 2000.
- [5] Emmanuel Knill. Conventions for quantum pseudocode. Technical Report. Los Alamos National Lab., NM (United States). 1996.
- [6] Q#: Enabling scalable quantum computing and development with a high-level domain-specific language. arXiv:1803.00652 [quant-ph], 2018.
- [7] Quipper: A Scalable Quantum Programming Language. Alexander S. Green and others. ACM SIGPLAN Notices 48(6). P. 333-342. 2013

*The article was delivered to editorial staff on the 20.10.2020*

С.Ф. Чалий<sup>1</sup>, В.О. Лещинський<sup>2</sup>

<sup>1</sup>професор кафедри інформаційних управляючих систем,  
Харківський національний університет радіоелектроніки, Україна,  
serhii.chalyi@nure.ua, ORCID:0000-0002-9982-9091

<sup>2</sup>доцент кафедри програмної інженерії,  
Харківський національний університет радіоелектроніки, Україна,  
volodymyr.leshchynskiy@nure.ua, ORCID:0000-0002-8690-5702

## ТЕМПОРАЛЬНІ ПАТЕРНИ ВПОДОБАНЬ КОРИСТУВАЧІВ В ЗАДАЧАХ ФОРМУВАННЯ ПОЯСНЕНЬ В РЕКОМЕНДАЦІЙНІЙ СИСТЕМІ

Розглянуто проблему врахування змін вимог користувача рекомендаційної системи при побудові пояснень щодо рекомендацій. Ця проблема може виникнути в результаті циклічних змін вимог користувача. Її рішення пов'язане з побудовою пояснення, що порівнює альтернативні варіанти вибору такого користувача. Розроблені моделі темпоральних патернів складаються з множини темпоральних відношень між подіями вибору користувачами товарів та послуг. Перший патерн містить альтернативи у вигляді послідовного вибору в часі декількох предметів або вибору тільки пари - першого і останнього предмета. Другий патерн, послідовно-альтернативного вибору, складається з послідовності виборів у часі, яка завершується першим патерном. Запропонований підхід до формування патернів заснований на побудові наборів даних, що містять в собі темпоральні залежності між групою виборів користувача для заданого рівня деталізації часу. Набір даних з темпоральними залежностями використовується для побудови темпорального графу процесу вибору користувача рекомендаційної системи. Останній включає в себе набір темпоральних патернів із зазначенням часу їх початку і завершення, що дає можливість визначити тривалість реалізації цих шаблонів. На основі патернів формуються підмножини темпоральних відносин для побудови пояснень щодо рекомендованого переліку товарів і послуг. Експериментальна перевірка розробленого підходу з використанням набору даних про продажі «Online Retail» показала можливість виділення темпоральних патернів навіть на коротких вихідних вибірках.

РЕКОМЕНДАЦІЙНА СИСТЕМА, ПОЯСНЕННЯ РЕКОМЕНДАЦІЙ, ТЕМПОРАЛЬНІ ПАТЕРНИ, ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ПРОЦЕСІВ

**Чалий С.Ф., Лещинський В.О. Темпоральные паттерны предпочтений пользователей в задачах формирования объяснений в рекомендательной системе.** Рассмотрена проблема учета изменений поведения пользователя рекомендательной системы при построении объяснений относительно рекомендаций. Данная проблема возникает в результате циклических изменений требований пользователя. Ее решение связано с построением объяснения, сравнивающего альтернативные варианты выбора пользователя рекомендательной системы. Разработанные модели темпоральных паттернов состоят из множества темпоральных отношений между событиями выбора пользователями товаров и услуг. Первый паттерн содержит альтернативы в виде последовательного выбора во времени нескольких предметов или выбора только пары - первого и последнего предмета. Второй паттерн, последовательно-альтернативного выбора, состоит из последовательности выборов во времени, которая завершается первым паттерном. Предложенный подход к формированию паттернов основан на построении наборов данных, содержащих в себе темпоральные зависимости между группой выборов пользователя для заданного уровня детализации времени. Набор данных с темпоральными зависимостями используется для построения темпорального графа процесса выбора пользователя рекомендательной системы. Последний включает в себя набор темпоральных паттернов с указанием времени их начала и завершения, что дает возможность определить продолжительность реализации этих шаблонов. На основе паттернов формируются подмножества темпоральных отношений для построения объяснений относительно рекомендованного перечня товаров и услуг. Экспериментальная проверка разработанного подхода с использованием набора данных о продажах «Online Retail» показала возможность выделения темпоральных паттернов даже на коротких исходных выборках.

РЕКОМЕНДАТЕЛЬНАЯ СИСТЕМА, ОБЪЯСНЕНИЕ РЕКОМЕНДАЦИЙ, ТЕМПОРАЛЬНЫЕ ПАТТЕРНЫ, ИНТЕЛЕКТУАЛЬНИЙ АНАЛИЗ ПРОЦЕССОВ

**Chalyi S., Leshchynskiy V. Temporal patterns of user preferences in the tasks of forming explanations in the recommender system.** The problem of taking into account changes in the user's behavior of the recommendation system when constructing explanations for recommendations is considered. This problem occurs as a result of cyclical changes in user requirements. Its solution is associated with the construction of an explanation comparing the alternative choices of the user of the recommendation system. The developed models of temporal patterns consist of a set of temporal relationships between the events of users' choice of goods and services. The first pattern contains an alternative in the form of sequential selection in time of several objects or the selection of only a pair - the first and the last object. The second pattern, sequential-alternative choice, consists of a sequence of choices over time, which ends with the first pattern. The proposed approach to the formation of patterns is based on the construction of data sets containing temporal dependencies between a group of user choices for a given level of time detail. The temporal dataset is used to construct a temporal graph of the recommender system user selection process. The latter includes a set of temporal patterns with an indication of the time of their beginning and end, which makes it possible to determine the duration of the implementation of these patterns. On the basis of the patterns, subsets of temporal relationships are formed to build explanations for the recommended list of goods and services. Experimental verification of the developed approach using the "Online Retail" sales data set has shown the possibility of identifying temporal patterns even on short initial samples.

RECOMMENDER SYSTEM, EXPLANATION OF RECOMMENDATIONS, TEMPORAL PATTERNS, PROCESS MINING

### Вступ

Рекомендаційні системи відбирають для користувача товари або послуги у відповідності до його персональних вподобань [1]. Застосування рекомендацій значно спрощує для споживача вирішення проблеми вибору з набору однотипних товарів з схожими характеристиками. Тому рекомендації широко використовуються при онлайн-продажах товарів, перегляді фільмів за допомогою стрімінгових сервісів, пропозиціях туристичних турів, виборі наукових конференцій, огляді новин [1, 2].

Для побудови рекомендацій такі системи використовують інформацію про продажі та рейтинги товарів, а також про зміну останніх з часом [1,3]. Однак рейтинги можуть бути спотворені в результаті шилінг-атак [4, 5]. Рекомендований перелік товарів або послуг після атаки відображає вподобання не користувача, а атакуючого. Такі рекомендації обмежують раціональний вибір користувача та знижують довіру до результатів роботи рекомендаційної системи.

Для підтримки довіри користувача рекомендації доповнюють поясненнями [6], яке має давати відповідь на запитання «Чому отримана така рекомендація?» [7-9]. Оцінка впливу пояснень на вибір користувачів показала, що користувачі рідше відмовляються від рекомендованого предмета після отримання пояснення [6, 10]. Уподобання користувача змінюються з часом, тому динаміку їх інтересів доцільно враховувати при побудові рекомендацій та пояснень до цих рекомендацій [11-15]. Пояснення, що враховує циклічні зміни вподобань споживачів, базується на використанні темпоральних знань. Останні визначають залежності між подіями вибору користувача [12, 16]. Формування цих знань потребує визначення темпоральних патернів зміни вподобань користувача. Такі патерни формуються для часового інтервалу зміни потреб споживачів [13, 14] з використанням даних про послідовність їх вибору [17]. Таким чином, проблема побудови пояснень з урахуванням змін вимог користувачів на основі темпоральних патернів є актуальною.

Для вирішення цієї проблеми необхідно визначити набір базових темпоральних відношень і з них сформулювати патерни.

В роботі [18] для формалізації темпоральних відношень запропоновано використовувати ймовірнісні темпоральні правила. Кожне правило задає впорядкованість у часі для пар подій, дій або станів. Ймовірність застосування правила залежить від його ваги. Вага правила розраховується з урахуванням частоти його реалізації для відомих процесів, які представляються у вигляді послідовності подій [19].

Модель пояснення, що враховує темпоральну динаміку вимог користувачів, запропонована в роботі [20]. Дана модель в якості пояснення використовує

числову оцінку зміни інтересів користувачів на заданому інтервалі часу. Основна ідея такого пояснення полягає в тому, що користувач з більшою ймовірністю вибере предмет, попит на який збільшується на момент прийняття рішення. В роботі [21] виконано деталізацію пояснення для декількох інтервалів в рамках заданого періоду часу. При деталізації виділяються інтервали, на яких темпоральні правила є узгодженими. Підхід до узгодження темпоральних правил відрізняється від класичного визначення, запропонованого в [22]. Темпоральні правила є узгодженими в тому випадку, якщо вони що відображають одночасне збільшення (або зменшення) інтересу до предмета з боку декількох користувачів. В цілому існуючі підходи до побудови пояснень з урахуванням темпоральної динаміки вподобань користувачів враховують узгоджені зміни попиту на товари і послуги. Темпоральне пояснення показує зростання попиту на рекомендований предмет, тим самим надаючи користувачу обґрунтування для вибору предмета в поточний момент часу. Такі пояснення інтерпретують процес формування рекомендації у вигляді «Предмет відібраний відповідно до інтересів користувача, популярний, його вибирають все частіше останнім часом. Тому рекомендаційна система запропонувала даних предмет користувачеві». Однак дане пояснення не дозволяє порівняти альтернативні варіанти вибору, щоб користувач отримав відповідь на питання «Чому цей предмет є кращим?. Чи є можливість вибрати альтернативний предмет?». Для того, щоб користувач переконався в раціональності отриманої рекомендації, в поясненні необхідно порівняти темпоральну динаміку [2, 12] для декількох альтернатив. На основі такого порівняння споживач отримує мотивацію для вибору найбільш популярної в даний момент альтернативи, що відповідає його інтересам.

Таким чином, існуючі підходи до побудови пояснень не дозволяють виділити альтернативні варіанти поведінки користувача. Для вирішення задачі побудови пояснень з урахуванням темпоральної динаміки альтернативного вибору необхідно виділити підмножини темпоральних залежностей, що характеризують ці альтернативи. Такі залежності входять до складу темпоральних патернів, що відображають циклічність вибору користувачів у часі. Тому вирішення задачі формалізації та побудови темпоральних патернів альтернативного вибору дає можливість виділити темпоральні відношення для таких альтернатив та використати отримані відношення для побудови пояснень.

### 2. Постановка задачі

Метою статті є розробка моделей темпоральних патернів зміни вподобань користувачів рекомендаційної системи з часом для формування пояснень щодо рекомендованого переліку товарів та послуг.

Для досягнення мети дослідження вирішуються такі задачі: розробка моделей темпоральних патернів альтернативного та послідовно-альтернативного користувача рекомендаційної системи; розробка підходу до автоматизованої побудови патернів вибору користувача; експериментальна перевірка підходу до побудови темпоральних патернів вибору користувача рекомендаційної системи.

### 3. Моделі темпоральних патернів вподобань користувачів

Темпоральний патерн складається з темпоральних залежностей, що визначають послідовність вибору множиною користувачів товарів або послуг із визначеної групи. Наприклад, групи м'яких дитячих іграшок, групи подарункових навчальних наборів, тощо. Даний патерн задає темпоральний порядок для елементів  $e_i$  множини подій  $E$  вибору споживача. Кожна подія відповідає дії користувача із вибору товару або послуги. Темпоральний порядок для кожної пари подій  $(e_i, e_j) \in E$  задається на послідовності дискретних моментів часу  $T = \langle t_1, t_2, \dots, t_i, \dots, t_j, \dots \rangle$ . Послідовність  $T$  є лінійно упорядкованою і тому має властивість антирефлексивності. Відповідно, множина упорядкованих у часі пар подій не може містити пари  $(e_i, e_i): e_i \in E$ .

Для кожної пари  $(e_i, e_j)$  упорядкованих у часі подій можуть виконуватись два альтернативних обмеження:

- пара  $(e_i, e_j)$  визначає події вибору споживачів, між якими не було зроблено іншого вибору, тобто ці події є послідовними у часі;
- між подіями пари  $(e_i, e_j)$  можуть існувати інші події вибору товарів або послуг користувачем.

Перший випадок відповідає темпоральному відношенню X-типу [18]. Дане відношення для подій вибору користувача має такий вигляд:

$$X_i : e_i \mapsto e_{i+1}, \quad (1)$$

де  $e_i$  та  $e_{i+1}$  – події вибору, що виникли на двох послідовних моментах часу  $t_i$  та  $t_{i+1} > t_i$ .

У другому випадку темпоральна упорядкованість задається  $F$  – відношенням [18], що має такий вигляд для подій вибору користувачів  $e_i$  та  $e_{i+n}$ :

$$F_{i,n} : e_i \mapsto e_{i+n}. \quad (2)$$

Темпоральне відношення (2) складається з послідовності відношень (1), тобто:

$$F_{i,n} = \langle X_i, X_{i+1}, \dots, X_{i+n-1} \rangle. \quad (3)$$

Згідно (3), темпоральне відношення  $F_{i,n}$  та послідовність відношень  $X_i, X_{i+1}, \dots, X_{i+n-1}$  задають альтернативні варіанти поведінки користувача для визначеного періоду часу. У випадку послідовності  $X_i, X_{i+1}, \dots, X_{i+n-1}$  користувач вибирає декілька предметів із заданої групи, а у випадку відношення  $F_{i,n}$  – вибирає лише перший (подія  $e_i$ ) та останній (подія  $e_{i+n}$ ).

Тому набір темпоральних відношень видів (1) та (2) задають патерн темпоральної динаміки вподобань користувачів.

В залежності від поведінки користувачів можуть бути виділені паралельний (альтернативний вибір) та послідовно-альтернативний патерни.

Перший патерн  $P$  містить щонайменше два альтернативних варіанти вибору споживача, представлених  $X$  та  $F$  – відношеннями.

$$P = \{F_{i,n}, \langle X_i, X_{i+1}, \dots, X_{i+n-1} \rangle\}. \quad (4)$$

Схематичне зображення патерна представлено на рис. 1.

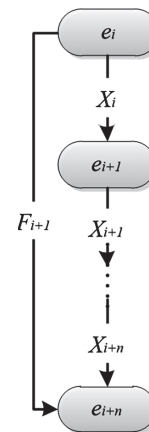


Рис. 1. Темпоральний патерн альтернативного вибору

Другий патерн  $L$  поєднує представлену X-залежностями послідовність вибору товарів та послуг та паралельний патерн  $P$ :

$$L = \langle X_{i-k}, X_{i-k-1}, \dots, X_{i-1}, P \rangle. \quad (5)$$

Схематичне зображення патерну  $L$  представлено на рис. 2.

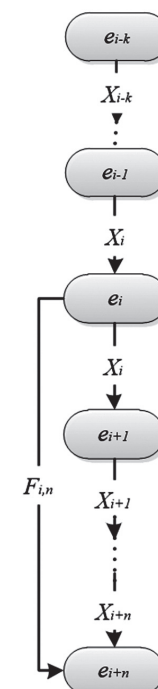


Рис. 2. Темпоральний патерн послідовно-альтернативного вибору

Представлені патерни задають базові альтернативи темпоральних змін у вподобаннях користувача, тобто визначають цикли його вибору. Поєднання декількох патернів дає можливість визначити більш складні варіанти поведінки користувача.

Особливість даних патернів полягає в тому, що вони визначаються для групи схожих з точки зору користувача предметів. Схожими ці предмети є в сенсі забезпечення потреб споживача на визначеному інтервалі часу.

Побудова таких патернів дає можливість створити пояснення згідно із представленим у роботі [20] методом. Використання моделей (4) та (5) дає можливість визначити множину темпоральних відношень, які забезпечують формування пояснення.

Пропонується підхід до побудови темпоральних патернів з використанням технології інтелектуального аналізу процесів. Остання орієнтована на формування графових моделей послідовності дій дискретних процесів (наприклад, бізнес-процесів) на основі аналізу логів (журналів подій). Лог складається з послідовності подій. Кожна подія містить інформацію про дії, що були виконані у відповідній інформаційній системі. Побудова темпоральних патернів передбачає виконання таких етапів.

*Етап 1.* Підготовка вибірки даних на заданому інтервалі часу. На даному етапі виконується відбір цільової підмножини товарів із апріорно заданої групи, що дає можливість відобразити у патерні альтернативний вибір споживачів. Відбір цільової підмножини може виконуватись, наприклад, за критерієм популярності (кількості продажів товарів), або ж для товарів із переліку, рекомендованого певному споживачеві.

*Етап 2.* Формування нової, упорядкованої у часі, множини подій на основі сумування кількості покупок (вибору) предметів для заданого рівня грануляції часу.

На даному етапі визначається кількість продажів певного предмету за день, тиждень, місяць, тощо, та формується подія продажу розрахованої кількості товару.

*Етап 3.* Підготовка вхідних даних для інтелектуального аналізу. На даному етапі формується нова послідовність подій, що відображає  $X$  та  $F$  – темпоральні відношення між вибором споживача для заданого рівня грануляції часу.

*Етап 4.* Формування множини темпоральних патернів вибору користувачів рекомендаційної системи з використанням одного із методів інтелектуального аналізу процесів. Дана послідовність етапів дає можливість сформувати та візуалізувати темпоральні патерни, визначити їх тривалість у часі та використати їх для аналізу поведінки користувача рекомендаційної системи.

Для побудови пояснень додатково виконується етап побудови множини темпоральних відношень.

*Етап 5.* Побудова темпоральних відношень  $X$  та  $F$  – типів для патернів. Дані відношення формуються з урахуванням визначеної на етапі 4 тривалості темпоральних патернів. Для кожного з відношень обчислюється вага як нормована різниця продажів для подій у складі відношення.

В подальшому зважені  $X$  та  $F$  – відношення у складі темпорального патерну можуть бути використані для побудови пояснень методами [20, 21].

### 3. Побудова темпоральних патернів засобами інтелектуального аналізу процесів

Експериментальна перевірка запропонованого підходу до побудови патернів зміни вибору користувачів у часі виконана з використанням набору даних «Online Retail», що містить інформацію про продажі подарункових товарів у мережі супермаркетів у Великобританії. Такі товари продаються у значних кількостях з певною періодичністю, з урахуванням святкових та робочих днів, що дає можливість сформувати темпоральні патерни циклічної зміни вподобань споживачів на вибірці невеликого розміру.

Набір даних «Online Retail» містить таку ключову інформацію, важливу для побудови темпоральних патернів: StockCode – код товару; InvoiceDateTime – дата та час покупки; Quantity – кількість проданого товару; CustomerID – код покупця.

Інформація про дату та час покупки, а також кількість проданих товарів дає можливість сформувати дані про продажі для визначеного рівня грануляції часу.

Результати експерименту представлені на рис. 3 та рис. 4.

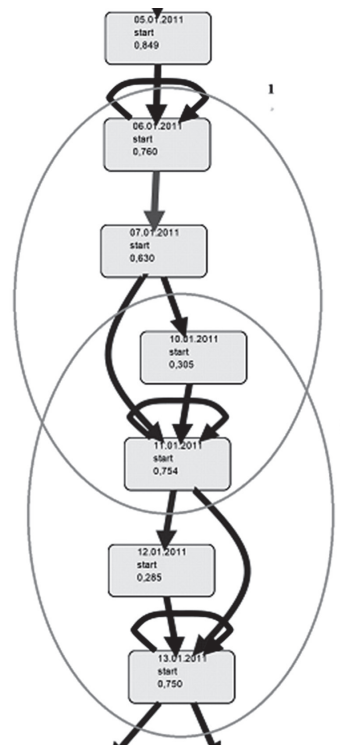


Рис. 3. Патерни послідовно-альтернативного вибору користувачів

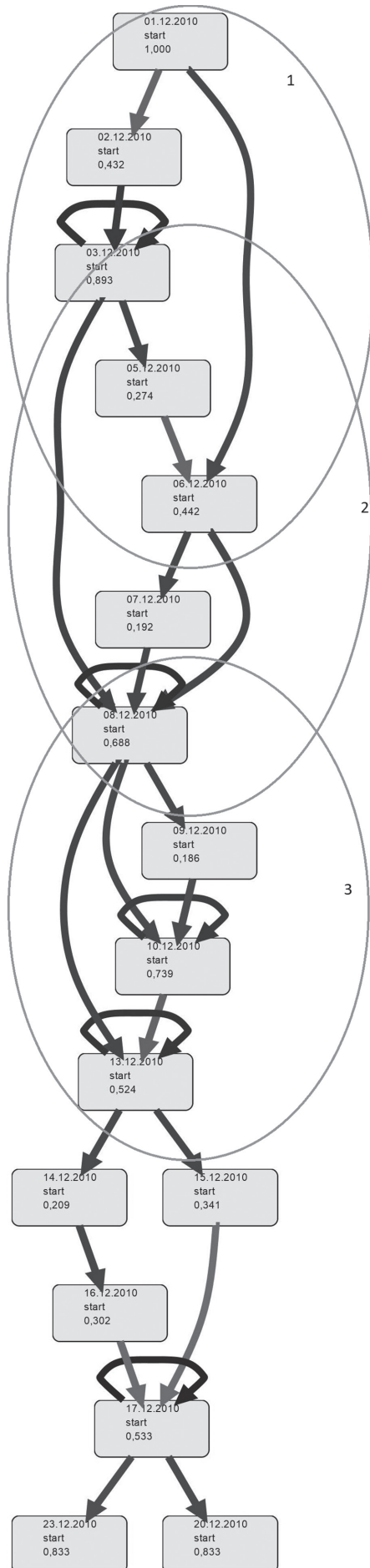


Рис. 4. Патерни альтернативного вибору користувачів

Реалізація підходу до побудови темпоральних патернів вподобань користувача в рамках експериментальної перевірки має такі особливості.

При відборі вхідних підмножин даних використовувався місячний період часу, оскільки повний набір даних містить інформацію за рік.

Для порівняння альтернативного вибору користувачів була відібрана група з трьох найбільш популярних подарункових товарів. Популярність оцінювалась за максимальною кількістю покупок за один день.

Тривалість патерну визначалась інтервалом часу між двома подіями темпоральної залежності типу  $F$ .

За результатами експерименту отримано темпоральні патерни альтернативного та послідовно-альтернативного вибору тривалістю 4-5 днів.

Отримані патерни відповідають моделям (4) та (5). Патерн альтернативного вибору містить два базових варіанти поведінки користувача. Перший варіант полягає у послідовному виборі 4-5 предметів протягом визначеного періоду часу. Цей варіант відображається послідовністю  $X$  – залежностей. Альтернативний варіант полягає у виборі предметів лише у першого та останнього предметів. Дана альтернатива представляється  $F$  – залежністю.

Патерни альтернативно-послідовного вибору також містить базову послідовність із  $X$  – залежностей. Однак в альтернативному варіанті покупець пропускає вибір одного із предметів, що відображається  $F$  – залежністю. Ключові характеристики отриманих патернів наведено у табл. 1 та табл. 2.

Таблиця 1

Характеристики темпорального патерну послідовно-альтернативного вибору

Тип залежності	Кількість залежностей у складі патерну	Тривалість, днів
Next	4-5.	1-3
Future	1-2	3-5
Патерн в цілому	4-5	3-5

Таблиця 2

Характеристики темпорального патерну альтернативного вибору

Тип залежності	Кількість залежностей у складі патерну	Тривалість, днів
Next	4	1-2
Future	1-2	2-5
Патерн в цілому	5-6	5

На рис. 3 представлені патерни з послідовно-альтернативного вибору, що відображають еволюційні зміни вподобань користувачів. В даному випадку задано базову послідовність вибору користувача, що представлена  $X$  – залежностями. Наприклад,  $(06.01.2011)X(07.01.2011)X(10.01.2011)X(11.01.2011)$ .

Споживач може пропустити один із товарів. Такий пропуск представлений, наприклад, залежністю (07.01.2011)F(11.01.2011).

На рис. 4 представлено темпоральні патерни, що відображають 5-денні цикли альтернативних вподобань користувача. Тривалість циклу задається правилами (01.12.2010)F(01.12.2010), (03.12.2010)F(08.12.2010), (08.12.2010)F(13.12.2010). Такі патерни показують, що користувач може вибрати від двох до 5 товарів із заданої групи на протязі 5 днів. Альтернативні шляхи по графу, наприклад (01.12.2010)F(01.12.2010) або (01.12.2010)X(02.12.2010)X(03.12.2010)X(05.12.2010)X(06.12.2010) вибираються різними користувачами.

Тому при виборі 01.12.2010 рекомендаційна система має надати такі рекомендації та пояснення, які б мотивували споживача реалізувати другу послідовність вибору. Наприклад, пояснення може подавати такі властивості вибраного 01.12.2010 предмету, які потребують доповнення предметами, що вибирались 02.12.2010, 03.12.2010 та в наступні дні. Фактично таке пояснення є подальшим розвитком традиційної рекомендації «разом з цим товаром покупці придбали такі товари ...».

Відмінність запропонованого підходу полягає в прогнозуванні вибору товару в майбутньому, з використанням патерну змін вподобань користувачів з часом.

### Висновки

Розглянуто проблему врахування змін поведінки користувача рекомендаційної системи при побудові пояснень до рекомендацій. Дана проблема виникає внаслідок як циклічних (наприклад, сезонних) змін вимог користувача, так і внаслідок еволюції його потреб.

Запропоновано моделі темпоральних патернів альтернативного вибору та послідовно-альтернативного вибору споживача. Патерни складаються із множин темпоральних відношень між подіями вибору користувачами товарів та послуг. Використовуються темпоральні відношення для пари послідовних подій, та парю подій вибору, між якими виникали інші події. Перший патерн містить альтернативу у вигляді послідовного вибору у часі декількох предметів або ж вибору лише пари – першого й останнього предмету. Другий патерн, складається з послідовності виборів у часі, яка завершується першим патерном.

Виявлення патернів у послідовності даних щодо зробленого користувачем вибору дає можливість сформувати набори темпоральних залежностей, які можуть бути враховані при формуванні рекомендацій, а також використані при побудові пояснень, що враховують темпоральну динаміку інтересів користувача.

Запропоновано підхід до виявлення темпоральних патернів з використанням технології інтелектуального аналізу процесів.

Даний підхід передбачає формування наборів даних, що містять у собі темпоральні залежності між групою виборів користувача для заданого рівня грануляції часу. Набір даних із темпоральними залежностями використовується для побудови темпорального графу процесу вибору користувача рекомендаційної системи, що містить у собі набір темпоральних патернів та дає можливість визначити тривалість цих шаблонів. З урахуванням тривалості патернів формуються підмножини темпоральних відношень для побудови пояснень щодо рекомендованого переліку товарів та послуг.

Експериментальна перевірка розробленого підходу з використанням набору «Online Retail» показала можливість виділення темпоральних патернів навіть на коротких вибірках для підмножини найбільш популярних предметів.

### Список літератури:

- [1] Aggarwal C. Recommender systems: The Textbook. – New York: Springer. – 2017. – 498 p.
- [2] Raza S., Ding C. News Recommender System Considering Temporal Dynamics and News Taxonomy // IEEE International Conference on Big Data. – 2019. – P. 920-929. doi: 10.1109/BigData47090.2019.9005459
- [3] Fararni K., Aghoutane B., Riffi J., Abdelouahed S. Comparative study on approaches of recommendation systems // Embedded Systems and Artificial Intelligence. – 2020. – P. 753-764.
- [4] Chala O., Novikov, L., Chernyshova, L. Method for detecting shilling attacks in e-commerce systems using weighted temporal rules // EUREKA: Physics and Engineering. – 2019. – Vol. 5. – P. 29-36. doi: 10.21303/2461-4262.2019.00983
- [5] Chala O., Novikov, L., Chernyshova, L. Method for detecting shilling attacks in recommendater systems using objective feedback // EUREKA: Physics and Engineering. - 2020. - Vol. 5. - P. 21-30. doi: 10.21303/2461-4262.2020.001394.
- [6] Tintarev N., Masthoff J. Evaluating the effectiveness of explanations for recommender systems // User Model User-Adap Inter. – 2012. – Vol. 22. – P. 399– 439.
- [7] Zachary C. L. The mythos of model interpretability // Communications of the ACM. – 2016. – P. 1-6.
- [8] Miller T. Explanation in artificial intelligence: Insights from the social sciences // Artificial Intelligence. – 2019. – Vol. 267. – P. 1-38. doi: 10.1109/BigData47090.2019.9005459
- [9] Tsai C., Brusilovsky P. Explaining recommendations in an interactive hybrid social recommender // Proceedings of the 24th International Conference on Intelligent User Interfaces (IUI '19). – 2019. – P. 391-396.
- [10] Dominguez V., Messina P., Donoso-Guzmán I., Parra D. The effect of explanations and algorithmic accuracy on visual recommender systems of artistic images // Proceedings of the 24th International Conference on Intelligent User Interfaces (IUI '19). – 2019. – P. 408-416.
- [11] Nan D., Yichen W., Niao, H., Le S. Time-Sensitive Recommendation From Recurrent User Activities // Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS'15). – 2015. – Vol. 2. – P. 3492–3500.

- [12] Campos P. G., Diez F., Cantador I. Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols // *User Modeling and User-Adapted Interaction*. – 2014. – Vol. 24(2). – P. 67-119. doi: 10.1007/s11257-012-9136-x
- [13] Ma Y., Narayanaswamy B., Lin H., Ding H. Temporal-contextual recommendation in real-time // *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '20)*. – 2020. – P. 2291–2299. doi: 10.1145/3394486.3403278
- [14] Rabiou I., Naomie S., Aminu D., Akram O. Recommender system based on temporal models: A Systematic Review // *Applied Sciences*. – 2020. – Vol. 10(7). – P. 2204. doi: 10.3390/app10072204
- [15] Chalyi S., Leshchynskiy V., Leshchynska, I. Method of forming recommendations using temporal constraints in a situation of cyclic cold start of the recommender system // *EUREKA: Physics and Engineering*. – 2019. – Vol. 4. – P. 34–40.
- [16] Chalyi S., Pribylnova I. The method of constructing recommendations online on the temporal dynamics of user interests using multilayer graph // *EUREKA: Physics and Engineering*. – 2019. – Vol. 3. – P. 13-19. doi: 10.21303/2461-4262.2019.00894
- [17] Wang C., Min Z., Ma W., Liu Y. Modeling item-specific temporal dynamics of repeat consumption for recommender systems // *WWW '19- The World Wide Web Conference*. – 2019. – P. 1977–1987.
- [18] Levykin V., Chala O. Development of a method for the probabilistic inference of sequences of a business process activities to support the business process management // *Eastern-European Journal of Enterprise Technologies*. – 2018. – Vol. 3 (95). – P. 16–24.
- [19] Levykin, V., Chala, O. Method of determining weights of temporal rules in markov logic network for building knowledge base in information control systems // *EUREKA: Physics and Engineering*. – 2018. – Vol. 5. – P. 3-10.
- [20] Chalyi S., Leshchynskiy V. Method of constructing explanations for recommender systems based on the temporal dynamics of user preferences // *EUREKA: Physics and Engineering*. – 2020. – Vol. 3. – P. 43-50.
- [21] Chalyi S., Leshchynskiy V., Leshchynska I. Detailing explanations in the recommender system based on matching temporal knowledge // *Eastern-European Journal of Enterprise Technologies*. - 2020. - Vol 4. - No 2 (106). - P. 6-13.
- [22] Thagard P. *Computational Models in Science and Philosophy. Introduction to Formal Philosophy*. – 2018. – P. 457-467.

*Надійшла до редколегії 07.10.2020*

UDK 004.8

DOI 10.30837/bi.2020.2(95).04



**І.В. Кириченко<sup>1</sup>, А. В. Назаренко<sup>2</sup>, Р. О. Попов<sup>3</sup>**

<sup>1</sup> к.т.н., ст.викладач кафедри програмної інженерії,  
Харківський національний університет радіоелектроніки, Україна,  
iryna.kyrychenko@nure.ua, ORCID iD: 0000-0002-7686-6439

<sup>2</sup> студент кафедри програмної інженерії,  
Харківський національний університет радіоелектроніки, Україна,  
anton.nazarenko@nure.ua

<sup>3</sup> студент кафедри програмної інженерії,  
Харківський національний університет радіоелектроніки, Україна,  
rostyslav.popov@nure.ua

## ОПТИМІЗАЦІЯ ТА МАСШТАБУВАННЯ NODE.JS ДОДАТКІВ

**Кириченко І.В., Назаренко А.В., Попов Р.О. Оптимізація та масштабування Node.js додатків.** Актуальність цієї роботи зумовлена тим, що Node.js швидко стає однією з найпопулярніших платформ для створення швидких, масштабованих веб та мобільних додатків. Опитування користувачів Node.js 2017 року показує, що в даний час в Інтернеті перебуває понад 7 мільйонів екземплярів Node.js, причому кожен четвертий користувач планує збільшити використання Node.js протягом наступних 12 місяців. І легко зрозуміти, чому 68 відсотків цих користувачів кажуть, що Node.js покращує продуктивність розробників, 58 повідомляє, що зменшує витрати на розробку, а 50 відсотків кажуть, що підвищує продуктивність додатків. Оскільки Node.js все частіше стає технологією, що обирається для розробки додатків, попит на досвідчених розробників Node.js також буде продовжувати зростати. У роботі розглянуто проблему масштабування та оптимізації Node.js додатків. Запропоновано декілька підходів розробки Node.js додатків, які допоможуть оптимізувати швидкість виконання програмного коду та розробити рішення, що буде легко масштабуватися.

ОПТИМІЗАЦІЯ, NODE.JS, МАСШТАБУВАННЯ, V8 GARBAGE COLLECTION, ПРОДУКТИВНІСТЬ

**Кириченко И.В., Назаренко А.В., Попов Р.А. Оптимизация и масштабирование Node.js приложений.** Актуальность данной работы обусловлена тем, что Node.js быстро становится одной из самых популярных платформ для создания быстрых, масштабируемых веб-сайтов и мобильных приложений. Опрос пользователей Node.js 2017 показывает, что в настоящее время в интернете пересматривается более 7 миллионов экземпляров Node.js, а также каждый четвертый пользователь планирует увеличить использование Node.js за 12 месяцев. Легко понять, почему 68 процентов этих пользователей говорят, что Node.js улучшает производительность разработчиков, 58 сообщает, что уменьшает затраты на разработку, и 50 процентов говорят, что увеличивает производительность приложений. Несмотря на то, что Node.js все чаще становится распространенной технологией для разработки приложений, спрос на опытных разработчиков Node.js также будет продолжено расти. В работе рассмотрена проблема масштабирования и оптимизации Node.js приложений. Предложено несколько подходов разработки Node.js приложений, которые помогут оптимизировать скорость выполнения программного кода и разработать решение, что будет легко масштабироваться.

ОПТИМИЗАЦИЯ, NODE.JS, МАСШТАБИРОВАНИЕ, V8 GARBAGE COLLECTION, ПРОИЗВОДИТЕЛЬНОСТЬ

**Kyrychenko I., Nazarenko A., Popov R. Optimization and scaling Node.js apps.** The relevance of this work is due to the fact that Node.js is rapidly becoming one of the most popular platforms for building fast, scalable web and mobile applications. In fact, the 2017 Node.js User Survey reveals that there are currently over 7 million Node.js instances online, with three in four users planning to increase their use of Node.js in the next 12 months. And it's easy to see why: 68 percent of those users say Node.js improves developer productivity, 58 report it reduces development costs, and 50 percent say it increases application performance. As Node.js increasingly becomes the preferred technology for application development, the demand for expert Node.js developers will also continue to increase. The paper considers the problem of scaling and optimization of Node.js applications. Several approaches to developing Node.js applications have been proposed to help optimize code execution speed and develop solutions that are easy to scale.

OPTIMIZATION, NODE.JS, SCALING, V8 GARBAGE COLLECTION, PERFORMANCE

### Вступ

Багато написано про те, що можна зробити за допомогою Node.js, як розробники можуть почати використовувати цю платформу, і чому вона стала основною технологією на стороні сервера з деякими найбільшими корпораціями світу, але мало що написано для допомоги новачкам або середнім розробникам Node.js підняти свої навички на новий рівень.

Типовий респондент використовує Node трохи більше 2 років і проводить більше половини свого часу на розробці Node. Досвід роботи з Node.js можемо побачити на рис. 1.

Враховувати продуктивність серверу, доступність та швидкість виконання програмного коду на ньому в момент проектування архітектури — дуже важливо. Чим раніше буде розуміння того, яке навантаження

буде на сервер — тим простіше буде у подальшій розробці та підтримці проекту. Покращувати роботу Node.js серверу можна багатьма шляхами, далі будуть приведені основні методи того, як збільшити кількість запитів, що може обробити сервер за певну одиницю часу.

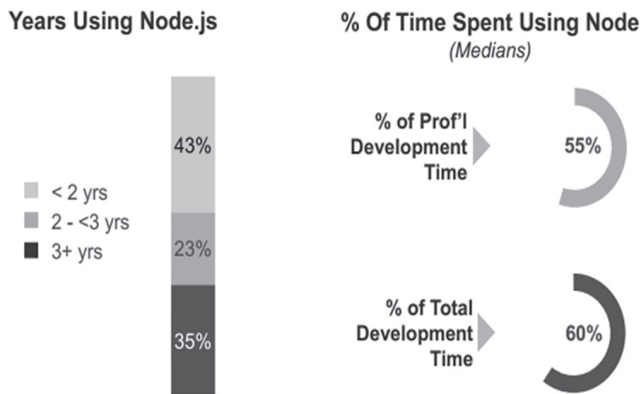


Рисунок 1. Досвід роботи з Node.js

Джерело: <https://nodejs.org/en/user-survey-report/>

У цій статті будуть наведені стратегії, які можуть призвести до несподіваних проблем у довгостроковій перспективі, але, якщо їх правильно впровадити, це значною мірою підвищить ефективність додатків.

## 1. Підготовка до запуску виробництва

Підготовка проекту завжди є критичним моментом у будь-якому шляху розробки додатків, і це, безумовно, стосується проектів Node.js. Це остання можливість команди знайти та виправити проблеми, перш ніж вони вплинуть на процес розгортання, ваших кінцевих користувачів або сам бізнес.

Тому слід звернути увагу на наступні пункти:

- оптимізація коду;
- найкращі практики щодо обробки помилок;
- підтвердження коду відповідає вимогам безпеки;
- налаштування для виробничого середовища;
- використання одного пакетного менеджера
- міркування щодо розгортання.

Що стосується оптимізації коду, то однією з багатьох найкращих практик перед виробництвом є процес, який називається "підключення". Це передбачає запуск автоматизованого інструменту якості коду, такого як ESLint або JShint, через вашу кодову базу. Зазвичай він охоплює лише дуже основні питання якості, але в цьому полягає суть: він виявляє помилки, яких можна уникнути — і, як правило, їх дуже легко виправити — до того, як вони загрожують виробничій програмі.

Якщо порівняємо популярність пакетних менеджерів, то побачимо, що найпопулярніший - NPM від розробників Node.js (рис. 2).

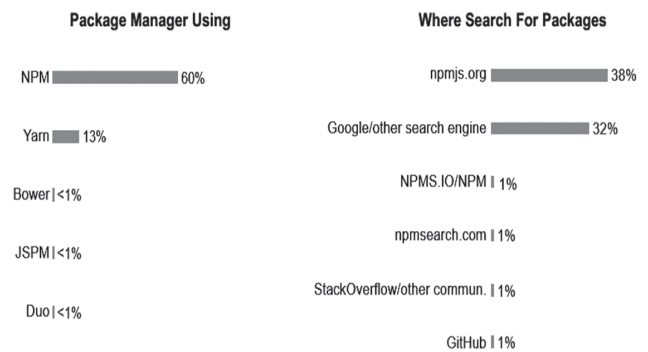


Рисунок 2. Використання менеджера пакетів

Джерело: <https://nodejs.org/en/user-survey-report/>

## 2. Розгортання програми

Охопивши основи ефективного робочого процесу перед виробництвом, далі розглянемо, чого очікувати і як реагувати протягом критичних перших годин після розгортання.

Розгортання корпоративної програми може бути важким. Приблизно до 30 відсотків усіх розгортань програм не вдається. Тим часом приблизно 80 відсотків організацій мають проблеми з випуском програмного забезпечення. Очевидно, що кожен, кому доручено розгортати програму, повинен бути готовий до того, що все піде не так — можливо, дуже неправильно.

Хоча надійний процес попереднього виробництва може допомогти мінімізувати вплив помилок конфігурації та інших проблем, яких можна уникнути, експерти-розробники Node.js повинні знати, як вирішувати загальні проблеми розгортання, особливо ті, що призводять до збоїв та проблем.

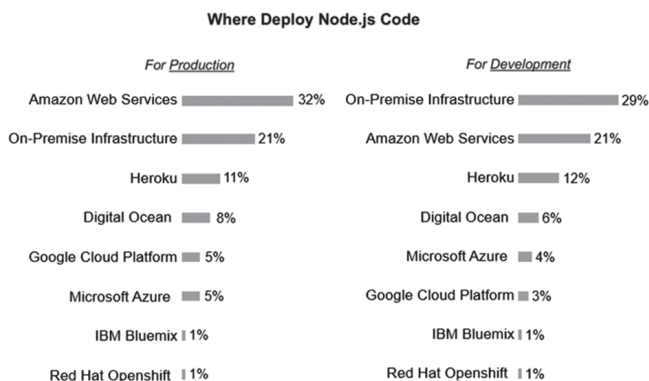
Типові проблеми, які можуть виникнути при розгортанні, включають:

- збій процесів Node.js;
- перевищення обмежень швидкості API;
- вирішення проблем із WebSocket;
- проблеми з залежностями;
- проблеми із завантаженням файлів;
- DDoS-атаки.

Хороша сторона проблем з розгортанням полягає в тому, що розробник дізнається багато нового про створення програм Node.js та про розгортання своїх програм із меншою кількістю проблем. Хоча проблеми можуть і будуть траплятися в майбутньому, справді серйозних проблем, ймовірно, буде менше.

Ще кращою стороною є, що як тільки ефективно усунуться основні проблеми розгортання, розробник матимете справу із більш стабільним та надійним додатком. Це, у свою чергу, звільняє від зосередження уваги на способах покращення продуктивності програми та модернізації власного процесу створення, тестування та розгортання програм Node.js.

Серед сервісів для деплою коду можна розглянути таких мастодонтів як AWS, Heroku, Digital Ocean. Більш детальніший список клауд платформ можна побачити на рис. 3.



**Рис. 3. Порівняння продуктивності версій Node.js**  
 Джерело: <https://nodejs.org/en/user-survey-report/>

### 3. Управління

Після успішного розгортання слід звернути увагу на управління вашим додатком Node.js. Хоча це не надто відрізняється від будь-якого іншого поширення додатків, є кілька особливостей, на які слід звернути увагу:

- витоки пам'яті;
- керування паралельністю Node.js;
- моніторинг.

Моніторинг продуктивності додатків є життєво важливим для підтримання стабільності розгортання вашого додатка та виявлення тонких регресій, які можуть призвести до уповільнення роботи програми або відмови, якщо її не встановити.

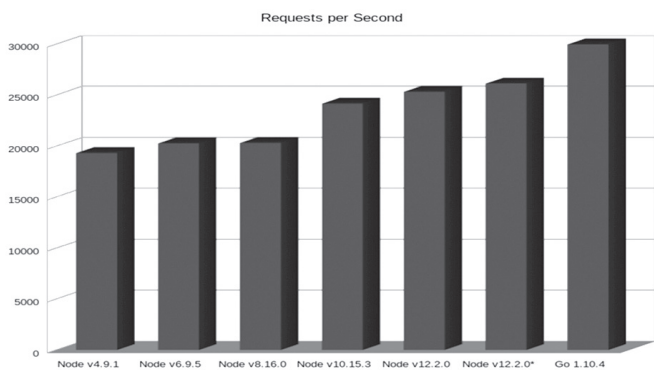
### 4. Експеримент

#### Різниця версій Node.js.

Регулярне оновлення версій Node.js дуже необхідне на річ, чим швидше відбувається оновлення на нову стабільну версію платформи Node.js, тим краще, в цілому, починає працювати сервер написаний з використанням Node.js.

Наприклад, різниця, в кількості запитів, що обробляються за секунду, між восьмою та дванадцятою версією складає понад 5 000 запитів. Тобто, лише оновлення з восьмої версії Node.js до дванадцятої дає вам приріст продуктивності сервера на 25-30 відсотків [1].

Побачити порівняння різних версій, та різницю їхньої продуктивності можна нижче на рис. 4.

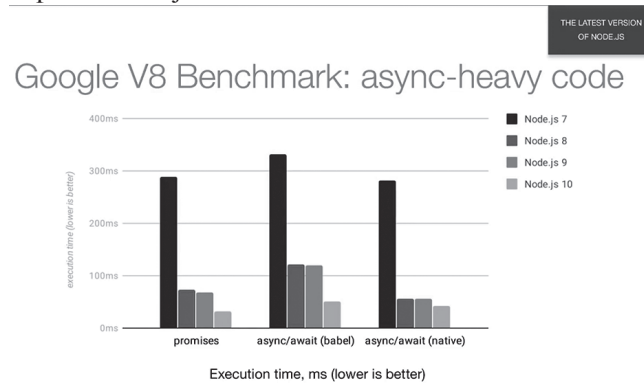


**Рис. 4. Порівняння продуктивності версій Node.js**  
 Джерело: складено авторами.

Також, після оновлення версії Node.js збільшиться й швидкість виконання вбудованих в V8 методів. Наприклад, різниця швидкості виконання методу find на V8 v7.0 відрізняється від V8 v7.1 майже на сто відсотків.

Різниця в швидкості виконання важкого асинхронного коду між Node.js 7 та Node.js 10 складає близько десяти разів. Для порівняння використовувалася одна й та сама функція з одними й тими ж параметрами, Node.js 7 виконала її за 290 мілісекунд, в той час, як Node.js 10 виконала її за 30 мілісекунд. На цьому прикладі чітко видно, що оновлення версії Node.js є дуже важливим та значно підвищує швидкість роботи вашого серверу, а значить й дає можливість обслуговувати більшу кількість користувачів.

На рис. 5 наведено приклади різниці у швидкості виконання однакового асинхронного коду на різних версіях Node.js.



**Рис. 5. Порівняння виконання асинхронного коду**  
 Джерело: складено авторами

#### Вибір пакетів Node.js.

Не завжди вбудовані в V8 методи є оптимальними с точки зору їх продуктивності та швидкості виконання. Наприклад, метод stringify, котрий вбудовано в V8 у багатьох випадках є менш продуктивним, ніж аналогічні методи із пакетів, що можна завантажити за допомогою node package manager (npm). Нижче, на рис. 6 наведено порівняння популярних npm модулів та стандартного JSON.stringify.

На даному рисунку чітко видно, що для деяких випадків, методи зі сторонніх пакетів відпрацьовують у дев'ять разів швидше. Через те, що даний метод дуже часто використовується при розробці серверів з використанням Node.js, то навіть заміна лише його на метод з зовнішньої бібліотеки може суттєво прискорити час відповіді енд-поентів, що використовують stringify.

Наступним прикладом вдалого вибору є бібліотеки Socket.IO та μWebSockets, якщо їх порівняти з вбудованим WS модулем в Node.js, то можна побачити, що найбільш популярний Socket.IO є найменш продуктивнішим, цей модуль обробляє приблизно 8 000 повідомлень, вбудований WS модуль – 35 000, а μWS – понад 200 000, тому що цей модуль повністю написано на C++ [2]. Фреймворк μWS має

кращу пропускну здатність коротких повідомлень, яка більша у 23 рази, у 21 раз кращу продуктивність з'єднання та у 52 рази менше використання пам'яті в порівнянні з вбудованим WS модулем.

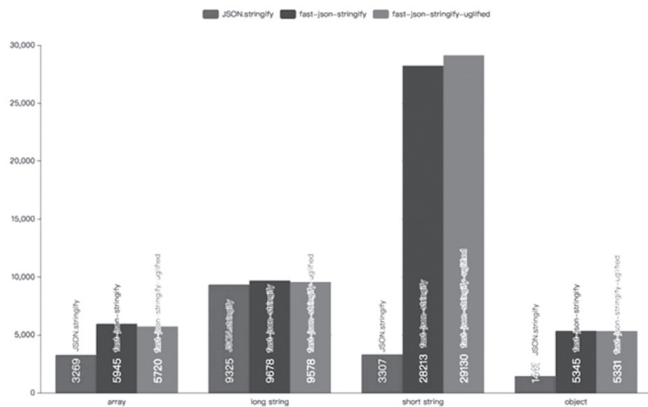


Рис. 6. Порівняння stringify методу

Джерело: складено авторами

Порівняння найпопулярніших фреймворків можна побачити на рис. 7.

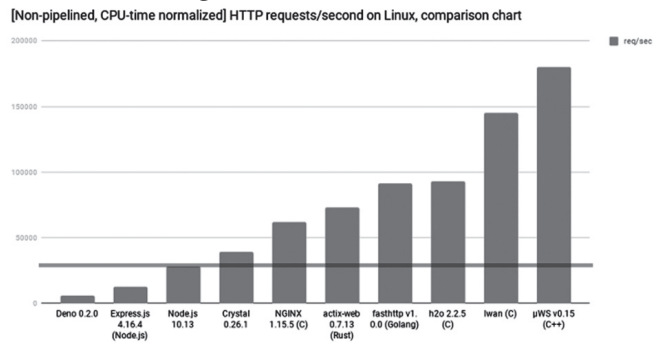


Рис. 7. Порівняння WS фреймворків

Джерело: складено авторами

Тому, якщо у додатку часто оновлюється інформація, наприклад, це крипто біржа чи тренінги літаків у реальному часі, має сенс використовувати такі високопродуктивні фреймворки, як  $\mu$ WS.

*Інші методи.*

Слід також зазначити, що є декілька методів зробити Node.js більш продуктивною:

- кластеризація Node.js додатку;
- використання інструментів для хешування;
- використання протоколу HTTP2;
- оптимізація V8 garbage collection.

Використання HTTP2 замість HTTP1.1 може призвести до збільшення швидкості обробки запиту у декілька разів [3]. Протестувавши один й той самий енд-поінт, але змінивши HTTP1.1 на HTTP2 вдалося досягти приросту більш ніж в 2 рази, завдяки більш досконалому і швидкому протоколу.

Також можна підвищити швидкість обробки запитів за допомогою хешування, використовуючи такі інструменти як Redis [4]. Використання даного інструменту дозволить зберегти певні дані в random access методу, що гарантує швидкий доступ до даних, аніж зробити вибірку з бази даних.

Кластеризація — дозволяє симулювати багатопоточність в Node.js за допомогою виконання додатку в кластерному режимі. Фактично, створюючи нові процеси з вашим додатком та балансує запити між усіма активними додатками за допомогою балансувальника. Це дозволить суттєво збільшити кількість запитів, що може обробити сервер [5].

Бажано, виносити важкі для CPU задачі з Node.js до веб серверу, наприклад, Nginx. Прикладі того, що можна винести з Node.js додатку до Nginx:

- HTTP/2;
- Gzip;
- SSL;
- роздачу статичних файлів;
- логування запитів та відповідей;
- метрики та моніторинг;
- авторизація.

## Висновки

Було обрано кращі шляхи прискорення додатків, та збільшення їх продуктивності. Використовуючи усі техніки, що було приведено вище, можна дуже добре оптимізувати та прискорити будь-який сервер написаний на платформі Node.js. А якщо була правильно обрана архітектура для сервера, то в результаті повинен вийти додаток, який можна легко масштабувати та підтримувати на протязі довгого часу. Також, у роботі було показано, як можна ефективно використовувати потужності сервера, на якому виконується Node.js додаток.

Дані техніки без проблем імплементуються на будь-якому сервері, та призводять, у деяких випадках, до загального приросту швидкості, доступності та продуктивності серверу у десять разів, а інколи і більше.

## Список літератури:

- [1] Young A., Meck B., Cantelon M. Node.js in Action, Second Edition, 2017. 392 p.
- [2] Herron D. Node.js Web Development: Server-side development with Node 10 made easy, 4th Edition, 2018. 494 p.
- [3] Dayley B., Dayley B. Node.js, MongoDB and Angular Web Development: The definitive guide to using the MEAN stack to build web applications, 2017. 636 p.
- [4] Casciaro M., Mammino L. Node.js Design Patterns: Design and implement production-grade Node.js applications using proven patterns and techniques, 3rd Edition, 2020. 660 p.
- [5] Pasquali S. Mastering Node.js: Expert techniques for building fast servers and scalable, real-time network applications with minimal effort, 2013. 346 p.
- [6] Thota N.R. Mastering Hyperledger Fabric: Master The Art of Hyperledger Fabric on docker, docker swarm and Kubernetes, 2020. 231 p.
- [7] Elliott E. Programming JavaScript Applications: Robust Web Architecture with Node, HTML5, and Modern JS Libraries, 2014. 254 p.
- [8] Mead A. Learning Node.js Development: Learn the fundamentals of Node.js, and deploy and test Node.js applications on the web, 2018. 658 p.

Надійшла до редколегії 12.11.2020



В.В. Білоцерковський<sup>1</sup>, С.Г. Удовенко<sup>2</sup>, Л.Е. Чала<sup>3</sup>

<sup>1</sup>ХНУРЕ, м. Харків, Україна,

vladyslav.bilotserkovskiy@nure.ua, ORCID iD: 0000-0003-4001-0660

<sup>2</sup>ХНЕУ ім. С. Кузнеця, м. Харків, Україна,

serhiy.udovenko@hneu.net, ORCID iD: 0000-0001-5945-8647

<sup>3</sup>ХНУРЕ, м. Харків, Україна,

larysa.chala@nure.ua, ORCID iD: 0000-0002-9890-4790

## МЕТОД НЕЙРОМЕРЕЖЕВОГО РОЗПІЗНАВАННЯ ФАЛЬСИФІКОВАНИХ ЗОБРАЖЕНЬ

Розглянуто методи генерації зображень, фальсифікованих за допомогою технологій Deepfake, і методи їх виявлення. Пропонується метод виявлення фальсифікованих зображень, який оснований на спільному використанні ансамблю згорткових нейронних моделей, механізму Attention та стратегії сіамського навчання мережі. Ансамблі моделей формувалися різними способами (з використанням двох, трьох або більшої кількості складових). Результат обчислювався як середнє значення показників AUC і LogLoss з усіх моделей, що входять в ансамбль. Такий підхід дозволяє покращити точність різних нейромережових класифікаторів для виявлення статичних та динамічних зображень, створених за технологіями Deepfake.

ТЕХНОЛОГІЯ DEEPFAKE, РОЗПІЗНАВАННЯ ФАЛЬСИФІКОВАНИХ ЗОБРАЖЕНЬ, ЗГОРТКОВА НЕЙРОННА МЕРЕЖА, ATTENTION МЕХАНІЗМ, АНСАМБЛЬ МОДЕЛЕЙ

**Белоцерковский В.В., Удовенко С.Г., Чала Л.Э. Метод нейросетевого распознавания фальсифицированных изображений.** Рассмотрены методы генерации изображений, фальсифицированных с помощью технологий Deepfake, и методы их выявления. Предлагается метод выявления фальсифицированных изображений, основанный на совместном использовании ансамбля сверточных нейронных моделей, механизма Attention и стратегии сиамского обучения сети. Ансамбли моделей формировались разными способами (с использованием двух, трех или более составляющих). Результат исчислялся как среднее значение показателей AUC и LogLoss из всех моделей, входящих в ансамбль. Такой подход позволяет улучшить точность нейросетевых классификаторов для выявления статических и динамических изображений, созданных по технологиям Deepfake.

ТЕХНОЛОГИЯ DEEPFAKE, РАСПОЗНАВАНИЕ ФАЛЬСИФИЦИРОВАННЫХ ИЗОБРАЖЕНИЙ, СВЕРТОЧНАЯ НЕЙРОННАЯ СЕТЬ, ATTENTION МЕХАНИЗМ, АНСАМБЛЬ МОДЕЛЕЙ

**Bilotserkovskyy V.V., Udovenko S.G., Chala L.E. Method of neural network recognition of falsified images.** Methods for generating images falsified using Deepfake technologies and methods for detecting them are considered. A method for detecting falsified images is proposed, based on the joint use of an ensemble of convolutional neural models, the Attention mechanism and a Siamese network learning strategy. The ensembles of models were formed in different ways (using two, three or more components). The result was calculated as the average value of the AUC and LogLoss indices from all the models included in the ensemble. This approach improves the accuracy of convolutional neural network classifiers for detecting static and dynamic images created using Deepfake technologies.

DEEPFAKE TECHNOLOGY, FALSE IMAGE RECOGNITION, CONVOLUTIONAL NEURAL NETWORK, ATTENTION MECHANISM, MODEL ENSEMBLE

### Вступ

Розвиток інтелектуальних технологій обробки даних призвів до стрімкого поширення підроблених статичних та динамічних (відеозаписів) зображень, що часто використовуються з метою маніпуляції соціумом. При цьому відбувається розповсюдження дезінформації, що призводить до значного викривлення інформаційного простору. Такі викривлення можуть мати зазвичай негативні наслідки [1, 2].

Діпфейк (англ. Deepfake) – методика створення зображення чи відео, що основана на технологіях штучного інтелекту. Назва походить від об'єднання термінів deer (англ. deer – глибинний, тобто пов'язаний з використанням технологію глибинного навчання) та fake (англ. fake – підробка) [3].

С кожним роком процес розвитку глибинного навчання стає дедалі ефективнішим завдяки

збільшенню потужності техніки та інформаційним збагаченням. Технології Deepfake на сьогоднішній день становлять загрозу для інформаційного суспільства, створюючи умови для генерування провокаційних медіа матеріалів. Питання того, як розпізнавати та боротись з негативним застосуванням технологій Deepfake охоплює медію, правову та технологічну складові, що різняться своєю гнучкістю, пошуком варіантів вирішення, а також способами впровадження та подальшої імплементації.

Методи генерації фальсифікованих за допомогою технологій Deepfake зображень та методи їх виявлення спираються на дуже схожі принципи. Більшість таких методів використовують різні варіанти так званих генеративно-змагальних штучних нейронних мереж, що складаються з двох основних елементів – генератора та дискримінатора [4].

Обидва елементи спільно навчаються на основі протилежних цілей. Мета генератора – створити фейкове зображення, яке важко відрізнити від реального, а мета дискримінатора – виявити синтетично створене зображення.

Таким чином, актуальною є проблема відокремлення фальсифікованих статичних або динамічних зображень (зокрема, зображень обличчя людини), створених за допомогою технологій Deepfake, від оригінальних.

У даній роботі пропонується підхід до виявлення фальсифікованих зображень, який оснований на спільному використанні ансамблю генеративно-змагальних моделей та механізму Attention. Такий підхід може привести до покращення точності різних нейронмережових класифікаторів для виявлення зображень, створених за технологіями Deepfake.

### 1. Технології генерації фальсифікованих зображень

Генерація цифрових зображень, як статичних, так і динамічних, є однією з найбільш витратних за кількістю операцій задачею. Для генерації зображень (а також для їх фальсифікації) зазвичай використовують спеціальні архітектури нейронних мереж для роботи із зображеннями, без якої будь-яка їх машинна підробка була б неможливою. До таких архітектур належать, насамперед, згорткові нейронні мережі (ЗНМ, або CNN – convolutional neural network), націлені на ефективно за потужностями та точністю розпізнавання зображень, що входять до складу технологій глибокого навчання [5, 6]. Основна особливість згорткових нейронних мереж полягає в чергуванні двох основних шарів (convolution layers і subsampling layers). Структура цієї мережі є односпрямованою та багат шаровою. Для її навчання використовуються стандартні методи, наприклад, метод зворотного поширення помилки [7]. На сьогоднішній день найрозповсюдженішим методом розпізнавання зображень та їх фрагментів (зокрема, обличчя людини) за допомогою ЗНМ є метод з використанням алгоритму, що передбачає побудову біометричних шаблонів (дескрипторів) по зображенню і пошуку заданого шаблону в базі вже обчислених дескрипторів. Класичні архітектури нейронних мереж з повнозв'язними шарами недоцільно застосовувати на практиці для аналізу зображень, оскільки кількість параметрів таких мереж експоненціально зростає з розміром вхідних даних і кількістю шарів.

Для синтезу фейкових зображень за допомогою ЗНМ останнім часом активно використовуються згорткові генеративно-змагальні мережі DCGAN (deep convolutional generative adversarial networks) [8].

Основна ідея побудови генеративно-змагальних мереж полягає у навчанні пари мереж в процесі їх

постійного змагання. У випадку генерації фальсифікованих зображень з використанням DCGAN мережа-генератор (G) намагається створити фальсифіковане зображення, яке важко буде відрізнити від справжнього. Мережа-дискримінатор (D), у свою чергу, намагається розрізнити підробки і справжні зображення. Важливо, що генератор ніколи не отримує реальні дані, на вхід подається тільки випадковий вектор (джерело ентропії, що іноді інтерпретується як простір прихованих змінних, latent-space). Єдиний спосіб навчання для нього – взаємодія з дискримінатором. Дискримінатор при цьому отримує на вході або створені генератором дані, або об'єкт реальної навчальної вибірки. Помилка навчання дискримінатора розраховується з урахуванням того, звідки прийшли дані. В процесі навчання (рис.1) генератор навчається розподілу початкової вибірки і починає створювати дані усе більш близькі до реальних в той час, як дискримінатор стає більш точним в розпізнаванні підробки від оригіналу.

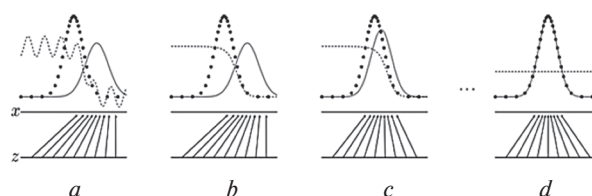


Рис. 1. Ілюстрація до процесу навчання генеративно-змагальної мережі

GAN мережі навчаються одночасно, оновлюючи розподіл дискримінатора (D, переривчаста лінія), який показує ймовірність належності розподілу генератора (G, безперервна лінія) до розподілу реальних даних (чорна точкова лінія). Нижня горизонтальна лінія – це область визначення, з якої вибирається z (в цьому випадку рівномірно). Верхня горизонтальна лінія вища – область визначення для x. В процесі навчання відбувається уточнення залежності  $x=G(z)$ . На останній ітерації (d) дискримінатор вже практично не може відрізнити реальні дані від згенерованих.

Мережі, які лежать в основі генератора і дискримінатора, є багат шаровими мережами, що складаються із згортальних і повнозв'язних шарів. D і G можуть відрізнитися і не обов'язково, щоб вони були повністю дзеркальними. Оскільки завдання генератора полягає в зіставленні простору прихованих змінних в простір даних, то можна це записати у вигляді:

$$G : G(z) \rightarrow R^{|x|}, \tag{1}$$

де  $z \in R^{|z|}$  – вибірка з простору прихованих змінних;  $x \in R^{|x|}$  – справжні дані.

Для дискримінатора загальна функція зіставлення даних у ймовірність їх належності до істинних може бути представлена таким чином:

$$D : D(x) \rightarrow (0,1). \tag{2}$$

Крім мереж DCGAN, для генерації фальсифікованих зображень за технологіями Deepfake широко використовуються згорткові мережі типу автоенкодер (Autoencoder – AE). Автоенкодер – це тип багатопарової згорткової нейронної мережі, яка має здійснювати таку автоасоціативну апроксимацію функції, щоб її вихідний сигнал якомога точніше відповідав значенню вхідного сигналу. Таке тотожне співвідношення виявляється не тривіальним, якщо накласти деякі специфічні обмеження на нейронну мережу автоенкодера. Такими обмеженнями можуть бути, перш за все, обмеження кількості нейронів на прихованому шарі і накладення критеріїв розрідженості на активацію цих нейронів. Слід зазначити, що в загальному випадку AE характеризується наступними особливостями: має симетричну структуру; містить непарну кількість шарів; складається з кодера і декодера; містить так зване «пляшкове горлечко», тобто вихідний шар кодера (вхідний шар декодера), в якому використовуються лінійні функції активації; кількість нейронів в «пляшковому горлечку» має бути менше, ніж розмірність вхідних даних, або ж повинна відбуватися лише часткова активація цих нейронів; у вхідному і вихідному шарах AE зазвичай використовуються гаусові функції активації.

Симетричність архітектури значно полегшує завдання настройки параметрів AE, оскільки необхідно визначати лише половину всіх параметрів мережі.

Розглянемо класичний AE з одним прихованим шаром. Позначимо вхідні сигнали мережі як  $x(1), x(2), \dots, x(n)$ .

Спочатку AE перетворює (кодує) вхідний сигнал  $x \in [0, 1]^d$  у деяке внутрішнє представлення  $y \in [0, 1]^d$  з використанням перетворення наступного вигляду:

$$y = s(Wx + b), \tag{3}$$

де  $W$  та  $b$  – загальні матриці ваг і зміщень мережі;  $s$  – нелінійна функція перетворення (наприклад, гіперболічний тангенс).

Потім внутрішнє представлення (код) вхідного сигналу  $y$  перетворюється (декодується) в сигнал  $z$ , який є реконструкцією вхідного сигналу і має таку ж розмірність. Дане перетворення можна записати в такий спосіб:

$$z = s(W'y + b'), \tag{4}$$

де  $W'$  і  $b'$  – настроювані матриці ваг і зміщень мережі декодуєчого перетворення.

Параметри моделі ( $W, W', b, b'$ ) налаштовуються таким чином, щоб мінімізувати помилку реконструкції вхідного сигналу, що може бути здійснено з використанням різних функцій втрат, наприклад, квадратичної функції або крос-ентропії.

Вдосконаленим варіантом AE, що може бути використаний для створення фальсифікованих зображень за технологіями Deepfake, є так званий

шумопригнічуючий автоенкодер (ШАЕ – DAE – Denoising Autoencoder). ШАЕ є модифікацією AE, в якій входи поєднуються з деякими завданнями і система навчається для відновлення даних без перешкод [9]. Таким чином, ШАЕ можна розглядати як стохастичне розширення класичного автокодера, яке змушує модель навчатися відновленню вхідного сигналу при подачі на вхід його зашумленої версії. Схема навчання ШАЕ може використовуватися в технологіях Deepfake для шумопригнічення і є ефективним способом навчання автоенкодера більш значущим прихованим уявленням даних.

Базовий варіант загальної архітектури ШАЕ представлений на рис. 2.

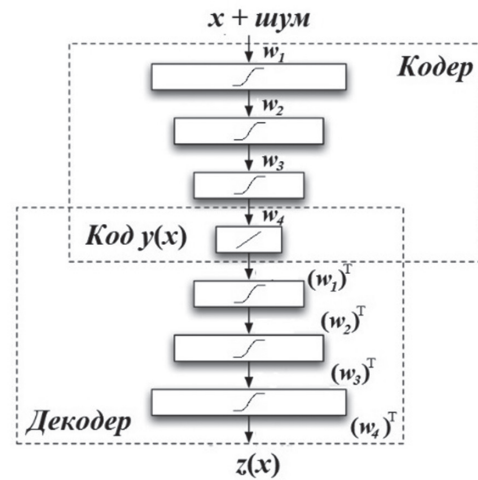


Рис. 2. Загальна структура ШАЕ

Технологія DeepFake використовує зазвичай два AE (або ШАЕ) з однаковими кодерами, але різними декодерами (А та В), що дозволяє мати однаково побудований базовий вектор для різних зображень, що і робить можливою деформацію оброблюваних зображень (зокрема, зображень людських обличч) надалі. Упродовж фази навчання, дві мережі відпрацьовують окремо. Декодер А навчається тільки на обличчях першого AE, а декодер В – на обличчях другого AE. В той же час усі базові вектори отримуються з використанням однієї і тієї ж архітектури кодерів, тобто кодер визначає загальні ознаки в обох зображеннях.

Після закінчення процесу навчання базовий вектор суб'єкта А передається в декодер суб'єкта В, який намагається відтворити суб'єкт В за інформацією, отриманою від суб'єкта А. Якщо мережа добре узагальнює особливості зображення, та базовий вектор буде враховувати вирази обличчя та їх розташування.

Видзначимо, що для отримання задовільних результатів необхідно мати від 200 до 2000 зображень. Навчання таких моделей є дуже витратним з погляду на обчислювальну складність (наприклад, щоб згенерувати однохвилинне відео і, треба близько 18 хвилин на одній відеокарті). Архітектуру DeepFake можна оптимізувати багатьма способами, проте найбільш

поширеними з них є комбінування моделей з GAN архітектурою. Генератори поступово вчаться створювати найбільш реалістичне зображення суб'єкта, а дискримінатори — визначати, яка з них згенерована, а яка оригінальна. За рахунок навчання GAN генераторів синтезоване зображення стає все більш реалістичним. Таким чином, Encoder і Decoder відповідають за перенесення зображення, а дискримінатор генеративних мереж — за покращення результату [8]. Цей підхід продемонстровано на рис. 3.

Ще один підхід передбачає використання архітектури GAN, що складається з декількох генеративно-змагальних мереж, кожна з яких відповідає за свою операцію. Для навчання такої нейромережі потрібен потужний кластер відеокарт. Незважаючи на це, такий підхід є найбільш перспективним, тому що дає кращий результат. Залучивши до цього підходу рекурентні мережі, можна ще більше покращити якість DeepFake. Рекурентні мережі дозволяють отримувати базовий вектор, ґрунтуючись не лише на одному зображенні, а на декількох зображеннях. Базовий вектор генерується на поточному зображенні, а також зображеннях з відео створених за декілька секунд до і після початкового зображення. При цьому переходи з одного стану і положення обличчя до іншого стають набагато реалістичніше.

Одним з останніх варіантів використання згорткових генеративно-змагальних мереж для генерації підроблених статичних та динамічних (відео) зображень є модель FSGAN (Face Swapping GAN). Головна особливість цієї моделі полягає в тому, що вона може застосовуватися до конкретних зображень без необхідності попереднього навчання на них [9]. Наприклад, у моделі FSGAN, що застосовується для зображень людських облич, одна нейромережа вчиться підганяти зображення обличчя під параметри цільового відео (поворот голови, нахил убік або вперед), друга переносить риси обличчя, а третя здійснює операцію image blending (злиття зображень), щоб зображення було більш реалістичним (без ушкоджень або артефактів). Особливістю такої технології DeepFake є наявність рекурентної нейронної мережі для реконструкції зображень осіб, яка коригує як їх позу, так і різні вирази обличчя.

Для відеозображень використовується підхід безперервної інтерполяції представлень обличчя на основі реконструкції, триангуляції Делоне і барицентричних координат. Закриті ділянки обличчя обробляються окремою мережею, а для об'єднання двох зображень обличч застосовується мережа, яка зберігає цільовий колір шкіри і особливості освітлення. На рис. 4. продемонстрований принцип використання моделі FSGAN.

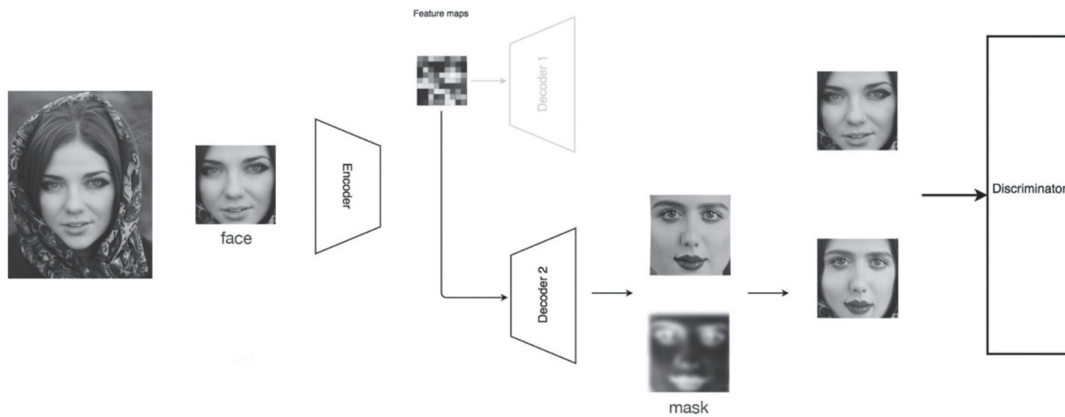


Рис. 3. Застосування GAN для покращення результатів DeepFake

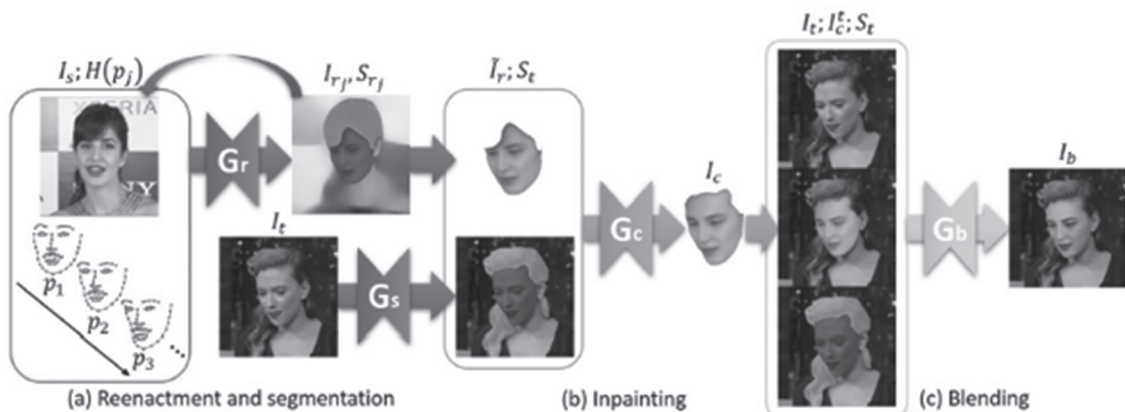


Рис.4. Принцип використання моделі FSGAN

На рис. 4  $I_s$  є початковим зображенням обличчя  $F_s$ , а  $I_t$  – підсумковим зображенням обличчя  $F_t$ . Алгоритм створює нове зображення, використовуючи  $I_t$ , та намагаючись замінити  $F_t$  на  $F_s$ , зберігаючи його положення і вираз. FSGAN складається з кількох компонентів. По перше, це генератор інсценування  $G_g$  і згортальна мережа для сегментації ділянок обличчя  $G_s$ . Генератор  $G_g$  визначає опорні точки обличчя  $F_t$  та формує таке зображення  $I_g$ , щоб  $F_g$  відображало  $F_s$  з таким же положенням і виразом, як у  $F_t$ . Він також обчислює маску сегментації  $S_g$  обличчя  $F_g$ . Мережа  $G_s$  сегментує особу і волосся обличчя  $F_t$ . На виділеному зображенні  $I_g$  можуть бути відсутніми деякі частини обличчя. Далі застосовується мережа  $G_c$ , що використовує сегментацію  $S_t$  та виявляє пропущені пікселі. Остання частина моделі об'єднує зображення обличчя  $F_c$  з початковим зображенням  $I_t$  для отримання кінцевого результату.

## 2. Методи виявлення фальсифікації зображень

Для виявлення підроблених зображень найчастіше використовується згорткова рекурентна мережа LSTM (Long Short Term Memory). Ця модель складається з двох основних блоків: згортальної мережі для витягання елементів зображення і блоку аналізу частини послідовності зображень [3]. Архітектура моделі представлена на рис. 5.

Отримуючи послідовність тестових зображень, модель витягає набір функцій для кожного кадру за допомогою згортальної мережі. Після цього функції декількох послідовних кадрів об'єднуються і передаються моделі LSTM для аналізу. Рекурентна мережа видає ймовірність правливості послідовності.

Як вхідні дані використовується послідовність векторів ознак, витягнутих згортальною мережею з поданих на вхід кадрів, і згортальна мережа з двома вузлами для визначення того, чи є зображення згенерованим або оригінальним відео. LSTM мережа є проміжним блоком системи і не вимагає додаткового навчання.

Для демонстрації точності моделі в роботі [3] було розглянуто 300 відеозображень, згенерованих за допомогою DeepFake. Автори моделі перевіряли

точність, використовуючи 20, 40 і 80 послідовних кадрів. Результати продемонстровано в табл. 1.

Таблиця 1

Оцінка точності генерації DeepFake-зображень з використанням рекурентних моделей

Модель	Training acc. (%)	Validation acc. (%)	Test acc. (%)
Conv-LSTM, 20 fraimes	99,5	96,9	96,7
Conv-LSTM, 40 fraimes	99,3	97,1	97,1
Conv-LSTM, 80 fraimes	99,7	97,2	97,1

Модель демонструє непогану точність, але вона може працювати тільки з послідовностями підроблених відеозаписів.

При використанні цього підходу, було зроблено спостереження, що хоча GAN-подібні моделі здатні генерувати фотореалістичні візуальні і геометричні сигнали, які виходять за рамки можливостей виявити їх людським оком, біологічні сигнали, приховані природою, як і раніше складно відтворити. Біологічні сигнали є невід'ємною частиною відео із зображенням осіб, яку у свою чергу є сферою застосування Deepfake технології.

Ще одна технологія виявлення Deepfake зображень використовує компактну нейромережу MesoNet. [10]. Ця технологія дозволяє ефективно розпізнавати підробні відеозаписи, згенеровані за допомогою алгоритмів DeepFake або Face2Face. Дослідження показали, що створити одну узагальнену модель для ефективного виявлення підробок, створених цими алгоритмами одночасно, за допомогою цієї архітектури неможливо. Тому доцільним є застосування для виявлення деяких типів підробок зображень різних варіантів нейромережі MesoNet з невеликою кількістю шарів, зокрема мереж Meso-4 та MesoInception4.

Meso-4 використовує чотири блоки згортки та пулінг з одним прихованим шаром. Згорткові шари використовують активаційну функцію ReLU, щоб запобігти ефекту розмитого градієнту. Для регуляризації і підвищення надійності в Meso-4 використовується Dropout шар.

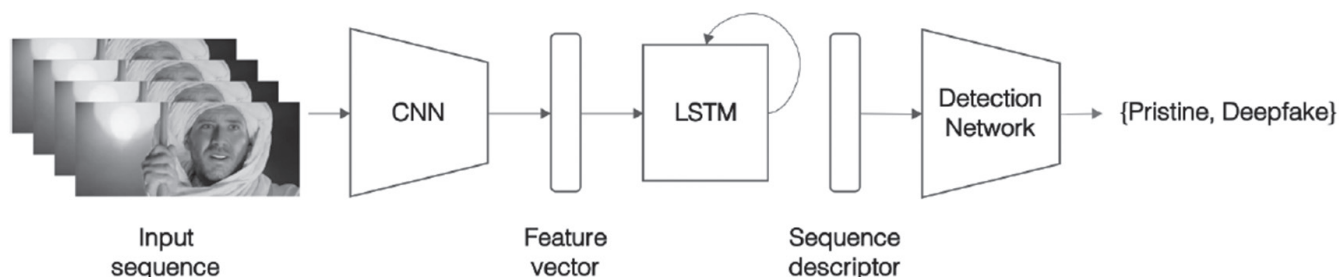


Рис. 5. Архітектура рекурентної моделі LSTM

Альтернативна модель MesoInception4 відрізняється від Meso-4 тим, що замість перших двох згорткових блоків використовується Inception модуль. Ідея цього модуля полягає в об'єднанні кількох згорткових шарів різної розмірності для збільшення розмірності простору в якому модель оптимізується. Основною перевагою цієї моделі є її компактність. У моделі відносно мало параметрів, завдяки чому навчання відбувається швидко.

Розглянемо можливість виявлення зображень, згенерованих за допомогою технологій Deepfake, за методом парного навчання, що використовує модифіковану структуру згорткової мережі, яка зветься мережею із загальними підробними функціями (CFFN – Cascade Feed Forward Networks). Ця структура використовує детектор глибокої підробки (DeepFD) для ідентифікації підробного обличчя і початкових зображень, а також двоетапний метод навчання, який поєднує в собі функцію CFF на основі стратегії парного навчання і навчання класифікатора (рис. 6).

Представлення стратегії контрольованого навчання у виявленні підробних зображень обличчя пов'язані з проблемами пошуку навчальної вибірки, яка б містила зображення, згенеровані різними варіантами GAN. Щоб здолати ці проблеми, підробні і реальні зображення об'єднуються в пару і слідує одне за одним, використовуючи парну інформацію для побудови порівняльної функції втрат, щоб упізнати відмітну загальну фальсифіковану особливість (CFF) за пропонованим CFFN. Як тільки функція розпізнавання CFF навчена, класифікатор використовує дискримінаційну функцію CFF, щоб ідентифікувати, чи є зображення реальним або підробним.

Чимало варіантів мереж CNN (наприклад, DenseNet, ResNet і Xception) можуть бути використані для виявлення ознак фальшування з тренувального набору та для створення класифікатора. Проте,

більшість з цих навчаються контрольованим чином, тому ефективність класифікації залежить від системи навчання.

Відомо, що CNN ефективно використовуються тільки в наданні ознак високого рівня, щоб ідентифікувати, чи є зображення підробним. Проте CFF підробних зображень можуть існувати не лише в представленні високого рівня, але також і в представленні об'єктів середнього рівня. Тому міжрівневі елементи можуть бути інтегровані в класифікаційний шар для підвищення продуктивності розпізнавання підробних зображень. Для навчання відповідної моделі доцільно використовувати дві функції втрат для оптимізації мережі CFFN і класифікатора одночасно. Проте зазвичай важко визначити вагові значення для двох функцій. Оскільки основною метою пропонованого CFFN є навчання дискримінаційних ознак, це дозволяє навчати CFF шляхом мінімізації контрастних втрат в першу чергу. Після цього будь-який класифікатор можна використовувати для розпізнавання підробного зображення обличчя на основі навченої CFF. Отже, каскадна мережа CFFN спочатку навчається на основі контрастних втрат, а потім оптимізується мережа класифікаторів шляхом мінімізації втрат від перехресної ентропії.

Основним недоліком контрольованого навчання за розглянутим підходом є те, що важко визначити фрагмент, який виключений з етапу навчання. Щоб підвищити продуктивність пропонованого методу, вводяться контрастні втрати при вивченні CFF шляхом парного навчання. Варіант відповідної архітектури CFFN наведено на рис. 7.

Основною проблемою цього методу є те, що модель CFFN не здатна класифікувати всі можливі підробки, які згенеровані за допомогою базових алгоритмів DeepFake. Зображення, створенні за допомогою деяких сучасних модифікацій згорткових

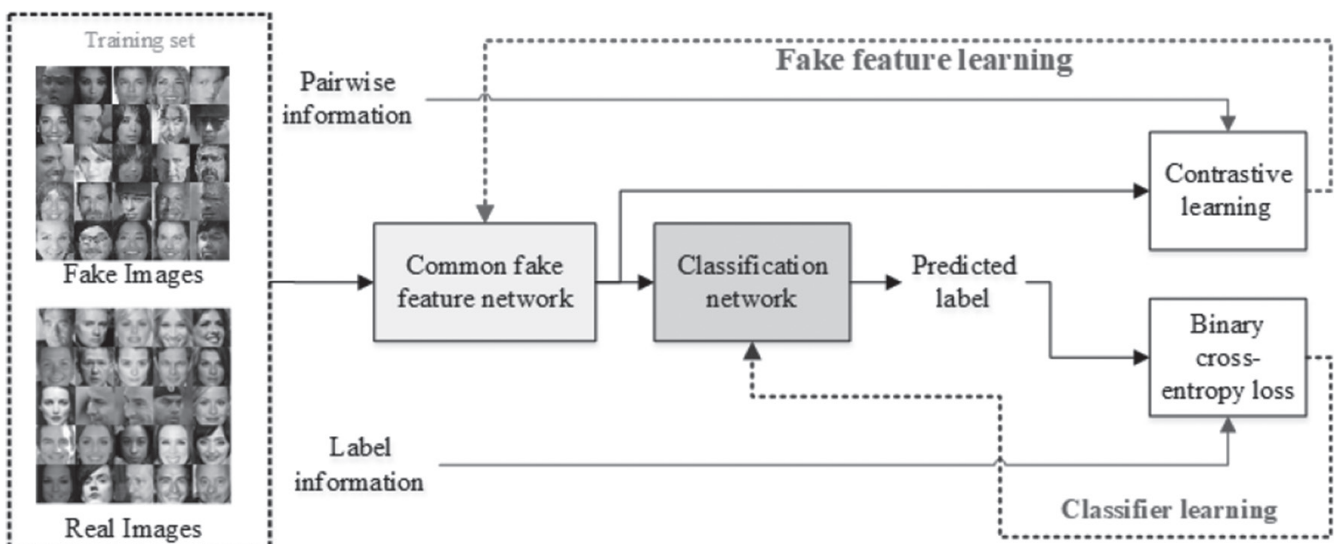


Рис. 6. Метод на основі CFF і стратегії парного навчання

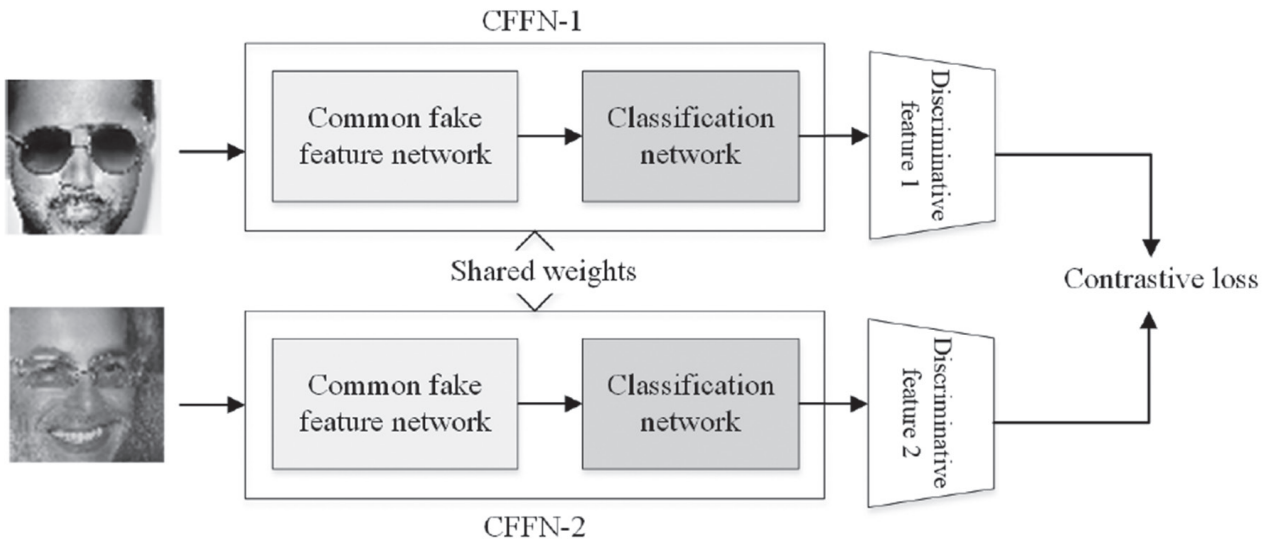


Рис. 7. Архітектура CFFN

нейронних мереж, таких як FSGAN, в основному класифікуються як початкові.

Існуючі методи класифікації фальсифікованих зображень можуть показувати непогані результати на невеликих наборах даних. Проте внаслідок того, що ці моделі мають невеликі розміри нейромереж, то на великих наборах даних результати суттєво погіршуються. Тому доцільним є створення узагальненого класифікатора на основі ансамблю з описаних згорткових моделей, що використовуються для виявлення Deepfake зображень.

### 3. Пропонований метод виявлення фальсифікованих зображень

Пропонований метод ґрунтується на концепції виявлення підроблених зображень за допомогою ансамблю згорткових нейронних мереж.

Відомо, що ансамбль моделей може привести до значного покращення точності різних класифікаторів на основі згорткових мереж для виявлення різних високорівневих семантичних особливостей, які

доповнюють одна одну, позитивно впливаючи на точність ансамблю для цієї конкретної проблеми.

Для цього в якості відправної точки було обрано сімейство моделей EfficientNet, які перспективними у багатьох задачах обробки зображень [11]. Цей набір архітектури досягає більшої точності та ефективності в порівнянні з іншими сучасними CNN. Приклад цієї архітектури продемонстрований на рис. 8.

Враховуючи особливості архітектури EfficientNet, пропонується здійснити дві модифікації, щоб зробити модель вигідною для ансамблювання. З одного боку, пропонується додати до моделі Attention механізм, який реалізує аналітичний метод, що дозволяє визначити, яка частина досліджуваного відео є більш інформативною для процесу класифікації. З іншого боку, до мережі пропонується додати сіамські стратегії навчання для екстраполяції додаткової інформації про дані.

Наведемо опис архітектури EfficientNet з пропонованим Attention механізмом і стратегіями навчання мережі. Серед сімейства моделей EfficientNet

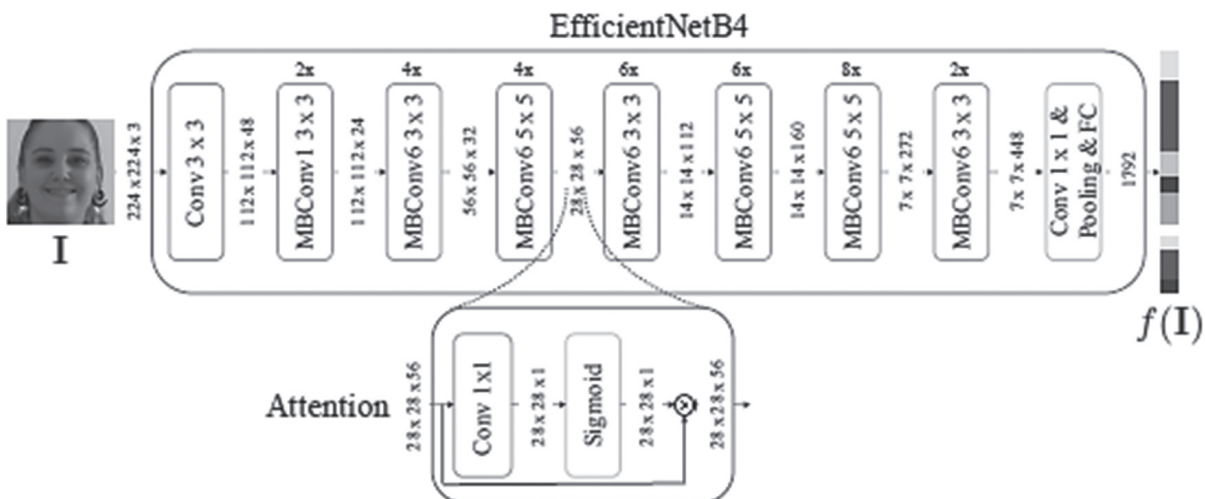


Рис. 8. Архітектура EfficientNetB4

була вибрана EfficientNetB4 в якості базового рівня для цих моделей. Цей вибір мотивований можливістю компромісу, пропонованого цією архітектурою, з точки зору вимірів (тобто кількості параметрів), часу виконання (тобто вартості FLOPS) і класифікацією продуктивності. З 19 мільйонами параметрів і 4,2 мільярдами FLOPS, EfficientNetB4 досягає 83,8% точності, що є кращим результатом, на наборі даних ImageNet. У тому ж наборі даних XceptionNet, використовуваний в якості базового методу виявлення маніпуляцій з обличчям, досягає 79% точність для 23 мільйонів параметрів і 8,4 мільярдів FLOPS.

Входом в мережу EfficientNetB4 є квадратне кольорове зображення, тобто в експериментах використовується обличчя, виділене з відеокадру. Насправді, зазвичай рекомендується відстежувати інформацію про обличчя, а не використовувати повний кадр в якості вхідних даних для мережі для підвищення точності класифікації. Крім того, обличчя можуть бути легко використані з зображень за допомогою будь-якого з існуючих детекторів зображень облич. Вихід мережі є вектором ознак з заданого числа елементів, визначуваним як  $f(I)$ . Остаточний результат, пов'язаний з цією функцією, характеризує якість класифікації.

Алгоритми архітектур Transformer або Residual Attention Networks дозволяють нейронній мережі визначити, яка частина її вхідних даних, частина зображення або послідовності слів є актуальною для виконання поставленого завдання. У контексті виявлення зображень, згенерованих за допомогою технології DeepFake, дуже корисно знати, яка саме частина входу дає нейронній мережі найбільшу частину інформації для ухвалення рішення. Механізм Attention реалізується шаром нейромережі, який має дві основні особливості:

- на вхід подається карта ознак, витягнута за допомогою EfficientNetB4 до певного шару. Вона обривається так, щоб ці ознаки мали достатню інформацію про вхідний кадр, не будучи занадто детальними або, навпаки, занадто невизначеними;

- ознаки обробляються тільки одним згортальним шаром розмірністю 1, після чого передаються в активаційну сигмоїдальну функцію для отримання єдиної Attention карти.

З одного боку, цей простий механізм дозволяє мережі зосередитися тільки на найбільш вагомих частинах ознак об'єктів, з іншого боку, він дає глибше розуміння того, які частини вхідного слою мережа вважає найбільш інформативними. Дійсно, отримана Attention карта може бути легко зіставлена з вхідним зразком, підкреслюючи, яким елементам було приділено більше уваги нейронною мережею. Результат Attention блоку остаточно обробляється блоками EfficientNetB4, що залишилися. Отримана модель була названа EfficientNetB4Att.

Навчання моделей відбувається за стратегіями наскрізного та сіамського навчання. Перша стратегія є більш класичною і використовує метрику оцінки на наборі даних DFDC (Deepfake Detection Challenge). Інша стратегія спрямована на використання можливостей узагальнень, пропонованих нейронними мережами, для отримання дескриптора функції, який дає перевагу подібності між вибірками, що належать до одного й того ж класу. Кінцева мета полягає в тому, щоб навчити представлення в просторі декодованих шарів мережі, які добре розділяють вибірку зображень (здебільшого зображень обличчя) реального і фальсифікованого класів.

Під час наскрізного навчання мережа отримує зразки зображення обличчя і формує оцінку  $\hat{y}_i$ , яка ще не пройшла через сигмоїдальну функцію активації. Оновлення ваг здійснюється за допомогою функції LogLoss:

$$L_L = -\frac{1}{N} \sum_{i=1}^N [y_i \log(S(\hat{y}_i)) + (1 - y_i) \log(1 - S(\hat{y}_i))], \quad (5)$$

де  $\hat{y}_i$  – змінна, що представляє  $i$ -у оцінку обличчя;  $N$  – загальна кількість зображень облич, використовуваних для навчання;  $S(\hat{y}_i)$  – активаційна сигмоїдальна функція.

Визначимо, що  $\hat{y}_i \in \{0,1\}$  та відповідає міткам обличчя. Зокрема, мітка 0 пов'язана з зображеннями облич, що виходять із справжніх початкових відеозаписів, а мітка 1 пов'язана з подробицями зображеннями.

Процедура сіамського навчання (triplet margin loss) була адаптована насупним чином:

$$L_T = \max(0, \mu + \delta_+ - \delta_-), \quad (6)$$

$$\delta_+ = \|f(I_a) - f(I_p)\|_2,$$

$$\delta_- = \|f(I_a) - f(I_n)\|_2,$$

де  $\delta_+$  та  $\mu$  – позитивні значення;  $I_a$  – реальне зображення;  $I_p$  – позитивний зразок, що належить до того ж класу, що й  $I_a$  (інше реальне зображення);  $I_n$  – зображення, згенероване за допомогою DeepFake.

Навчання завершується настроюванням шару класифікації поверх мережі, наслідуючи описаний вище наскрізний підхід.

#### 4. Експериментальні дослідження

Експерименти проводилися на двох наборах даних: FF++ та DFDC [12]. FF++ – це великий набір даних по маніпуляціях з обличчям, створений з використанням state-of-the-art методів редагування відеозаписів. Цей набір даних містить два класичних підходи маніпуляції обличчями, а саме Face2Face і FaceSwap, разом з двома стратегіями, ґрунтованими на навчанні (DeepFake і NeuralTextures). Кожен метод застосовувався до 1000 високоякісних відеозаписів, завантажених з YouTube, щоб показувати зображення без перешкод і зайвих об'єктів. Усі послідовності

містили не менше 280 кадрів. Для імітації реалістичних налаштувань відеозаписи було стиснено з використанням кодека H.264. Відеозаписи високої та низької якості генерувалися з використанням параметра квантування з постійною швидкістю, рівною 23 і 40 відповідно.

DFDC – відкритий набір фальсифікованих відеозаписів, зібраний спільно компаніями Microsoft, Facebook і Amazon. Він складається із понад 119000 відеофрагментів, створених спеціально для задачі класифікації відеозаписів, згенерованих за допомогою DeepFake, що представляють як реальні, так і підробні відеоролики. Справжні відеоролики – це послідовності невеликих відео з акторами (або іншими персонажами), обраними по декількох чинниках (стать, шкіра, вік і тому подібне).

Підробні відеоролики створюються, паралельно із справжніми, застосовуючи різні методи DeepFake, наприклад, різні алгоритми зміни особи. Точні алгоритми, використовувані для створення підроблених відео не відомі.

Для створення ансамблю були обрані наступні моделі архітектури: XceptionNet (найбільш ефективна модель для виявлення маніпуляцій на зображеннях з обличчями); EfficientNetB4 (архітектура state-of-the-art у багатьох завданнях, пов'язаних з обробкою зображень); EfficientNetB4Att (архітектура, що здатна виділяти найбільш релевантні області зображень).

Кожна модель навчалася і тестувалася незалежно від інших. В наборі FF++ розглядалися тільки відеозаписи, згенеровані при квантуванні з постійною швидкістю. Модель XceptionNet навчалася з використанням одного і того ж підходу для навчання,

в той час як дві моделі EfficientNet навчалися як наскрізним, так і сіамським способами.

В результаті були досліджені чотири навчених моделі: EfficientNetB4 і EfficientNetB4Att, які навчалися за допомогою класичного наскрізного підходу, а також EfficientNetB4ST і EfficientNetB4AttST, навчені з використанням сіамської стратегії. Усі ці моделі є похідними від EfficientNetB4 та можуть сприяти остаточному об'єднанню.

Для кожного набору даних були застосовані різні способи розділення даних. Набір DFDC був розбитий по структурі папок (перші 35 папок для навчання, папки від 36 до 40 для перевірки і останні 10 папок для тестування). Для набору FF++ використовувалося таке розділення: 720 відеозаписів для навчання, 140 для перевірки і 140 для перевірки з пулу оригінальних послідовностей, отриманих з YouTube. Відповідні фальсифіковані відеозаписи присвоюються одному і тому ж розділенню.

В експериментах була розглянута обмежена кількість кадрів для кожного відеозапису. На етапі навчання цей вибір мотивується двома основними міркуваннями: при використанні дійсно невеликої кількості кадрів на відеозапису спостерігається сильна тенденція до перенавчання; збільшення кількості кадрів не покращує продуктивність виправданим чином. Це впливає з рис. 9, де представлені втрати при навчанні і перевірці залежно від ітерацій навчання, а також при виборі змінної кількості кадрів відеозапису.

Слід відзначити, що мінімальні втрати при перевірці не покращують вибір 15 кадрів на відео замість 32, проте вибір 32 кадрів на відео допомагає запобігти перенавчанню. Для тестування також треба брати до

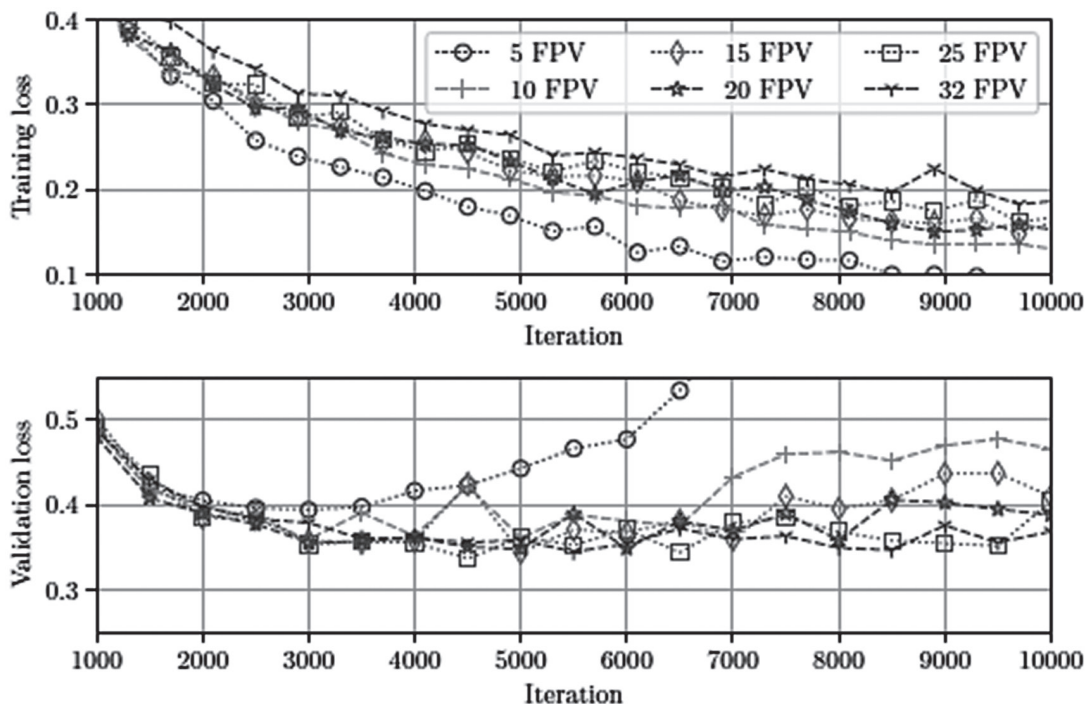


Рис. 9. Графік взаємозв'язку кількості кадрів з помилкою

уваги апаратні і часові обмеження, що накладаються завданням набору DFDC. Згідно з цим було обмежено кількість аналізованих кадрів з кожної послідовності до 32 для етапів навчання і тестування.

Обсяг даних можна скоротити, адже не вся інформація про кадри корисна для процесу глибокого виявлення. Дійсно, можна в основному сфокусувати аналіз на фрагменті, де знаходиться зображення обличчя суб'єкта. Отже, на кроці попередньої обробки витягається обличчя з кожного кадру сцени за допомогою екстрактора BlazeFace, який в експериментах виявився більш швидкий, ніж детектор MTCNN. У разі виявлення більше однієї особи зберігається обличчя з найкращою оцінкою достовірності. В результаті вхідними даними для мереж вибрано квадратне кольорове зображення розміром  $224 \times 224$  пікселі.

Під час навчання і перевірки, щоб зробити моделі надійнішими, виконувалися операції доповнення даних на вхідних гранях. Зокрема, випадковим чином застосовувалися масштабування, горизонтальне відображення, випадковий контраст яскравості, насиченість відтінку, додавання шуму і стискання JPEG. Зокрема, використовувалася бібліотека Albumentation в якості бібліотеки доповнення даних, тоді як Pytorch використовувався як фреймворк Deep Learning. Для навчання використовувався оптимізатор Адама з гіперпараметрами, що становлять  $\beta_1 = 0,9; \beta_2 = 0,999; \epsilon = 10^{-8}$ , а початкова швидкість навчання становила  $10^5$ .

Для наскрізного навчання задавався максимум ітерацій (20000) обробки пакету з 32 прикладів (16 реальних, 16 підроблених), вибраних випадковим чином і рівномірно з усіх відеозображень навчальної вибірки, або відбувалося переривання навчання при досягненні плато на функції втрат. Валідація моделі в цьому контексті виконувалася кожні 500 ітерацій навчання на 6000 прикладах, також вибраних випадковим чином і рівномірно з усіх відеозображень навчального набору даних. Початкова швидкість навчання зменшується з коефіцієнтом 0,1, якщо втрати перевірки не зменшуються після 10 процедур перевірки (5000 ітерацій), і навчання припиняється, коли досягнута мінімальна швидкість навчання.

Для сіамського навчання процедура витягання особливостей здійснювалася з тією ж кількістю

ітерацій, валідацією і швидкістю навчання, як і при наскрізному навчанні. Основна відмінність цих двох підходів полягала в застосуванні різних функцій втрати і у розмірі пакету, який для сіамського навчання складав дванадцять триплетів прикладів (шість початкових, шість підроблених і шість з наявністю як початкових так і підроблених ознак). Пакети обиралися пропорційно їх навчальній вибірці. Точне налаштування рівня класифікації виконувалося послідовно, як і для процедури наскрізного навчання з параметрами, наведеними вище.

Навчання моделей відбувалося з використанням NVIDIA 2070 RTX Super та пакету Apex, а також із застосуванням техніки Accumulation Gradient. Ці підходи дозволили тренувати глибокі моделі на цій відеокарті, проте уповільнили навчання і показали результати гірше, ніж без них.

Після навчання був проведений аналіз отриманих результатів. При застосуванні EfficientNetB4Att моделі, щоб показати ефективність Attention механізму при витяганні найбільш інформативного вмісту обличчя, була оцінена Attention карта, розрахована для декількох обличчя з набору FF++. Як вихідний шар був вибраний сигмоїд у блоці Attention, який є 2d-картою з розміром  $28 \times 28$ . Потім він був збільшений до вхідного розміру обличчя ( $224 \times 224$ ) і накладений на вхідне обличчя. Результати наведено на рис. 10.

Слід відзначити, що цей простий механізм уваги дозволяє виділити найбільш деталізовану частину обличчя, наприклад очі, рот, ніс і вуха. Навпаки, плоскі області (де градієнти малі) не є інформативними для мережі. Було показано, що артефакти методів глибокої генерації в основному локалізовані навколо рис обличчя. Наприклад, грубо змодельовані очі і зуби з надмірно білими областями, як і раніше, є основними ознаками методів підробки зображень. Щоб визначити, чи є ознаки, що створюються шарами кодування мережі при навчанні сіамським способом, дискримінаційними для завдання, було обраховано проекцію по зменшеному простору, використовуючи відомий алгоритм t-SNE [13].

Аналіз проекції, отриманої за допомогою EfficientNetB4Att, починаючи з 20 відеороликів набору даних FF++, показав, що кадри одного і того ж відеозапису об'єднуються в невеликі субрегіони, причому усі

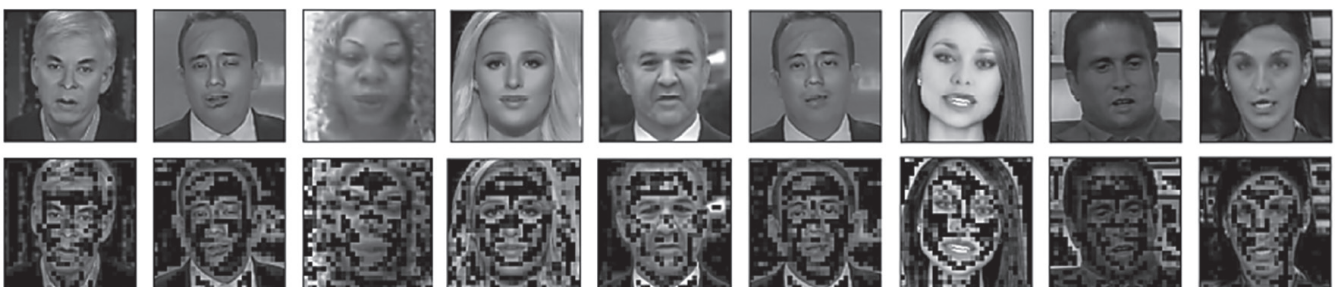


Рис. 10. Ефект застосування Attention механізму

початкові зразки згруповані у верхній області діаграми, тоді як подрібні зразки знаходяться в нижній області. Кадри з тих же відеокластерів об'єднуються в менші субрегіони. Це підтверджує доцільність використання механізму Attention на додаток до класичного наскрізного підходу. З отриманих моделей формувалися ансамблі різними способами (з використанням двох, трьох або більшої кількості моделей). Результат обчислювався як середнє значення з усіх моделей, що входять в ансамбль. В табл.2 наведені результати обчислення показників AUC і LogLoss, отримані в експериментах.

Таблиця 2

Показники точності та якості для різних ансамблів моделей

Xception Net	EfficientNet				AUC		LogLoss	
	B4	B4ST	B4Att	B4AttST	FF++	DFDC	FF++	DFDC
✓					0.9273	0.8784	0.3844	0.4897
	✓				0.9382	0.8766	0.3777	0.4819
		✓			0.9337	0.8658	0.3439	0.5075
			✓		0.9360	0.8642	0.3873	0.5133
				✓	0.9293	0.8360	0.3597	0.5507
	✓	✓			0.9413	0.8800	0.3411	0.4687
	✓		✓		0.9428	0.8785	0.3566	0.4731
	✓			✓	0.9421	0.8729	0.3370	0.4739
		✓	✓		0.9423	0.8760	0.3371	0.4770
			✓	✓	0.9393	0.8642	0.3289	0.4977
		✓		✓	0.9390	0.8625	0.3515	0.4997
	✓	✓	✓		0.9441	0.8813	0.3371	0.4640
	✓	✓		✓	0.9432	0.8769	0.3269	0.4684
	✓		✓	✓	0.9433	0.8751	0.3399	0.4717
		✓	✓	✓	0.9426	0.8719	0.3304	0.4800
	✓	✓	✓	✓	0.9444	0.8782	0.3294	0.4658

Аналізуючи отримані результати, слід відзначити, що стратегія ансамблювання моделей зазвичай обирається з точки зору характеристик. Кращі результати завжди досягаються комбінацією з двох або більше мереж, тобто об'єднання мереж допомагає збільшити як точність глибокого виявлення (оцінену за допомогою міри AUC), так і якість виявлення (оцінену за допомогою міри LogLoss).

**Висновки**

Запропонований метод сприяє виявленню маніпуляцій з обличчям у подрібних відеопослідовностях, згенерованих за допомогою глибинного навчання та технологій DeepFake. Метод використовує набір моделей EfficientNet, навчених з використанням двох основних концепцій: Attention механізму, який дозволяє нейронній мережі визначити, яка частина її вхідних даних є актуальною для виконання поставленого завдання, та стратегії сіамського навчання по сіамському триплету, яка спрямована на використання можливостей узагальнень, запропонованих нейронними мережами, для отримання дескриптора функції, який дає перевагу подібності між вибірками, що належать до одного й того ж класа. Дослідження показали, що застосування сімейства згорткових моделей EfficientNet разом з Attention механізмом, дає відмінний результат. Об'єднання декількох таких моделей

в ансамбль сприяє підвищенню якості виявлення фальсифікацій.

Перспективним продовженням досліджень є розширення ансамблю моделей архітектури EfficientNet (зокрема, за рахунок використання моделей EfficientNetB6 і EfficientNet7). Крім того, доцільно розглянути можливість застосування більш складних функцій втрат для процедур навчання нейромережевого класифікатора.

**Список літератури:**

- [1] Gonzalez R.C. Digital Image Processing / R.C. Gonzalez, R.E. Woods // 4th edition Pearson/Prentice Hall, 2018. – 1168p.
- [2] Norvig P. Artificial Intelligence: A Modern Approach / P. Norvig, S. Russell // Global Edition. – Pearson Education Limited, 2016. – 1152p.
- [3] Güera D. Deepfake Video Detection Using Recurrent Neural Networks / D. Güera, E. Delp // Proc. of 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zealand. – 2018, P. 1-6.
- [4] McCann M. Convolutional neural networks for inverse problems in imaging / M. McCann, K. Jin., M Unser //: A review //IEEE Signal Processing Magazine. – 2017. – V. 34. – №. 6. – P. 85-95.
- [5] Sikorskiy O. Convolutional neural networks in image classification / O. Sikorskiy // Information Innovative Technologies – 2017. – №1. – P. 397-401.
- [6] Schmidhuber J. Deep learning in neural networks: An overview / J. Schmidhuber // Neural networks. – 2015. – V. 61. – P. 85-117.
- [7] Goodfellow Ian J. Generative Adversarial Networks / Ian J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio // Proc. of 27th International Conference on Neural Information Processing Systems. – 2014. – P. 2672-2680.
- [8] Bezzonov O. Processing of noisy digital images with use of evolving autoencoders / O. Bezzonov, O. Rudenko, S. Udoenko, O. Dudinova // Eastern-European Journal of Enterprise Technologies. – V. 6/9(90). – 2017. – P. 63-69.
- [9] Nirkin Y. FSGAN: Subject Agnostic Face Swapping and Reenactment / Y. Nirkin, Y. Keller, T. Hassner // . – arXiv:1908.05932v1 – 2014. – P. 2672-2680.
- [10] Afchar D. MesoNet:: a Compact Facial Video Forgery Detection Network / D. Afchar, V. Nozick, J. Yamagishi, I. Echizen // eprint arXiv:1809.00888. – 2019. – 8 p.
- [11] Tan M. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks / M. Tan, Q. V. Le // – 2019. – Arxiv link: <https://arxiv.org/abs/1905.11946>
- [12] Dolhansky B. et al. The deepfake detection challenge dataset / 2020. – URL <https://arxiv.org/abs/2006.07397>
- [13] Schubert E. Intrinsic t-Stochastic Neighbor Embedding for Visualization and Outlier Detection / E. Schubert, M. Gertz // Proc. of 10th International Conference on Similarity Search and Applications. – 2017. – P. 188–203.

Надійшла до редакції 28.10.2020

О.Б. Дудінова<sup>1</sup>, С.Г. Удовенко<sup>2</sup>, Л.Е. Чала<sup>1</sup><sup>1</sup>ХНУРЕ, м. Харків, Україна, olga.dudinova@nure.ua, ORCID: 0000-0003-2709-9752<sup>1</sup>ХНУРЕ, м. Харків, Україна, larysa.chala@nure.ua, ORCID iD: 0000-0002-9890-4790<sup>2</sup>ХНЕУ ім. С. Кузнеця, м. Харків, Україна, serhiy.udovenko@hneu.net, ORCID iD: 0000-0001-5945-8647

## ІНТЕЛЕКТУАЛЬНА ОБРОБКА ПРОСТОРОВИХ ДАНИХ В ГІС ЛАНДШАФТНО-ЕКОЛОГІЧНОГО МОНІТОРИНГУ

Пропонується підхід до створення модульних підсистем інтелектуальної обробки і стиснення просторових даних в складі ГІС ландшафтно-екологічного моніторингу. Визначено функції та методи реалізації завдань цих підсистем. До основних модулів належать: модуль попередньої обробки просторових даних з формуванням оцифрованих зображень; модуль сегментації зображень та виділення контурів; модуль категорійної класифікації зображень ландшафтних об'єктів; модуль стиснення зображень з використанням фрактальної моделі та генетичного алгоритму; модуль стиснення та відновлення зашумлених оцифрованих зображень з використанням шумоподавляючого автоенкодера.

ЕКОЛОГІЧНИЙ МОНІТОРИНГ, НЕЙРОМЕРЕЖЕВА ОБРОБКА ЗОБРАЖЕНЬ, СЕГМЕНТАЦІЯ, ГЕНЕТИЧНА ОПТИМІЗАЦІЯ, АВТОЕНКОДЕР

Дудінова О.Б., Удовенко С.Г., Чала Л.Э. **Интеллектуальная обработка пространственных данных в ГИС ландшафтно-экологического мониторинга.** Предлагается подход к созданию модульных подсистем интеллектуальной обработки и сжатия пространственных данных в составе ГИС ландшафтно-экологического мониторинга. Определены функции и методы реализации задач этих подсистем. К основным модулям относятся: модуль предварительной обработки пространственных данных с формированием оцифрованных изображений; модуль сегментации изображений и выделения контуров; модуль категорийной классификации изображений ландшафтных объектов; модуль сжатия изображений с использованием фрактальной модели и генетического алгоритма; модуль сжатия и восстановления зашумленных оцифрованных изображений с использованием шумоподавляющего автоенкодера.

ЭКОЛОГИЧЕСКИЙ МОНИТОРИНГ, НЕЙРОСЕТЕВАЯ ОБРАБОТКА ИЗОБРАЖЕНИЙ, СЕГМЕНТАЦИЯ, ГЕНЕТИЧЕСКАЯ ОПТИМИЗАЦИЯ, АВТОЭНКОДЕР

Dudinova O.B., Udovenko S.G., Chala L.E. **Intellectual processing of spatial data in the GIS of landscape-ecological monitoring.** An approach to the creation of modular subsystems for intelligent processing and compression of spatial data as a part of GIS landscape-ecological monitoring is proposed. The functions and methods of implementing the tasks of these subsystems are determined. The main modules include: a module for preliminary processing of spatial data with the formation of digitized images; module of image segmentation and edge highlighting; module for categorical classification of images of landscape objects; image compression module using a fractal model and a genetic algorithm; a module for compressing and restoring noisy digitized images using a noise-canceling autoencoder.

ECOLOGICAL MONITORING, NEURAL NETWORK IMAGE PROCESSING, SEGMENTATION, GENETIC OPTIMIZATION, AUTOENCODER

### Вступ

На сьогодні екологічний моніторинг навколишнього середовища і природних ресурсів здійснюється за допомогою спеціалізованих систем спостережень і подальшої обробки отриманих даних. Комп'ютерна обробка реалістичних картографічних зображень ландшафтних об'єктів в геоінформаційних системах (ГІС) екологічного моніторингу передбачає зазвичай реалізацію етапів попередньої обробки, сегментації, розпізнавання та інтерпретації [1,2]. Особливий практичний інтерес представляють завдання розпізнавання об'єктів за супутниковими знімками і аерознімками. До основних труднощів такого розпізнавання відноситься зміна видимості об'єктів, що викликається різними внутрішніми та зовнішніми чинниками (освітлення, орієнтація, стан, наявність спотворених фрагментів зображення тощо). Аналіз і інтерпретація знімків є важливою частиною реалізації багатьох ГІС додатків (наприклад, побудова

топографічних карт, кадастрових планів, локалізації районів забруднення, моніторинг зміни контурів окремих ділянок зображень під впливом природних і антропогенних факторів). Завданням обробки просторових даних (ПД) в ГІС екологічного моніторингу природних ресурсів (зокрема, ландшафтних об'єктів) може бути як поліпшення (відновлення) зображення за деяким критерієм, так і спеціальне перетворення зображень. В останньому випадку обробка зображень може бути проміжним етапом для подальшого розпізнавання зображення (наприклад, для виділення контуру об'єкта) [3]. Оброблювані зображення, як правило, спотворені шумами та викривленнями різної природи. Функціями пониження шумів в задачах обробки просторових даних в ГІС можуть бути як поліпшення візуального сприйняття аналізованих зображень, так і збільшення чіткості при виділенні контурів об'єктів, що є необхідним для їх подальшого розпізнавання. Одним з можливих варіантів

реалізації завдань обробки зашумлених просторових даних є використання нейромережових методів, що дозволяють реалізувати паралельні схеми обробки зображень за допомогою штучних нейронних мереж (ШНМ) [4].

Втім слід відзначити, що на сьогодні не існує універсальних методів і технологій інтелектуальної обробки даних в геоінформаційних системах екологічного моніторингу регіонального та загальнодержавного рівнів. Таким чином, розробка і дослідження сукупності методів, моделей і алгоритмів обробки цифрових зображень в ГІС екологічного моніторингу, заснованих на застосуванні сучасних підходів, об'єднаних в інформаційну технологію інтелектуальної обробки цифрових зображень в ГІС екологічного моніторингу ландшафтних об'єктів (інакше, ГІСЛЕМ – ГІС ландшафтного екологічного моніторингу) та промислових зон, є актуальною науково-практичною задачею, яка визначила напрямок досліджень даної роботи.

У даній роботі пропонується підхід до створення модульних підсистем інтелектуальної обробки та стиснення просторових даних в складі ГІСЛЕМ. Такий підхід сприяє більш ефективному вирішенню низки задач, пов'язаних з екологічним моніторингом ландшафтних об'єктів (зокрема, з поточним контролем зміну їх стану).

### **1. Технології обробки просторових даних в ГІС екологічного моніторингу**

Розвиток систем екологічного моніторингу вимагає використання сучасних технологій для вирішення задач різного масштабу, пов'язаних з охороною довкілля на рівнях від локального до загальнонаціонального.

Геоінформаційні технології використовуються для вирішення багатьох екологічних проблем, зокрема, дослідження забруднення компонентів ландшафтів, динаміки їхніх змін у районах інтенсивного антропогенного навантаження тощо. Такі технології призначені для збору, зберігання, перетворення і подання інформації про стан та динаміку компонентів навколишнього середовища. Дистанційні методи дослідження дозволяють виявляти та картографувати зміни ландшафтного природного середовища внаслідок антропогенної діяльності.

Розвиток ГІС, систем GPS (Global Positioning System, глобальна система позиціонування), цифрової фотограмметрії та цифрових методів дистанційного зондування Землі (ДЗЗ) зумовив становлення наскрізних інформаційних технологій збору та обробки геопросторових даних, спричинив трансформацію геоінформаційних методів у самому картографуванні [1, 2]. Первинною продукцією інформаційних технологій є бази геопросторових даних, моделі об'єктів

у яких не зазнають картографічних “спотворень”, оскільки вони не зазнають ні генералізації, ні змін складу та роздільної здатності в контексті певного масштабу карти. Об'єкти в таких моделях відображаються з точністю й роздільною здатністю геодезичних вимірів та застосовуваних технологій збору первинних даних. Цифрові картографічні моделі, як і моделі інших геозображень, а також самі карти, перетворюються в похідну (від баз геопросторових даних) продукцію.

З розвитком ГІС і накопиченням в них великих об'ємів картографічних та предметних даних виникає необхідність в обміні інформацією між системами, які створювались на різних ГІС-платформах. Традиційні ГІС мали гібридну архітектуру за ознакою уніфікації обробки картографічних і атрибутних даних [5].

На сьогодні в арсеналі ГІС-засобів з'явилися перші інструментальні рішення для побудови відкритих геоінформаційних систем (OpenGIS), які забезпечують динамічну інтеграцію даних з різних джерел та підтримку обробки просторових геоданих з використанням технології мережі Internet.

Сьогодні компоненти відкритих ГІС є в арсеналі всіх провідних розробників ГІС-технологій. Вони розраховані на платформу Windows з використанням її основних механізмів інтеграції застосувань: об'єктних моделей (COM, DCOM, CORBA), методів інтеграції (OLE і OLE4D&M) і розробки (OLE Automation), методів доступу до баз даних (ODBC), технології візуалізації (OpenGL, GDI), та доступу до ресурсів Internet (Internet Services).

Використання геоінформаційних систем ГІСЛЕМ, що дозволяють проводити одночасний аналіз багатовимірних даних з застосуванням цифрових карт, спрощує процедури екологічного прогнозу та оцінку комплексного впливу на природне середовище, робить можливим оперативне виявлення аномалій і прийняття необхідних заходів для їх усунення.

Комп'ютерна обробка реалістичних картографічних зображень в ГІСЛЕМ передбачає в загальному випадку реалізацію етапів попередньої обробки, сегментації, розпізнавання і інтерпретації. Особливий практичний інтерес представляють завдання розпізнавання об'єктів по аерознімкам та знімкам ДЗЗ. До основних труднощів такого розпізнавання відноситься зміна видимості об'єктів, що викликається різними внутрішніми та зовнішніми чинниками (освітлення, орієнтація, стан і т.д.) [6]. У зв'язку з цим, розпізнавання ландшафтних оцифрованих зображень часто не може бути якісно проведено з безпосереднім застосуванням стандартних алгоритмів. Доцільно об'єкт представляти в категорійній шкалі з подальшим його віднесенням до однієї з категорій за інформацією, що отримується з зображення. З огляду на специфіку розпізнаваних об'єктів на аерознімках,

їх вихідне уявлення доповнюється такими факторами, як кольоровість, форма і текстурність, що дозволяє підвищити якість категоризації. Далі для кожної формованої категорії можна визначити максимум близькості між її елементами (з урахуванням всіх факторів), а потім здійснюється побудова варіанту категоризації за критерієм максимальної відстані між категоріями.

В ГІСЛЕМ доцільно реалізувати метод розпізнавання оцифрованих зображень ландшафтних об'єктів в зоні екологічного моніторингу за результатами їх попередньої категорійної класифікації.

Аналіз і інтерпретація зображень ландшафтних об'єктів є важливою частиною реалізації багатьох ГІС додатків, наприклад, побудови топографічних карт, кадастрових планів, діагностики захворювань лісових насаджень, локалізації районів забруднення тощо [7]. Відзначимо, що в загальному випадку контрастність фрагментів знімків не завжди достатня для використання стандартних алгоритмів детектування контурів. У зв'язку з цим в виділених зображеннях необхідно сегментувати фрагменти для пошуку гомогенних зон в сенсі заданих критеріїв. Ця проблематика характеризує фрагменти районів по їх освітленості і текстурі. Тривимірна інформація, яка використовується для виявлення на знімках височин, будівель тощо, може оброблятися різними методами, але при цьому в загальному випадку вирішуються завдання: обмеження зони зображення (сегментація) та ідентифікація об'єкта (розпізнавання). В даний час відсутній універсальний підхід до вирішення цієї проблеми в рамках створення ГІС-додатків. Завданням обробки просторових даних в ГІСЛЕМ може бути поліпшення (відновлення) зображення за деяким критерієм. Обробка зображень може бути проміжним етапом для подальшого розпізнавання зображення (наприклад, для виділення контуру об'єкта). Методи обробки зображення можуть суттєво відрізнятись в залежності від того, яким шляхом зображення було отримано. Виділення меж (контурів) займає особливе місце в переліку завдань обробки зображень в ГІС, пов'язаних з подальшим пошуком і виділенням об'єктів. Воно ґрунтується на алгоритмах, які визначають точки цифрового зображення, в яких різко змінюється яскравість або присутні інші види неоднорідностей. Алгоритми виділення меж і прив'язки їх до суттєвих елементів зображення називають детекторами меж або детекторами країв (edge detector) [8]. Результатом виділення меж є набір пов'язаних кривих, що позначають межі об'єктів, граней і відбитків на поверхні, а також кривих, які відображають зміни положення поверхонь.

Таким чином, застосування фільтра виділення меж до зображення може істотно зменшити кількість оброблюваних даних, так як відфільтрована частина

зображення вважається менш значущою, а найбільш важливі структурні властивості зображення зберігаються. Детектори контурів використовують відносно локальні фрагменти зображення, на підставі аналізу яких робиться висновок про наявність крайових точок – точок зображення, в яких яскравість змінюється особливо сильно. Сукупність крайових точок формує краї областей зображення.

Зазвичай детектори кордонів дають задовільний результат при наявності різких локальних змін яскравості. Однак в умовах наявності шумів і перешкод істотно спотворюється інформація про стан крайових точок, що призводить до появи двох типів можливих помилок. Перший тип пов'язаний з пропуском реально існуючої крайової точки, а другий – з появою помилковою. Поліпшення якості функціонування детекторів кордонів вимагає збільшення площі фрагмента зображення, використовуваного для прийняття рішення про наявність крайової точки, а також складних алгоритмів попередньої обробки.

З точки зору реалізації це означає необхідність виконання значного обсягу обчислень. Одним з можливих варіантів реалізації завдань обробки зашумлених просторових даних, зокрема, завадостійкого детектування меж, є застосування нейромережевих методів, що дозволяють використовувати паралельні схеми попиксельної обробки зображень за допомогою штучних нейронних мереж (ШНМ).

До найбільш поширених методів фільтрації шумів відносяться згладжуючі фільтри, фільтри Вінера, медіанні фільтри і ранжуючі фільтри.

В наступному розділі даної роботи запропоновано структуру та описано функції модульної підсистеми інтелектуальної обробки просторових даних в складі ГІСЛЕМ, що дозволяє підвищити якість формування ландшафтних цифрових зображень для подальшого аналізу стану зон моніторингу.

## 2. Структура та функції модульної підсистеми інтелектуальної обробки просторових даних

На рис. 1 наведено варіант загальної структури ГІСЛЕМ, у складі якої виділено підсистему обробки просторових даних (підсистема «Обробка ПД ГІСЛЕМ»).



Рис. 1. Загальна структура ГІСЛЕМ

До складу підсистеми «Обробка ПД ГІСЛЕМ» (згідно з її функціоналім призначенням) доцільно ввести такі модулі: модуль попередньої обробки ПД з формуванням початкових оцифрованих зображень; модуль сегментації зображень та виділення контурів; модуль категорійної класифікації оцифрованих зображень ландшафтних об'єктів.

Розглянемо функції кожного з цих модулів та методи їх реалізації.

Модуль попередньої обробки ПД з формуванням початкових оцифрованих зображень передбачає необхідність покращання їх якості перед подальшими операціями підсистеми «Обробка ПД ГІСЛЕМ».

Крім стандартних фільтрів пониження шумів (зокрема згладжуючих, медіанних та ранжуючих фільтрів) в модулі пропонується використання (за необхідністю) таких операцій: додаткової бінаризації та масштабування оброблюваних зображень з нерівномірним фоном; підвищення якості початкових зображень з використанням модифікованого методу гамма-корекції; підвищення якості початкових зображень з використанням локально-адаптивного нейромережевого підходу.

Для можливості сегментації зображень об'єктів з малою і великою кількістю пікселів одним методом доцільно модифікувати зображення виділеної області шляхом його масштабування з заданим коефіцієнтом  $\mu_{sc}$ . Операція бінаризації дозволяє уникнути труднощів, пов'язаних з викривленням зображення в процесі масштабування. Проведення масштабування змінює розмір зображення сегментованого об'єкта. При операції збільшення розміру зображення результуюче зображення формується шляхом копіювання  $\mu_{sc}$  раз кожного пікселя вхідного зображення (рис. 2, а). При операції зменшення розміру зображення кожні  $\mu_{sc}$  пікселів вихідного зображення формують єдиний піксель в результуючому зображенні (рис. 2, б). При цьому, якщо значення хоча б одного з пікселів вхідного зображення було рівним одиниці, значення результуючого пікселя буде теж дорівнювати одиниці.

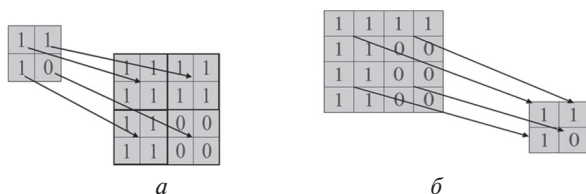


Рис. 2. Операції збільшення (а) і зменшення (б) розміру бінарного зображення при масштабуванні

В модулі передбачено можливість підвищення якості вхідних зображень для подальшого використання в ГІСЛЕМ. При цьому вирішується завдання обробки кольорових картографічних зображень на основі модифікованого методу гамма-корекції, який дозволив би збільшити детальність і якість вхідного зображення. Гамма-корекція зображення

виконується з метою збільшення або зменшення його детальності. Зазвичай, для здійснення гамма-корекції використовується ступенева функція виду:

$$I_{out} = c \cdot I_{in}^{\gamma}, \quad (1)$$

де  $I_{in}$  – інтенсивність вихідного зображення;  $I_{out}$  – інтенсивність перетвореного зображення;  $c, \gamma$  – позитивні константи.

Неправильне використання гамма-корекції може призвести до зменшення числа півтонів і переважання областей одного тону. В ході фотометричної обробки знімків виникає необхідність різного роду їх перетворень (наприклад, при зменшенні або збільшенні зображень в процесі їх трансформації). Основним завданням обробки кольорового картографічного зображення корекція його яскравості при незмінності колірних тону. Тому доцільно перетворити початкове RGB зображення в колірну модель HSI, а після поліпшення якості зображення шляхом зміни компоненти  $I$  здійснити зворотний перехід зображення в систему RGB. Запропонований в [9]. модифікований метод гамма-корекції дозволяє збільшити детальності вихідного зображення і визначити значення  $\gamma$  на основі аналізу гістограми інтенсивності. Метод може бути ефективно використаний для попередньої комп'ютерної обробки зображень в ГІСЕМ, яка є попереднім етапом подальшої нейромережевої обробки просторових даних з метою класифікації об'єктів моніторингу.

Процедура підвищення якості початкових оцифрованих зображень з використанням локально-адаптивного нейромережевого підходу [10].

Особливість цього підходу полягає в незалежній обробці кожного пікселя зображення, виходячи з наявного набору його локальних і глобальних характеристик. На відміну від відомих підходів, де повнопіксельне зображення надходить для обробки на вхід ШНМ (наприклад, мережі Хопфілда і карти Кохонена), в даному способі обробки зображень ШНМ навчається попіксельній обробці. При цьому зменшуються вимоги до обсягу оперативної пам'яті, необхідної для зберігання інформації про ШНМ. Такий підхід припускає об'єднання декількох ШНМ, що обробляють по одному пікселю, в двовірний масив для розпаралелювання обчислень. ШНМ використовується для апроксимації перетворення такого вигляду:

$$L^*(x, y) = T(L(x, y), D_{(x,y)}, m_{(x,y)}), \quad (2)$$

де  $L^*(x, y)$  і  $L(x, y)$  – відповідно оброблене і вихідне значення яскравості пікселя  $(x, y)$ ;  $m_{(x,y)}$  і  $D_{(x,y)}$  – відповідно середні значення яскравості і дисперсії яскравості в локальному оточенні оброблюваного пікселя.

Для перетворення (2) ШНМ, що апроксимує функцію  $T$ , повинна мати 3 входи та 1 вихід. При обробці кольорових зображень спочатку здійснюється

перетворення зображень в напівтонові, потім проводиться їх обробка з використанням ШНМ, а після цього інформація про колір відновлюється.

Модуль сегментації зображень та виділення контурів дозволяє здійснювати (за необхідністю) такі операції: нейромережеве виділення контурів на оцифрованому зображенні, попередньо покращеному за допомогою розглянутого вище локально-адаптивного нейромережевого підходу; сегментацію та виділення растрових зображень із застосуванням марковських моделей.

Схему нейромережевого виділення контурів на оцифрованому зображенні наведено на рис. 3.



Рис. 3. Схему нейромережевого виділення контурів зображення

Початкове зображення представляється матрицею значень яскравості елементів вхідного зображення. Результатом обробки цієї інформації є матриця опису контурів оброблюваного зображення також розміром. Для формування значення кожного елемента матриці контурів використовуються  $N * N$  елементів матриці вихідного зображення (після процедури 1), які задають квадратну маску фільтра контурів. Таким чином, для прийняття рішення про належність точки до підмножини крайових точок використовується  $N * N$  прилеглих елементів. Ці елементи упорядковуються і з них формується вектор  $S$  розмірності  $N_{sen} = N * N$ , що містить значення яскравості в околі  $(i, j)$ . Розмірність вектора  $S$  на одиницю більше, ніж кількість використовуваних точок. Компонента  $N + 1$  вектора має фіксоване значення 1, що забезпечує можливість врахування постійної складової. Вектор показань сенсорів  $S$  подається на вхід нейромережевого детектора контурів (НМДК), що містить лінійну і нелінійну частини. Лінійна частина НМДК, що має  $S$  вхід, виконує скалярне множення вектора на  $(N + 1)$ -мірний вектор ваг. Нелінійна частина є кусково-лінійною активаційною функцією, описану парою  $(T_{min}, T_{min})$  і  $(T_{max}, T_{max})$ . Ця частина НМДК може мати складну структуру, що виконує послідовно кілька кусочно - лінійних перетворень із заданими параметрами. Обробка зображення одношаровим НМДК складається з  $K_y * K_x$  однотипних перетворень елементів вхідного зображення. Параметрами обробки є вектор ваглійної частини і параметри активаційної функції нелінійної частини.

Така структура фільтра передбачає можливість повного розпаралелювання.

Схему сегментації та виділення контурів растрових зображень із застосуванням марковських моделей наведено на рис. 4 [11].

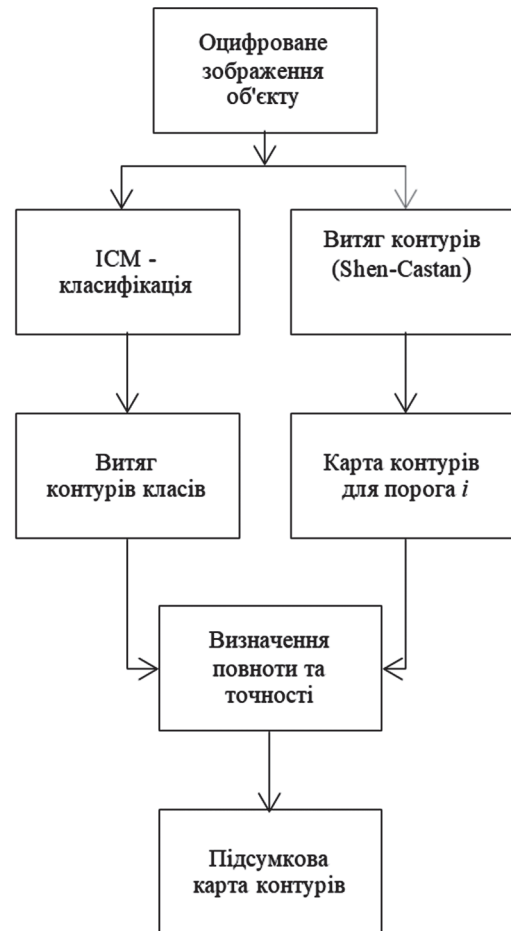


Рис. 4. Схему виділення контурів растрових зображень із застосуванням марковських моделей

Растрове зображення тут представлено як прямокутна решітка, що містить  $S$  пікселів. Кожному пікселю відповідає скалярний дескриптор  $\lambda$ , який визначає стан пікселя і приймає значення в просторі станів  $E$ . Для завдання локальних взаємодій треба структурувати просторові відносини між різними пікселями мережі. Для цього множині  $S$  ставиться у відповідність система околів  $V_s$ , які відповідає набір клік, що містять набори сусідніх точок решітки, при цьому порядок кліки визначається кількістю вхідних до неї пікселів.

На кожному етапі пропонованого алгоритму детектування контурів здійснюються такі операції:

- обчислюються локальні умовні енергії;
- визначається зображення, відповідне максимуму отриманих локальних умовних ймовірностей.

Для зображень в градаціях сірого  $(E = (0, 1, \dots, 255))$ , що формуються за допомогою алгоритму ІСМ (Iterated Conditional Mode), енергетична функція  $U(x)$  для поля Маркова з чотирьохзв'язковим околom приймає вигляд:

$$U(x) = \sum_{c=(s) \in C_1} U_c(x_s) + \sum_{c=(s,k) \in C_2} U_c(x_s, x_k), \quad (3)$$

де  $C_1, C_2$  – кліки 1-го і 2-го порядку відповідно.

Пропонований підхід до детектування контурів полягає в поданні зображення за допомогою вікон, що визначають зону інтересу. Віконний оператор обробляє пікселі вікна, щоб визначити, чи присутня в ньому ознака обраного атрибута. Цей оператор обчислює різницю між середніми значеннями з кожного боку вікна від центрального пікселя, здійснюючи експоненціальне зважування. При цьому реалізується згладжування, контрольоване контурами карти сегментації, і формується таблиця ваг на кожній ітерації, щоб визначити домінантні класи для кожного сегмента. Для вибору порога сегментації зображення контурів запропоновано використання параметрів точності і повноти. Підсумкова карта контурів відповідає найбільш прийнятному варіанту сегментації вихідного зображення. Розглянутий підхід до комп'ютерної обробки растрових зображень дозволяє враховувати характер околу аналізованого пікселя і задавати залежність між класами сусідніх пікселів.

Модуль категорійної класифікації оцифрованих зображень ландшафтних об'єктів, схему якого наведено на рис. 5, реалізує метод розпізнавання об'єктів в зоні екологічного моніторингу за результатами їх попередньої категорійної класифікації [12].



Рис. 5. Схема мультикритеріальної сегментації об'єктів з використанням категорійної класифікації

Аналіз і інтерпретація зображень ландшафтних об'єктів є важливою частиною реалізації функцій ГІСЛЕМ, наприклад, побудови топографічних карт, кадастрових планів, діагностики захворювань лісових насаджень, локалізації районів забруднення тощо.

В базовому варіанті методу розглядається інформація про колір, текстуру і форму аналізованого зображення. Будемо комбінувати ці три характеристики для моделювання об'єктів на етапі навчання, здійснюючи розпізнавання на рівні зон (а не на рівні пікселів, як при класифікації).

Зашумлений об'єкт сегментується за допомогою різних критеріїв (текстур, кольорів, кордонів) (слайд 14). Зокрема, алгоритм сегментації за кольором в розглянутому підході використовує простір п'яти

характеристик: трьох характеристик простору кольорів та двох координат пікселя в зображенні. При цьому: зображення попередньо фільтрується; локалізовані басейни атракції рекурсивно об'єднуються до збіжності; найбільш великі басейни зводяться до регіону, для якого визначається середнє значення пікселів (центр).

Відповідно до кожного критерія сегментоване зображення представляє зони входу процесу категоризації, який відбирає згідно з функцією штрафу один або кілька прототипів (моделей). Сегментація зображення за трьома критеріями (текстура, колір і детектування контурів) задає відповідні зони:  $S_1$  (оцінка текстури);  $S_2$  (оцінка кольоровості);  $S_3$  (оцінка градієнтів для контурів).

Після репараметризації процедур, щоб зберегти тільки зони великих розмірів і найбільш довгі контури (фіксується поріг для розміру і поріг для довжини), отримують остаточні результати.

### 5. Структура та функції модульної підсистеми стиснення та відновлення зображень

До складу підсистеми ГІСЛЕМ «Стиснення та відновлення зображень» (рис. 1) доцільно ввести такі модулі: модуль стиснення оцифрованих зображень з використанням фрактальної моделі та генетичного алгоритму (ГА); модуль стиснення та відновлення зашумлених оцифрованих зображень з використанням шумопригнічуючого автоенкодера (ШАЕ).

Модуль стиснення оцифрованих зображень з використанням фрактальної моделі та ГА (рис. 6) реалізує фрактальний метод стиснення відображення (метод квадродерева), де стисуючий оператор розглядається як набір тривимірних (дві координати визначають положення елементів зображення на площині і одна – їх яскравість) афінних перетворень одних областей зображення (доменних) в інші (рангові).



Рис. 6. Схема стиснення зображення з використанням фрактальної моделі та ГА

Найбільші витрати часу при цьому пов'язані з пошуком доменного блоку. У зв'язку з цим виникає завдання пошуку ефективного алгоритму пошуку мінімального елемента двовимірного масиву в просторі масивів, що формуються для фрактальної моделі зображення за допомогою квадродерева в процесі фрактального стиснення ландшафтних зображень [13].

Пошук результуючого домену відповідає мінімізації функції двох змінних (координат верхнього лівого кута домену). Ця функція є зазвичай багатаекстремальною, через що для її мінімізації доцільно використовувати генетичний алгоритм.

Як генотип ГА приймається вектор, компонентами якого є піксельні координати області доменного блоку і параметри, що визначають афінне перетворення. Існує 8 способів афінного перетворення квадрату, отже, для кодування цього перетворення достатньо трьох біт.

У знаменнику функції придатності під знаком суми задається евклідова відстань між вихідним і перетвореним блоками. Ця функція задовольняє обов'язковій для ГА умові невід'ємності і придатна для реалізації оператора рулеточної селекції. У пропонуваному алгоритмі фрактального стиснення використовується додаткова процедура, яка для класу фотореалістичних зображень може значно зменшити обсяг обчислень. Параметрами такої процедури служать рівень втрат при кодуванні і розмір областей зображень. Ця процедура забезпечує рівномірну якість кодування всього зображення.

Модуль стиснення та відновлення зашумлених оцифрованих зображень з використанням ШАЕ (рис. 7) реалізує метод нейромережевої обробки зашумлених зображень, які можуть містити викривлені фрагменти (зокрема, для обробки даних в ГІСЛЕМ). Принцип відновлення частково пошкоджених або зашумлених ландшафтних зображень за допомогою ШАЕ полягає в тому, що мережі пред'являється для навчання ряд незашумлених (чистих) зображень. В ході навчання ці зображення стискаються і їх параметри зберігаються в пам'яті автоенкодера. Після навчання мережі її можна використовувати в реальних умовах, зашумленими або частково пошкодженими [14].

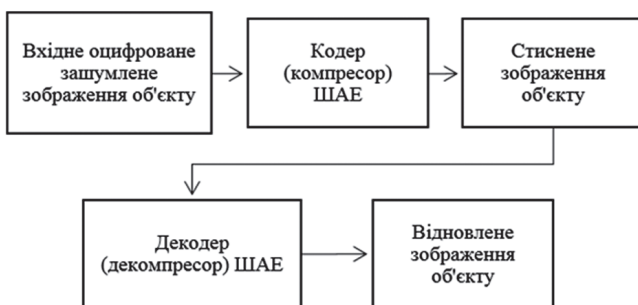


Рис. 7. Схема стиснення та відновлення зображення з використанням ШАЕ

Після пред'явлення зашумленого або частково пошкодженого зображення навченому ШАЕ в його пам'яті здійснюється пошук найбільш близького зображення до представленого, що дозволяє усунути часткові втрати інформації. Очевидно, що ШАЕ може при цьому використовувати лише зображення з набору занесених до пам'яті зразків, в зв'язку з чим виникає проблема формування навчальної вибірки.

У модулі, що розглядається, застосовано модифікацію ШАЕ, де використовується нейроеволюційний алгоритм (НА), що дозволяє не тільки налаштувати параметри мережі, а й визначити її структуру. Всі операції тут проводяться не з однією мережею, а з деякою їх множиною (популяцією). Це дозволяє виключити необхідність виконання таких складних операцій, як обернення матриць при навчанні мережі. При цьому зникає необхідність у визначенні структури мережі (кількості прихованих шарів), кількості нейронів прихованого шару і типу базисних функцій. Крім того, автоматично вирішується завдання визначення коефіцієнтів, що входять в алгоритм навчання і впливають на тривалість самого процесу навчання ШАЕ.

Довжина хромосоми в НА обмежується максимально допустимою кількістю нейронів. Кожна хромосома складається з генів, в яких зберігається інформація про відповідні параметри мережі. У старших розрядах хромосоми присутні гени, які містять інформацію про загальні параметри мережі (максимальна кількість прихованих шарів  $M$ , максимальна кількість нейронів в кожному шарі). Наступні гени кодують інформацію про усунення нейронів вихідного шару ШАЕ. Потім йдуть блоки генів, що кодують параметри відповідних нейронів прихованого шару. Після формування початкової популяції проводиться оцінка пристосованості кожної її особи. При вирішенні задачі ідентифікації зображення відомими є бажана реакція мережі  $y^*(k)$  і її реальний вихідний сигнал  $\hat{y}(k)$ . Очевидно, що мережа, для якої різниця між цими сигналами є мінімальною, може вважатися найбільш пристосованою для розв'язання задачі ідентифікації. У разі навчання off-line при наявності повної вибірки вхідних-вихідних сигналів об'єкта в модулі використовується така функція пристосованості  $i$ -ї особи:

$$f_i(x_j) = \frac{1}{M} \sum_{j=1}^M |y_j^*(x_j) - \hat{y}_j(x_j)|$$

де  $M$  – розмір вибірки.

Для визначення пристосованості мережі здійснюється її симуляція на всій вибірці та порівнюється реакція мережі з реальним вихідним сигналом об'єкта. Усереднена помилка і буде шуканим значенням функції пристосованості.

Розглянуті модулі стиснення та відновлення зображень (в тому числі зашумлених та частково викривлених) використовуються для обробки та зберігання просторових зображень в базах даних (БД) ГІСЛЕМ (рис. 1).

#### 4. Експериментальні дослідження

Експериментальні дослідження дозволили виявити переваги застосування запропонованих модулів нейромережевої обробки просторових даних в ГІСЛЕМ.

Результати, отримані в ході експериментів, свідчать про скорочення часу виконання обчислень для деяких запропонованих процедур обробки даних та підвищення якості обробки просторових зображень.

Зокрема, для тестування модуля сегментації зображень та виділення контурів використовувалися просторові дані та зображення з бази даних ГІС екологічного контролю, представлені растровою моделлю. В процесі тестування здійснювалося імітаційне моделювання процедури попередньої нейромережевої обробки зашумлених зображень і процедури нейромережевого виділення контурів на зображенні, покращеному за допомогою попередньої процедури. Середній час вибору структури і навчання ШНМ, що здійснює попередню обробку, склало близько 70 секунд (на процесорі з частотою 4 ГГц). Середній час обробки кольорового зображення розміром 512x512 пікселів склало близько 1 секунди. Параметри алгоритму, використовуваного для вибору структури ШНМ: розмір початкової популяції – 40; максимальна кількість поколінь (критерій зупинки) – 50. Для формування навчальних прикладів і були згенеровані зашумлені зображення розмірністю 100 \* 100 з накладенням шуму заданої інтенсивності. Виділення контурів здійснювалося як за допомогою синтезованого НМДК, так і за допомогою фільтрів Собеля. Було модифіковано програмне забезпечення для навчання і тестування ШНМ для обробки зображень модулем сегментації зображень та виділення контурів. Вікно інтерфейсу програми для обробки зображень з використанням навчених ШНМ представлено на рис. 8.

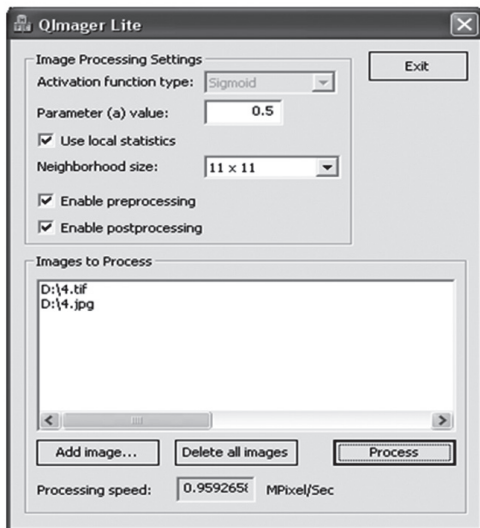


Рис. 8. Вікно інтерфейсу для обробки зображень модулем сегментації зображень та виділення контурів

### Висновки

Запропонований підхід до створення модульних підсистем інтелектуальної обробки та стиснення просторових даних в складі ГІСЛЕМ. Такий підхід сприяє більш ефективному вирішенню задач, екологічного моніторингу ландшафтних об'єктів та прогнозування зміну їх стану.

В розглянутих підсистемах використовуються такі методи та засоби обчислювального інтелекту, як шумопригнічуючі нейромережі, фрактальні моделі, методи еволюційної оптимізації. Дослідження, зокрема, показали, що застосовані в складі підсистеми обробки ПД нейроevolюційні моделі шумопригнічуючого автоенкодера характеризується поліпшеними апроксимуючими властивостями, що свідчить про можливість їх практичного використання запропонованого методу для стиснення та відновлення зачумлених зображень в ГІСЛЕМ.

Перспективним продовженням досліджень є розширення складу модулів розглянутих в роботі підсистем. Крім того, доцільно розглянути можливість застосування різних варіантів архітектури та процедур навчання нейромережевих класифікаторів, що використовуються в ГІСЛЕМ.

### Список літератури:

- [1] Зацерковний В. І. Геоінформаційні системи і бази даних: монографія. – Кн. 2 / В. І. Зацерковний, В. Г. Бурачек, О. О. Железняк, А. О. Терещенко. – Ніжин :НДУ ім. М. Гоголя, 2017. – 237 с.
- [2] Бондар О.І. та ін. Моніторинг стану навколишнього середовища засобами ГІС / О.І. Бондар та ін.. // К. : ДЕА (Державна екологічна академія), 2018. – 52 с.
- [3] Gonzalez R.C. Digital Image Processing / R.C. Gonzalez, R.E. Woods // 4th edition Pearson/Prentice Hall, 2018. – 1168p.
- [4] Kussul N. Land Cover Changes Analysis Based on Deep Machine Learning Technique / N. Kussul, N. Lavreniuk, A. Shelestov, B. Yailymov B. et al. // Journal of Automation and Information Sciences. – 2016. – Vol. 48. No. 5. – P.42-54.
- [5] Hnatushenko V.V. Satellite Monitoring of Consequences of Illegal Extraction of Amber in Ukraine / V.V. Hnatushenko, D.K. Mozgovyi, V.V. Vasyliiev, O.O. Kavats // Scientific bulletin of National Mining University. State Higher Educational Institution «National Mining University
- [6] Луньова О.В. Методологічні засади застосування ортотрансформованих космічних знімків для оцінки стану навколишнього середовища та програмне забезпечення досліджень / О.В. Луньова, О.В. Буглак // Український журнал дистанційного зондування Землі – 2019. № 22. – С.27–34
- [7] Sikorskiy O. Convolutional neural networks in image classification / O. Sikorskiy // Information Innovative Technologies – 2017. – №1. – P. 397-401.
- [8] McCann M. Convolutional neural networks for inverse problems in imaging / M. McCann, K. Jin., M Unser //: A review //IEEE Signal Processing Magazine. – 2017. – V. 34. – №. 6. – P. 85-95.
- [9] Шамраев А.А. Метод компьютерной обработки цифровых аэрофотоснимков / А.А. Шамраев, Е.О. Шамраева, О.Б. Дудинова // Системы обработки информации. – 2014. – Вып. 7(123). – С.168-171.
- [10] Дудинова О.Б. Метод нейросетевой обработки зашумленных пространственных данных / О.Б. Дудинова, С.Г. Удовенко // Автоматизированные системы управления и приборы автоматки. – 2015. – Вып.173. – С. 14 – 23.
- [11] Дудинова О.Б. Компьютерный анализ пространственных изображений с применением марковских моделей / О.Б. Дудинова, С.Г. Удовенко // Бионика интеллекта. – 2017. – Вып. 1 (88). – С. 75-79.
- [12] Дудинова О.Б. Метод категорийной классификации объектов при компьютерном анализе аэроснимков / О.Б. Дудинова // Системы обработки информации. – 2016. – Вып. 7 (144). – С. 36-44.
- [13] Дудинова О.Б. Гибридный метод сжатия растровых данных в системе экологического мониторинга / О.Б. Дудинова, С.Г. Удовенко // Бионика интеллекта. – 2016. – Вып. 1 (86). – С. 76-80.
- [14] Bezsonov O. Processing of noisy digital images with use of evolving autoencoders / O. Bezsonov, O. Rudenko, S. Udovenko, O. Dudinova // Eastern-European Journal of Enterprise Technologies. – V. 6/9(90). – 2017. – P. 63-69.

Надійшла до редколегії 05.11.2020



О.І. Повзун<sup>1</sup>, Н.Б. Козел<sup>2</sup>, О.А. Каратаєв<sup>3</sup>, Г.Г. Четвериков<sup>4</sup>

<sup>1</sup>Кандидат технічних наук, доцент, Донецький національний технічний університет,  
povzun.aleksey@gmail.com

<sup>2</sup>Старший викладач кафедри ПІ ХНУРЕ, м. Харків, Україна,  
ORCID iD: 0000-0001-9276-9877, natalia.kozell@nure.ua

<sup>3</sup>Аспірант кафедри ПІ ХНУРЕ, м. Харків, Україна, oleksandr.karataiev@nure.ua

<sup>4</sup>Доктор технічних наук, професор, ХНУРЕ, м. Харків, Україна,  
ORCID iD: 0000-0001-5293-5842, chetvergg@gmail.com

## КОНЦЕПЦІЇ ОРГАНІЗАЦІЇ ІНФОРМАЦІЙНО-ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ ТА ІНТЕЛЕКТУАЛЬНОЇ ПІДТРИМКИ СУСПІЛЬНО-ЕКОНОМІЧНИХ ПРОЦЕСІВ: k-ЗНАЧНІ ЗАСОБИ. ЧАСТИНА 2

Стаття присвячена аналізу проблеми створення систем штучного інтелекту, які дозволяють моделювати на логічному та апаратному рівнях економічні рішення описані математичними операціями над природною мовою, і які є елементами *k*-значної структурної організації інформаційно-інтелектуальних технологій. Показана необхідність і можливість розробки загальної теорії побудови інтелектуального управління і систем штучного інтелекту, яка могла б стати методологічною основою цілеспрямованого створення нових інформаційних технологій.

*k*-ЗНАЧНА СТРУКТУРА, ЛОГІКА, ПРИРОДНА МОВА, ШТУЧНИЙ ІНТЕЛЕКТ, БАЗА ЗНАНЬ

Повзун А.И., Козел Н.Б., А.А. Каратаев, Четвериков Г.Г. Концепции организации информационно-интеллектуальных технологий и интеллектуальной поддержки общественно-экономических процессов: *k*-значные средства. Часть 2. Статья посвящена анализу проблемы создания систем искусственного интеллекта, которые позволяют моделировать на логическом и аппаратном уровнях экономические решения описанные математическими операциями над естественным языком, и которые являются элементами *k*-значной структурной организации информационно-интеллектуальных технологий. Показана необходимость и возможность разработки общей теории построения интеллектуального управления и систем искусственного интеллекта, которая могла бы стать методологической основой целеустремленного создания новых информационных технологий.

*k*-ЗНАЧНАЯ СТРУКТУРА, БАЗА ЗНАНИЙ, ЛОГИКА, ЕСТЕСТВЕННЫЙ ЯЗЫК, ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

Povzun A.I., Kozel N.B., Karataiev O.A., Chetverykov G.G. Conceptions of organization of informatively-intellectual technologies and intellectual support of social-economic processes: multiple-valued tools. Part 2. The article is devoted the analysis of problem of creation of the intelligence systems, which allow to design at logical and vehicle levels economic decisions described mathematical operations above a human language, and which are the elements of multiple-valued of structural organization of informatively-intellectual technologies. A necessity and possibility of development of general theory of construction of intellectual management and intelligence systems is shown, which would become methodological basis of purposeful creation of new technologies of informations.

*k*-VALUED STRUCTURE, KNOWLEDGE BASE, LOGIC, NATURAL LANGUAGE, ARTIFICIAL INTELLIGENCE

### Вступ

На рубежі XXI століття тематика інтелектуальних систем і інтелектуального управління зазнає значних змін. Явно намітилася позитивна тенденція переходу від ігравково-модельного підходу до інтелекту, до його сприйняття як деякої характеристичної властивості систем високої організаційної складності, властивості достатньо специфічної і виразимої достатньою мірою тільки на мовах контекстно-залежного рівня. Так же корисною тенденцією можна вважати і поступове усвідомлення дослідниками, що комп'ютер фон-Нойманівської архітектури, кінцевий автомат за своєю суттю, не може бути інструментом створення інтелектуальних систем, інтелектуального управління бо є системою контекстно-незалежного рівня. Це породило новий напрям, який стверджує що у міру

просування до вищих рівнів ієрархії – підвищується інтелектуальність системи при зниженні її точності і навпаки. Зрозуміло, що такі системи призначені для роботи в умовах невизначеності (неможливості точного математичного опису) інформації про властивості і характеристики системно-складних об'єктів і середовища їх дії.

В умовах роботи реальних систем із високим рівнем невизначеності інформації для побудови інтелектуальних систем неминуче використання нових інформаційних технологій, зорієнтованих на потоки контекстно-залежної інформації, тобто фактично необхідна розробка природно-мовних принципів побудови інтелектуального управління – теорії інтелектуальних систем управління (ІСУ) – для систем вищих рівнів системної складності. Виходячи зі

казаного, для правомірного використання кінцевого автомата (комп'ютера) у складі інтелектуальної системи, теорія повинна розглядати можливість побудови абстрактних конструкцій, що реалізують не обчислювані в звичному значенні об'єкти. Все, що дотепер винайдене, всі узагальнені функціональні перетворення годяться тільки для подання рахункових сукупностей процесів, поданих потоками, хоча і нескінченними, але однорідними, що складаються з нескінченно малих невиразимих сутностей. У разі відкритих (інтелектуальних) систем ми маємо справу з незчисленною множиною потоків, кожний із яких може розкритися в більш ніж зчисленну сукупність потоків, що складаються не з безликих нескінченно малих, а з нескінченної різноманітності структур.

Можна прогнозувати і подальший розвиток інтелектуальних систем. Це можна зробити виходячи навіть всього з двох достатньо очевидних міркувань. По-перше, інтелектуальні системи повинні одержати свою власну “інформаційну машину” — машину, що подолала теоретичні обмеження кінцевих автоматів і практично нереалізованої машини Тюрінга. Тут потрібне наукове і технічне рішення, що долає бар'єр „невчислимості”. По-друге, сам факт побудови такої машини, практика спілкування з нею приведуть до остаточного переосмислення поняття і значення інформаційної науки — інформатики. Достатньо очевидний факт неможливості існування інформаційної машини такого роду інакше як у безперервному режимі засвоєння і реструктуризації інформації приведе до розуміння інтелектуальної системи як системи істотно динамічної.

Отже основні аксіоми теорії інтелектуальних систем управління (ІСУ) [1–17]:

- комп'ютер фон-Нойманівської архітектури, кінцевий автомат за своєю суттю, не може бути інструментом створення інтелектуальних систем;
- підвищується інтелектуальність системи при зниженні її точності і навпаки;
- теорія повинна розглядати можливість побудови абстрактних конструкцій, що реалі
- необхідна розробка природно-мовних принципів побудови інтелектуального управління;

### 1. Мета роботи

Основним завданням цієї роботи є виклад новостворюваної концепції організації інформаційно-інтелектуальних технологій та інтелектуальної підтримки суспільно-економічних процесів, зокрема діяльності фінансово-кредитних установ. Предметом досліджень є моделювання інтелектуальної діяльності людей як у зовнішньому її прояві (вирішення складних завдань, розуміння природної мови, інтерпретація візуальної інформації та мови), так і у внутрішньому (накопичення, надання і використання знань).

### 2. Природно-мовні принципи побудови систем штучного інтелекту

Виходячи із поставленої мети та враховуючи основні аксіоми теорії інтелектуальних систем управління інтегруємо необхідні та уже розроблені природно-мовні принципи побудови інтелектуального управління і систем штучного інтелекту. У цій роботі, перш за все, хотілося б показати необхідність і можливість розробки загальної теорії побудови інтелектуального управління і систем штучного інтелекту, яка могла б стати методологічною основою цілеспрямованого створення нових інформаційних технологій.

#### Морфологічний аналіз

Однією з перших є задача морфологічного аналізу є [1 - 5], яка полягає в ідентифікації словоформ та присвоєнні кожній словоформі комплексу морфологічної інформації (КМІ). Такий комплекс складається із морфологічно-інформаційних рядків (строк) (МІ-строк), структура яких наступна:

- номер, <основа чи ознаки основи), МІ>, де номер — порядковий номер даної словоформи у фразі;
- основа (ознака основи) — код семантичної ознаки, номер синтаксичної чи семантичної моделі керування, що присвоєні даній основі в словнику основ;
- МІ — частина мови та її граматичні категорії: рід, число, відмінок, час, особа тощо.

Існує два методи реалізації [1–5] морфологічного аналізу (МА): словниковий (декларативний) (використовується для аналізу мов із нерозвинутим відмінюванням слів (англійська, французька тощо)). Для аналізу української мови використовується алгоритмічний (процедурний) морфологічний аналіз. При МА здійснюється розчленування словоформ на основу та закінчення і в словниках зберігаються як основи, так і їх закінчення. МА здійснюється шляхом пошуку в складі словоформи, що аналізується деякої словникової основи та певного словникового закінчення. Потім здійснюється порівняння інформації про основу та закінчення і отримується комплекс морфологічної інформації для всієї словоформи.

Під час МА змінюваної словоформи її кінцеву частину за чергою порівнюють із закінченнями словника. Якщо порівняння відбулося, то ту частину словоформи, що збіглася виокремлюють і отримують припустиму основу (ПОС), припустиме закінчення (ПЗК) та припустиму морфологічну інформацію (ПМІ).

Дані про ПЗК (ПМІ) зчитують із словника закінчень (морфологічної інформації). Потім переходять до пошуку інших ПЗК, ПОС та ПМІ.

На другому кроці аналізу словоформи виконується ідентифікація її можливих основ шляхом перевірки збіжності отриманих припустимих основ із вмістом машинного словника основ.

На третьому кроці МА словоформи зрівнюється інформація з тими припустимими основами та ПЗК, що отримали підтвердження з допомогою словника основ.

Ефективність МА суттєво залежить від виду подання машинних словників у пам'яті ЕОМ та способу їх кодування. При цьому доцільно мати окремий допоміжний словник перенумерованих основ, що наявні у одному примірнику та розташовані в алфавітному порядку.

Для подання значень граматичних категорій будь-якої словоформи використаємо 9-ти розрядний 10-значний код. Порозрядно у  $p(1)$ ,  $p(2)$  — закодовано частина мови словоформи,  $p(3)$  — тип та клас прийменника чи одушевленості (іменника, повного прикметника);  $p(4.i)$  — дієслово 1 - 3 особи відповідно;  $p(5)$  — код значення числа (однина, множина);  $p(6)$  — код відмінка (іменний, родовий, давальний ...);  $p(7)$  — код категорії пасивності-активності;  $p(8)$  — код часу (теперішній, минулий, майбутній);  $p(9)$  — код категорії виду (доконаний, недоконаний) закінчення [1].

Для формування однієї МІ-строки до всієї словоформи порівнюють код основи та код закінчення на відповідність їх перших п'яти розрядах, якщо збіжності немає, то дані несумісні. Для порівняння вибирається черговий код закінчення. Якщо відповідність встановлена, то решта розрядів результуючого коду формуються за правилами 10-значної диз'юнкції значень відповідних розрядів кодів основи та закінчення. При цьому попередньо перевіряється умова збіжності операндів чи рівність одного з них нулеві.

Таким чином, описаний алгоритм дозволяє інтерпретувати різноманіття граматичного оброблення українських флексій (аналіз, синтез, нормалізація, корегування помилок тощо) із допомогою розв'язків канонічних рівнянь виду  $L_{\phi}(X, Y) = 1$ .

### Синтаксичний та семантичний аналіз

Тепер ми переходимо до розгляду необхідних відомостей про контекстно-залежні (КЗ) мови, тобто про реалізацію інтелектуального управління, коли нас цікавить можливість мінімізації яких би то не було втрат при використанні скінченого автомата як несучої основи КЗ-мови.

КЗ-мова (природна мова людини) в галузі наукової термінології володіє великою невизначеністю, що пояснюється частково поліморфізмом і контекстно-залежним поданням наукової інформації, а іноді (і для інформатики і для інтелектуального управління це особливо важливо) — недбалістю використання термінів.

*Семантика* визначає відношення між знаками і їх концептами, тобто задає зміст чи значення конкретних знаків.

Семантика, як апарат забезпечення “поточного розвитку думки” — контекстно-залежного

спілкування, що забезпечує спостережуваність складної системи.

Слова в мові не йдуть у довільному порядку і закони їх упорядкування є предметом синтаксису. Синтаксис описує структуру можливих фраз. Опис синтаксичних структур використовує наступні граматики (формалізми) [1, 2]:

- дерева синтаксичного підпорядкування;
- системи складових;
- розширені мережі переходів.

Таким чином, синтаксичний аналіз використовує заготовлені з допомогою граматики шаблони до вхідних фраз із метою виявлення (встановлення) відповідності між послідовністю, що аналізується та значущими синтаксичними структурами. Основним формальним засобом математичного опису природної мови, у рамках синтаксичного та семантичного аналізу, є алгебра скінчених предикатів (АСП), оскільки мова є скінченою, дискретною та  $k$ -значною. АСП у процесі її дії використовує процедури збіжні з розв'язуванням рівнянь, а не алгоритмів та процедур із однозначним результатом. У роботі [1] в процесі синтаксичного розбору природномовних висловлень розроблено метод побудови синтаксичних дерев для аналізу простих речень. Для встановлення інтегральних закономірностей обробки природної мови проаналізуємо, що власне відбувається у процесі аналітичних досліджень вищих лінгвістичних механізмів дії російської мови, проведених у свій час в науковій школі Харківського інституту радіоелектроніки

### Семантика

Вихідним матеріалом для семантичного аналізу природної мови є синтаксична структура фрази чи її фрагмента, а також дані про значення словоформ. Основна задача семантичного аналізу — це зняття неоднозначності, морфологічної та лексичної багатозначності словоформ та синтаксичних структур речень.

У роботі [8] об'єктом математичного моделювання є словосполучення, що мають інструментальне значення: Для аналізу семантики повідомлення на природній мові необхідно визначити *значення одиниць повідомлення*. Значення слів класифікують згідно з набором апріорних ознак: дія — інструмент дії, або іншими словами — дієслово (конкретної дії чи акційне) — іменник у певному відмінку (називний, родовий, давальний тощо).

Метод, що покладений в основу — метод семантичного аналізу. Для аналізу семантики повідомлення на природній мові необхідно визначити значення одиниць повідомлення. Значення слів класифікують згідно з набором апріорних ознак: дія — інструмент дії, або іншими словами — дієслово (конкретної дії чи акційне) — іменник у певному відмінку (називний, родовий, давальний тощо). Для дослідження

семантики словосполучень такого виду використовують семантичні мережі та зв'язаний з ними математичний апарат — у даному випадку для кожного словосполучення у вигляді двох графів. У якості формального апарату подання семантики використовують АСП.

Якщо побудова результуючого графу і відповідного йому предиката АСП можлива, то це означає, що розглянута комбінація слів утворює осмислене словосполучення, а також можливо встановити чи володіє об'єкт деякою властивістю; визначити якими властивостями повинен володіти інструмент для dokonання дії та відновити іменник чи дієслово за набором ознак тощо.

Наступною фазою досліджень стала робота [10] про змістовну інтерпретацію алгебри ідей. Тут об'єктом математичного моделювання стали: смислова однозначність; ситуаційно-предикатна; ситуаційно-множинна; ситуаційно-кодова ідея.

Роботах [6-9] об'єктом математичного моделювання обрана семантика похідних слів із модифікаційними значеннями.

Відповідно у роботі [9] досліджено міжморфемні семантичні зв'язки, які виникають у процесі словоутворення між префіксними та кореневими морфемами, кореневими та суфіксними морфемами, а також між основами та закінченнями.

І нарешті роботи [8, 9] присвячені процедурам приписування словоформам семантичних ознак і моделюванню семантики — похідних слова (мідь-мідний, залізо-залізний) відповідно.

У всіх випадках ми маємо управління як результат “оптимізованого інформаційного пошуку”, мета якого — вироблення управляючого рішення, тобто відповідного повідомлення на основі аналізу структури даних, закладеної в машину при проектуванні інформаційної системи, і її біжучого наповнення.

“Апріорна семантика” присутня лише у “власне даних”, тобто в значеннях первинних сигналів і в “словнику”, в наборі термінів, які “стали константами, що забезпечують життєздатність системи”.

“Семантика як така”, тобто метрика і як спосіб вимірювання, і як набір масштабів і коефіцієнтів, з'являється лише в процесі інтерпретації, в процесі взаємодії потоків даних “від середовища” і “від системи”.

У випадку відкритих інтелектуальних систем управління структурно-динамічне, мета якого — формування деякої “структури знань”, змінної в часі іменованої структури зв'язків, організація “інформаційного резонансу”. Власне вироблення того чи іншого рішення є завданням найважливішої, але вторинної, похідної від основного завдання системи — “бути в курсі всіх змін і в постійній готовності до сприйняття сенсу запиту чи повідомлення за результатами попередньої інформаційної посилки”.

Далі семантика накопиченої інформації вже забезпечить у потрібний момент вироблення структури зв'язків, що може служити для перетворення в будь-які мислимі дії: організаційні, правові, судові, особові, моральні, чим, власне кажучи, і визначається інтелектуальне управління в нашій поточній інженерній постановці. Саме поточній, тому, що на її понятійній базі ми будемо розглядати інші більш довершені підходи до роботи з інформацією.

“Семантика динаміки”, семантика “другого порядку” вже дає спостережуваність системи, недосягну ніяким іншим способом, але семантика “першого порядку”, все, що може бути відоме не про процес вироблення, а про сам продукт також повністю наявна в комп'ютері з достовірністю в точності рівній документу, супроводжуючому продукт, а більшого досягти не можна.

Складається враження, що семантика — довільна вигадка теоретиків, а в природі її і немає зовсім.

Насправді семантика нікуди не пропала, просто адекватно реалізований апарат “здвоєної W-граматики [10-11] акуратно і послідовно “розрізає” її на дві частини — “константну”, яку вона вкладає в БД у вигляді літералів і зв'язків і іменує після цього ієрогліфом словника, і “плинну”, змінну, яка “залишається в розпорядженні” ПЗ і користувача. Велика частка народів Землі успішно чинить так само — користується ієрогліфами, до яких ми повинні відносити і всю термінологію професійних сленгів. Звідси випливають два висновки.

Теоретичний — семантика по суті динамічний об'єкт зі всіма витікаючими наслідками.

Практичний — чи варто використовувати в реалізації програмних продуктів “функції семантичних оцінок” тощо, бо це не більше ніж часткова статистика. Саме часткова, зроблена конкретно і на конкретному матеріалі і що має дуже опосередковане відношення до всіх решти випадків. Користь від неї сумнівна, зате неприємності — гарантовані, у чому ми пересвідчилися на прикладах робіт.

Намагаючись досягти “чистої абстракції” ми розриваємо деякі взаємодії, можливо “знищуємо” процеси, які відповідальні за виникнення та існування досліджуваних феноменів. Тим самим цілком імовірно знищується сама можливість рішення початкової проблеми, відбувається її неусвідомлена підміна на довгі роки, до тих пір, поки знов не проявлять себе факти, що не вклалися в поточний варіант “чистої абстракції”.

Другий варіант — варіант відмови від проблеми або, що те ж саме — введення революційних перебудов, фактично є логічним доповненням першого, принаймні до тих пір, поки ми орієнтуємося на будь-які раціональні чи ірраціональні методи *винайдення* аксіоматики.

Дійсно, чом би не припустити, що істина навіть не посередині, а в нерозривному зв'язку вказаних позицій? А ми повторимо тут наші висновки про необхідність постійної зміни правил формальної логіки, але вже з посиланням на Л. Керрола – “суть полягає в тому, що правила (гри) постійно міняються”.

Можливо, причина якраз у “правилах виникнення зміни правил” і зовсім не потрібно нескінченних сходів “правил-над-правилами” саме через динаміку системи. А нескінченні ієрархії “правил зміни правил” у багатьох дослідженнях виникають виключно від того, що розглядаються “мертві” статичні моделі чи довільно вирізані шматки систем.

Інакше кажучи, якісь із цих “правил над правилами”, об'єктивно існуючі закони відкритих систем можуть проявляти свою дію (або взагалі виникати!) тільки в деякій “мінімальній сукупності взаємодій”. Достатньо незначного спрощення “моделі”, відходу від реальної ситуації, щоби або ніколи їх не знайти, або одержати цілком реальні і несподівані наслідки їх дії. Саме ця ситуація найбільш типова для всіх модельних реалізацій інформаційних систем. Практично весь попередній матеріал є коментарем цієї проблеми і пошуком інженерних шляхів її рішення.

Взагалі кажучи, сказане в деяких аспектах вже давно всім знайоме, нікого не дивує, що в світі елементарних частинок, у квантовій механіці діють закони окремі, не схожі на макромеханіку. В світі елементарних частинок взаємодія, розпад одних структур і утворення інших відбуваються “в перебігу одного кванта часу”, тобто “усередині процесу нічого немає” (або ми поки що не уміємо уявити собі, що там є). Навпаки, весь зміст існування інформаційних систем полягає в процесі перетворення одних структур в інші.

Звернемо увагу, що при цьому для термодинаміки і статистичної механіки, тобто для подання ентропійних процесів, виявляється зручним і адекватним подання процесів у вигляді потоків однорідних нерозрізних сутностей, які можна розділити “до нескінченно малого стану”. Навпаки, сутність інформаційних систем, що самоорганізуються, полягає у взаємодії різних потоків, у нескінченній різноманітності структур, які представимі в скінченному вигляді, у вигляді скінчених (інше просто не цікаве) алфавітів, знаків складених із кінцевого числа розрізних елементів – “бітів”.

### 3. Машини баз даних і знань

Вище ми встановили формальну можливість побудови абстрактних конструкцій, що реалізують необчислювані в звичайному сенсі структури. Як наслідок, ми маємо можливість подання контекстно-залежної проблемно-орієнтованої мови механізмом обчислення предикатів на основі алгебри відношень, що змінюються.

Для “апаратної підтримки” можливостей предикативного підходу ми можемо запропонувати деякий механізм, орієнтований на практичне завдання сприйняття семантики інформації як управління з урахуванням всіх необхідних умов контекстного аналізу і реструктуризації зв'язків при роботі зі знаннями.

Такий механізм забезпечення управління для системно-складних об'єктів у теорії ІСУ природно назвати *машиною баз знань* (МБЗ), яку не треба путати з давно відомими машинами баз даних (МБД).

Як відомо, МБД появились досить давно і розглядалися як основа обчислювальних систем 5-го покоління. Підставою для цього служив відомий факт невідповідності фон Нойманівської структури комп'ютера вимогам СУБД. Зокрема, реалізація пошуку, оновлення, захист даних, обробка транзакцій тільки програмним способом неефективні як за продуктивністю, так і за вартістю.

Крім того, МБД спочатку були орієнтовані на реляційні моделі даних, що нас не може влаштувати вже хоч би тому, що реляційні і постреляційні бази не розраховані на забезпечення постійного процесу реструктуризації своїх зв'язків. Відповідно, перед цими машинами не ставилася і не може бути поставлена задача роботи з контекстно-залежними мовами.

Розглянемо деякі можливості реалізації необхідних нам машин баз знань, якщо хочете – машин 6-го покоління.

Із отриманого під час розгляду проблеми реалізованості баз знань висновку, впливає принципова можливість, але не процедура практичного здійснення реалізації роботи з контекстно-залежними описами, навіть і зведеними до вказаного вище варіанту числення предикатів із контекстно-змінними відношеннями.

Проте для цього нам треба мати достатньо гнучку структуру, на якій можна ефективно і безболісно вести цю саму реструктуризацію, що забезпечує реалізацію поведінки (прийняття рішення), що не зводиться до скінченної моделі.

Виходячи з сказаного, розглянемо наступну архітектуру МБЗ (рис. 1).

Машина складається з необхідного числа  $n$  окремих модулів, на вхід кожного з яких після перетворення вхідної текстової посилки в мову предикатів надходить трійка виду  $(ARB)_i$ ,  $i \in 1...n$  для її наступного безпосереднього розкриття за відповідними законами.

У зв'язку з тим, що число законів перетворення невелике, тут ефективно використання динамічно утвореної матриці виду, наведеного на рис. 1. Рядки цієї матриці відповідають не тільки порядку перетворення вхідних предикатів, але і складають таблицю всіх формально-логічних відношень, використаних

ARB <sub>1</sub>	Правила перетворення	ARB <sub>11</sub>	ARB <sub>12</sub>	ARB <sub>13</sub>	ARB <sub>14</sub>	ARB <sub>15</sub>	ARB <sub>16</sub>	...ARB <sub>17</sub> ....		
ARB <sub>2</sub>		ARB <sub>21</sub>	ARB <sub>22</sub>	ARB <sub>23</sub>	ARB <sub>24</sub>	ARB <sub>25</sub>	ARB <sub>26</sub>			
ARB <sub>3</sub>		ARB <sub>31</sub>	ARB <sub>32</sub>	ARB <sub>33</sub>	ARB <sub>34</sub>	ARB <sub>35</sub>	ARB <sub>36</sub>	ARB <sub>37</sub>		
ARB <sub>4</sub>		ARB <sub>41</sub>	ARB <sub>42</sub>	ARB <sub>43</sub>	ARB <sub>44</sub>					
ARB <sub>5</sub>		ARB <sub>51</sub>	ARB <sub>52</sub>	ARB <sub>53</sub>	ARB <sub>54</sub>	ARB <sub>55</sub>	ARB <sub>56</sub>	ARB <sub>57</sub>		
...		...								
ARB <sub>n</sub>		ARB <sub>n1</sub>	ARB <sub>n2</sub>	ARB <sub>n3</sub>	ARB <sub>n4</sub>	ARB <sub>n5</sub>	...			

Рис. 1. Архітектура машин баз знань (реалізація на основі мови числення предикатів)

у поточній інформаційній посилці. До кожної наступної операції прийому вхідної інформації машина переходить у стан готовності прийняти вхідні сигнали на окремі входи, для чого в процесі надходження сигналу формується чергова матриця.

Вихідним повідомленням кожного модуля є набір розширюючих вхідний текст предикатів виду  $ARB_{i,l}$ ,  $i \in 1...n$ ,  $l \in 1...k$ , що містить у собі всі можливі логічні перетворення початкового предиката. У зв'язку з тим що цей процес суттєво паралельний, можлива організація кожного окремого модуля як паралельної машини. В деякому розумінні такий модуль є не тільки скінченим автоматом, але і нейроном, що має декілька входів і множину виходів, відповідну числу нових предикатів, отриманих із вхідної посилки.

Декілька входів корисно передбачати і для отримання висновків “другого порядку” – висновків, що впливають із сукупної оцінки декількох різних вхідних предикатів. Зрозуміло, що тим самим модулі отримують повну універсальність і можуть у міру закінчення своєї роботи на першому рівні перетворення використовуватися на другому, а якщо треба – то і наступних рівнях.

#### 4. Паралельні обчислення з управлінням від потоку даних

Порівняємо можливості машини фон-Нойманівської архітектури і МБЗ стосовно предикативного варіанту механізму обробки контекстно-залежної мови.

У першій ми маємо класичний варіант архітектури складових частин, орієнтований на програмно-алгоритмічну організацію процесу роботи з даними. Це істотно послідовна машина з відомими проблемами налагодження і скінченності обчислювального процесу. Останнє пов'язане з тим, що проблеми оцінки його рекурентності або рекурсивності вносять суттєві обмеження на застосовність такої машини для задач управління, що розглядаються.

У МБЗ вхідна інформація є і управлінням і, одночасно, командою для організації дій щодо її обробки. Це досить природно, бо йдеться про вхідну мову контекстно-залежного рівня. При бажанні говорити в термінології алгоритмів тут можна сказати, що алгоритм роботи явно заданий в оброблюваному контекстно-залежному записі.

Проте пам'ятатимемо: вхідна інформація може суттєво змінити структуру скінченого автомата і

затребувати всіх дій, необхідність яких показана при розгляді проблеми стійкості. Проте, тут можливе введення логічного критерію зупинки (наприклад, за глибиною перетворення предикатів) і тому процедура обробки запису скінчена. МБЗ суттєво паралельна машина зі всіма витікаючими із цього корисними можливостями для обробки даних і зв'язків між ними.

Відзначимо, що МБЗ яка зовнішньо представляється скінченим автоматом, у кожний конкретний момент часу змінює свою структуру під впливом вхідного потоку інформації, реалізуючи тим самим вказаний вище випадок моделювання в частотному сенсі нерекурсивних об'єктів при ситуації “незводимості до скінченої моделі, але скінченої модельованості в частотному сенсі”.

Ця досить проста за змістом теза містить у собі насправді цілий спектр фундаментальних положень. По суті це неявно висловлене положення про існування “інженерного” чи “механістичного” інтелекту, деякого “псевдорозуму”, здатного не тільки навчатися, але і “навчатися навчанню”. Насправді, поки що, саме цього ми і хочемо, ставлячи завдання створення МБЗ, а решта властивостей інтелекту, наприклад, свободу волі вважаємо “зайвою” для такої системи.

Дійсно, якось незатишно опинитися в залежності від деякої машини, яка сильно “полегшує нам життя”, але в обмін на це ми будемо змушені зважати на те, що вона має “своє власне трактування” одержаних нею даних, “своє розуміння предмета”. Інакше кажучи, створюючи МБЗ ми бажаємо створити щось, здібне до розуміння, але хочемо апріорі позбавити себе від обов'язку взаєморозуміння.

Розглянемо чисто інженерні аспекти такого підходу, сформулюємо в явному вигляді, що ж ми хочемо від “інженерного” інтелекту в цілому і від його компонентів. У першу чергу потрібно звернути увагу на неявні формулювання дуже сильних вимог, по суті тих, що визначають саму можливість постановки задачі створення МБЗ.

Формулювання “управління від потоку даних” насправді передбачає, що “потік даних” якраз і містить у собі деякі структури, “виловлення” і правильне використання яких дозволить отримати від МБЗ здатність до “самоструктуризації”, самоорганізації.

Тобто передбачається фундаментальна відмінність феномена, званого “потік даних”, від потоку

в класичному розумінні поточкових систем. У останніх потік припускається за умовчанням деякою безструктурною сутністю, структури ж виникають із взаємодії потоку з межами, поверхнями розділу різних середовищ. Так, наприклад, потік у задачах теплопровідності – щось апріорі, за визначенням безструктурне, “потік даних” – навпаки, підрозуміває наявність деяких прихованих структур, але апріорі невідомих, які і належить ідентифікувати і використовувати.

Не можна обійти увагою і сугубу довільність терміну “самоструктуризація”, що все частіше використовується в роботах, пов’язаних із проектуванням інформаційних баз, тобто, зрештою, і з МБЗ. Резонно виникає питання “самоструктуризація до чого”? До якої моделі чи структури? Деяким дослідникам вистачає мужності визнати об’єктивність існування деякої “природної структури проблемної галузі”, проте це поки що не більше, ніж блага побажання пошуку такої. Правда, слід зазначити, побажання далеке не безпідставне.

Дійсно, наприклад, масові хімічні формули частково об’єктивно відображають властивості хімічних сполук, структурні формули адекватні топології (але не геометрії) розташування атомів у молекулах, структура механіки Ньютона частково адекватна всій структурі сучасної фізики тощо. Є достатні підстави передбачати існування спільних законів взаємодії структур, що охоплюють не лише фізичні (енергетичні), але й інформаційні явища і взагалі всю суть, що “має бути у Всесвіті”.

Нарешті, залишається група проблем, яку можна позначити як завдання пошуку і проектування адекватних моделей даних і реалізації цих моделей в деяких фізичних структурах МБЗ.

Спробуємо сформулювати вимоги до моделі даних і її реалізації як інженерної, в конкретній формі, але при цьому не знижуючи рівня узагальнення, так, щоб не знищити суть задачі. Для цього достатньо послідовно перебрати всі вимоги і до самої “узагальненої моделі даних”, і до реалізованості цієї моделі в деяких фізичних структурах, тобто вимоги до можливості відображення деякої абстрактної структури МБЗ на фізичну реалізацію, архітектуру конкретної фізичної машини.

Тут нас цікавлять випадки “скінченно-модельовані, але з необчислюваним апріорі числом кроків”, оскільки умова існування відповідного перетворювача предикатів записується у вигляді набору післяумов (постумов). У сенсі реалізації це автоматично означає, що “абстрактна модель даних” може бути подана, в загальному випадку, як віртуальна структура, за принципом свого устрою не пов’язана з фізичними (числовими) параметрами реалізації. Тим більше, вся структура МБЗ не може бути інакше як віртуальною.

Але реалізація будь-якого конкретного прикладного завдання все одно вимагатиме відображення на

фізичний адресний простір. Звідси випливає вимога: структури “власне даних”, тобто способи (можливо і один, але універсальний спосіб) адресації повинні бути інваріантні до абстрактної моделі даних, що відображається. З урахуванням фізичної реалізації, це означає, що затрати на реалізацію способу адресації повинні “майже завжди (крім “множини нульової міри” випадків) не зростати швидше, ніж обсяг даних” Насправді втрата ефективності адресації єдина до цих пір відома причина саморуйнування інформаційних систем. Дешевше стає спорудити нову систему, ніж підтримувати неефективний механізм. Цей аспект ототожнюють із проблемою “прибирання сміття”, ідентифікації і знищення неефективних структур. Проте це не в точності те ж саме, неефективні структури повинні з’являтися в будь-якій системі такого роду, якщо ми визнали за ними “право на помилку”, що відображаються.

Необхідно виділити і наступний аспект, навіть не стільки як пряму “вимогу”, скільки як “підказку” для подальшого аналізу проблеми. Умови частотної реалізації перетворювача предикатів, тобто МБЗ, яка, скажемо поки що так – “може більше, ніж традиційна СУБД” мають вигляд деяких *перерахувань* предикатів другого порядку, але суттєво, “не в цілому”, а на кожному скінченному відрізьку часу.

Це означає, що таблиця на рис. 1 представляє лише перебування МБЗ на деякому “елементарному такті” її існування, для подання її на скінченному відрізьку часу буде потрібна вже як мінімум ієрархія з 3-х рівнів таблиць предикатів. Тобто, крім таблиці на рис. 1 необхідно ще два рівні “опису умов застосування”. Втім це факт до певної міри очевидний.

МБЗ “у цілому” є цілісною ієрархічною системою динамічних процесів. Звідси безпосередньо випливає висновок – до таких систем не припустиме застосування механічного принципу декомпозиції, в усякому разі його використання, так, як це прийнято в класичній теорії управління. Тут необхідно знайти подання, що охоплює всю сукупність динамічних процесів “вкладених один в одного”

Завершимо цей екскурс у питання реалізованості МБЗ наступним. Із міркувань чисто обчислювальних архітектури адресних, тобто по суті фон Нойманівських машин домінуватимуть на ринку, можливо, ще не одне десятиліття. Це означає, що питання відображення структури МБЗ на адресну архітектуру залишатиметься актуальним, але, враховуючи нетривіальність проблеми, її дослідження – окреме інженерне завдання. Інакше кажучи, поки існують адресні машини, питання розробки моделей даних будуть проблемою, доповнюючою завдання пошуку адекватних структур МБЗ.

Зі сказаного виходить, що поточним завданням розробки архітектури обчислювальної техніки на

достатньо тривалий відрізок часу є орієнтація на МБЗ. Звернемо увагу на наступний момент. Звичайно, сьогоднішні реальні інформаційні системи (технології, моделі даних), що “саморозвиваються”, називаються так, скоріше, в порядку авансу на їх майбутнє вдосконалення. Насправді, про здатність до саморозвитку поки що мова не йде, цим позначається лише достатня гнучкість моделей даних і наявність інструментарію для управління розвитком цих моделей. Але звідси випливає відразу і реальна постановка завдання саморозвитку, і постановка завдання створення архітектури машин вже наступного покоління.

У постановці завдання про створення “наступної за архітектурою машини, після МБЗ” ми хочемо вже більшого, здатності деякої машини дійсно самостійно сконструювати структури даних і свою загальну структуру. Можна сказати, що ми хочемо переходу до “живих” систем, що володіють якщо не “свободою волі”, то, принаймні, “прагненням вижити”.

Отже ще раз, все що робиться, є питанням технології конструювання, все інше просто “не потрібне”. (“Інформодинаміка”, В.М.Лачинов, А.О.Поляков).

### 5. Варіанти вертикальної інформаційної машини

У постановці завдання про створення “наступної за архітектурою машини, після МБЗ” ми хочемо вже більшого, здатності деякої машини дійсно самостійно сконструювати структури даних і свою загальну структуру. Можна сказати, що ми хочемо переходу до “живих” систем, що володіють якщо не “свободою волі”, то, принаймні, “прагненням вижити”.

Перейдемо до розгляду декількох можливих варіантів “блок-схеми” вертикальної машини.

Вельми цікаво порівняти приведені на рис. 2 три “схеми” елемента інформаційної машини (або самої машини, що те ж саме через самоподібність структур) – “абстрактну”, тобто побудовану логічно; “оптичну реалізацію”; “живий” високий нейрон (велику піраміду).

На рис. 2 використані наступні позначення:

а) **абстрактна машина**: Z – зона “дзеркала”, тривимірна пам’ять для двох тришарових M-структур даних; S – структура (метаструктура), S, S’ – “виходи”.

б) **оптична машина**: 1, 1’ – тришарова оптична (голографічна) пам’ять для структур даних, що “запам’ятанні” і що “сприймаються”; 2 – пучки “малих” оптичних волокон; 3 – “суматор” хвильових фронтів; 4 – “велике волокно” – формувач структуриршення;  $\Phi, \bar{\Phi}$  – “синфазний” і “парафазний” виходи.

с) **нейрон**: T – тіло; Ш – шипіковий механізм; У – волокна “навчаючих” і “робочих” входів; АБ – аксонний пагорб; В – волокна “верхніх зв’язків”; А – аксонне волокно.

Подальші подробиці, функціональне призначення решти типів нейронів і інших компонентів мозку встановити, у принципі, не важко, навіть шляхом логічного аналізу, зіставлення логіки роботи системи в цілому і топології зв’язків елементів.

Можливо навіть не знадобиться значний обсяг експериментів, хоча вся сукупність робіт і може виявитися досить запаморочливою, але це вже не має принципового значення.

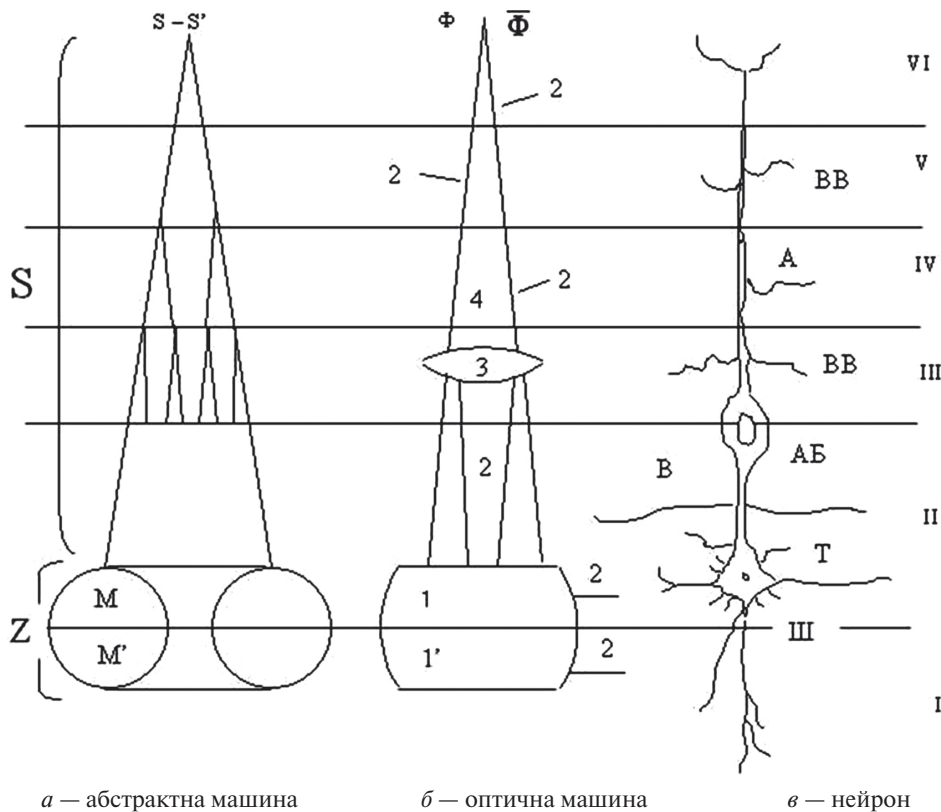


Рис. 2. Порівняння “схем” інформаційних машин

До того ж у виборі конкретних числових параметрів існує велика свобода – це очевидно з викладеного тут і наявного обсягу експериментальних матеріалів.

Дійсно, зіставимо – при абсолютно жорсткому, однаковому принципі дії “високого нейрона” і унітарній загальній схемі їх з’єднання, кількість входів кожного типу міняється від примірника до примірника більше ніж на порядок, те ж можна сказати і про співвідношення кількості входів різного типу (рівня). Тобто при абсолютній жорсткості принципу побудови спостерігається велика свобода реалізації.

Природно припустити, що у виборі параметрів, наприклад порогів, діапазонів їх зміни (регулювання) і дискретності регулювання свобода буде ще більша.

Але тоді виходить, що всі принципові речі щодо конструкції і роботи інтелектуальної машини тут вже сказані, розписані і можна прямо приступати до проекту реалізації машини – аналога, наприклад, на базі омігайда [10, 11]. Так би мовити, братися за викрутку і паяльник. І навпаки, - подальші теоретичні міркування зовні конкретного проекту мало що дадуть.

Важливий лише факт поступового всередині рівня і стрибкоподібної між рівнями якісної зміни сенсу сприйняття і обробки інформації, переходу від сигнального і контекстно-вільного до структурного і контекстно-залежного аналізу інформації.

У постановці завдання про створення “наступної за архітектурою машини, після МБЗ” ми хочемо вже більшого, здатності деякої машини дійсно самостійно сконструювати структури даних і свою загальну структуру. Можна сказати, що ми хочемо переходу до “живих” систем, що володіють якщо не “свободою волі”, то, принаймні, “прагненням вижити”.

Перші ж прикидки дають результат, цілком адекватний розмаху постановки:

- основні затрати підуть на навчання “всьому непотрібному” – так само як і у випадку з людиною;
- система також помилятиметься, матиме “власну думку” і з цим треба щось робити, в усякому разі, приймати це як належне;
- систему не можна вимкнути – для механізмів саморозвитку це “смертельно”
- систему не можна позбавляти надовго мінімально необхідного “потоків спілкування”, інакше неминуча деградація щодо відкритих систем, що “пішли вперед”, із складу його оточення.

Кожна проблемна галузь має не більше трьох рівнів метаструктур над структурою “елементної бази”.

Для цього з процесу еволюції виділимо деякі етапи, перед перерахуванням яких вкажемо: ми беремо “великі етапи”, ті, на яких сформувалися структурно-повні системи різних рівнів. Відзначимо їх характерні особливості.

Усередині кожного з етапів, що сформувалися, подальший розвиток йде тільки “вшир” – за рахунок різноманітності структур (підвидів) одного порядку. Чим “нижчий” рівень – тим “менші відмінності щодо розумності” і більше різноманіття.

На кожній наступному етапі інформаційна система попереднього етапу виявляється вже як цілісна периферійна система, у тому числі і морфологічно цілісна, тобто виділяється адекватний функціональному призначенню “апаратний рівень”. Зокрема, у хребетних спостерігаються чотири “великі етапи” – відповідно до поділу організації нервової системи.

Вперше зворотний зв’язок замкнувся на передінформаційному рівні в одноклітинних, тобто їх вихідні реакції на тому ж рівні, що й вхідні – фізичні, фізико-хімічні.

Структурно-повний рівень організації первинних сигналів склався у комах, а вищий рівень організації у них – зовнішній, як, наприклад, у бджіл. Тут ми спостерігаємо метарівень як колективну істоту. Це останній рівень, на якому припустимо говорити про існування якоїсь повної системи ізоморфізмів вхідних і вихідних сигналів поодинокі особи. Для наступних рівнів таке сказати вже не можна, більше того ізоморфізм сигналу і “того, що циркулює в контурі зворотного зв’язку ОС 2” існує лише для окремої особи, зразка.

Загалом, для сукупності осіб це “циркулююче” структурно, за своєю організацією, складніше, ніж сукупність сигналів. Строго кажучи, тут вже закінчується сигнальний рівень подання, ми можемо іменувати цей рівень сигнальною системою лише умовно, для прив’язки до традиційної термінології. Справедливо вважати, що саме тут і з’являється процес (феномен, явище) структурно складніший, ніж сукупність сигналів. Тут появляється інший, *інформаційний* рівень організації або *власне інформація як явище*, найперший рівень її організації, який ми позначимо як .

Надалі термін “сигнальна система” вживається для прив’язки до традиційної термінології і позначення того, що це подання використовується для зовнішньої, між окремими особами, комунікації. Так само стає умовністю позначення вищих рівнів організації як контуру зворотного зв’язку, це вже не функціональна залежність, але деяка згортка, не уявна, в загальному випадку, кінцевим алгоритмом.

У хребетних цей рівень, тобто сукупність периферичних нервових клітин і рівня, позначеного  $I_1$ , вже всередині. Він, і утворює функціонально і морфологічно повну зовнішню нервову систему. Їх (хребетних) “зовнішня сигнальна система” друга –  $I_2$ .

Відповідно у ссавців, аж до приматів третя –  $I_3$ .

“Сигнальна система людини” – найпримітивніша із стійких мов, що з’явилися, – “четверта сигнальна

система” (що нам доводиться констатувати всупереч твердженням біологів).  $I_4$  позначає вихідне подання деякої фізичної структури (кори), функціонально це вищий рівень організації нервової системи і у вищих, і у людини, але організований суттєво різним чином.

У людини сформувався інший рівень базової мови і базових образів, ніж у приматів. Для приматів “слово” і “образ слова” (“знак”) одне і те ж, для нас (принаймні, для більшості) – ні. Це можна спостерігати за тому ефекту, що у людини в корі строго спеціалізовані “мовна” і “зорова” зони, у приматів же повинні бути морфологічні або функціональні відмінності, оскільки при активній роботі “мовного центру” утворюється інший характер взаємодії з рештою кори. Людина “чує слова”, але “враховує й образи”.

Мова  $I_4$  це не потік слів, а потік образів, добудованих за допомогою контекстного аналізу, тобто це четверта сигнальна система – інакше неминуча плутанина в аналізі. Те ж саме і зір: образ видимий і образ сприйманий свідомістю різні.

Але нерозуміння того, що на вхід мовного аналізатора, наприклад, для управління комп’ютером, надходить сигнал четвертого рівня, тобто зі середовища, що має над фізичним рівнем (власне сигнальним) структурно-повну організацію, веде до створення нестійко працюючих конструкцій, це спроба обробки того контексту, який спочатку не був передбачений в системі.

## 6. Структура $k$ -значної площинно-просторової комірки

Таким чином, усі відкриті системи, незалежно від “матеріалу” і способу реалізації “апаратної бази”, утворюють чотири класи або рівні складності  $I_1$ ,  $I_2$ ,  $I_3$ ,  $I_4$  [17].

$I_1$  – кібернетичні системи, в яких поняття і подання інформації ізоморфне потоку даних (потоків станів сигналу);

$I_2$  – ізоморфізм “сигналу” і інформації зникає, “верхнього” контексту набуває властивість самостійного руху без передачі сигналу;

$I_3$  – клас “повних” інформаційних систем, що складаються з трьох рівнів, організованих за тією ж схемою, що й сам рівень. Структура є і метаструктурою, тобто схемою організації всієї системи, при цьому структура є мінімаксною, тобто мінімальною структурою, що дозволяє відобразити стан системи для загального випадку (для довільних потоків даних за відсутності яких-небудь спеціальних апріорних обмежень) і, одночасно, це максимальна за складністю структура, яка може бути реалізована в тривимірній “апаратурі” знову ж таки без спеціальних апріорних умов;

$I_4$  – системи типу “інтелект”, що існують як надбудова над рівнем (системою)  $I_3$  у вигляді віртуальної “вертикальної” інформаційної машини принципово необмеженої “висоти” в спеціальним

чином організованому “апаратному середовищі”. “Вертикальна машина” за способом облаштування є безадресною за рахунок того, що працює на основі квазістаціонарних потоків, точніше за двома системами потоків – пара потоків “власне даних”, тобто структур типу  $I_3$  і пара “потоків уяв” – структур класу  $I_4$ . Разом із набором правил-умовчань і “системою команд”  $I_4$  складає єдино можливу елементну базу для універсальної інформаційної машини – розуму.

Розглянемо рівні організації складних систем, і надамо їм новий сенс і порядок відповідно до результатів проведеного нами аналізу та проілюструємо сказане вище додатковою схематичною побудовою:

**Комахи:** сигнал → згортка 1 → “мова 1”; (між сигналом і мовою повний ізоморфізм, це колективна істота).

**Хребетні:** сигнал → згортка 1 → інформація → “мова 1” → згортка 2 → “мова 2”.

**Примати:** сигнал → згортка 1 → інформація → “мова 1” → згортка 2 → “мова 2” → згортка 3 (межа інтелекту) → “мова 3”.

**Людина:** сигнал → згортка 1 → інформація → “мова 1” → згортка 2 → “мова 2” → згортка 3 (межа інтелекту) → “мова 3” → згортка 4 → “мова 4”.

Теоретичні та експериментальні дослідження, а також ускладнення, що виникають під час створення обчислювальних систем новітніх поколінь [10-14], сприяють висуненню концепції адекватності  $k$ -значної логіки та структур, завданню створення обчислювальних систем новітніх поколінь з очікуваними властивостями й можливостями щодо підвищеної швидкодії і широких функціональних можливостей. Зокрема, універсальні  $k$ -значні структури просторового типу із витоків володіють граничним паралелізмом на рівні базового елемента і працюють із багаторівневим сигналом, є універсальними в рамках вибраного структурного алфавіту і володіють гнучкістю перенастроювання без зміни безпосередньо самої структури.

Довільна структура на системному рівні характеризується набором функцій, що реалізуються нею та функціональними вузлами, які реалізують ці функції, а також інформаційним обміном під час функціональних перетворень. Відповідно до задач, що вирішуються, структурно-функціональна комірка узагальненого виду на рівні системного підходу декомпозується (рис. 3) на три ієрархічні рівні: *функціональний* (аналітико-синтетичний) –  $I_1$ ; *тактичний* (аналізаторно-координаційний) –  $I_2$ ; *стратегічний* (координаційний) –  $I_3$ . Крім цього, повна структура  $k$ -значної комірки ШІ, за внутрішньою логікою дії, повинна включати зв’язок із наявною і необхідною базою знань, що утворює четвертий рівень ієрархії –  $I_4$ .

Відповідно на функціональному рівні до складу  $k$ -значної площинно-просторової комірки входять:  $n$ -вимірний комутатор сигналів; комплекс порогових



Рис. 3. Концептуальна структурно-функціональна модель  $k$ -значної комірки – для рішення задач систем штучного інтелекту

пристроїв, дешифратори просторових проміжних ознак та формувачі  $k$ -значних функцій.

*Комутатор сигналів* є керуючим і спрямовуючим пристроєм входів обчислювальних систем новітніх поколінь, що визначає, з яким за типом вхідним сигналом працює комірка: зовнішнім чи від стратегічного рівня.

*Комплекс порогових пристроїв* дозволяє здійснювати перетворення неперервних чи дискретних за часом та за рівнем  $k$ -значних сигналів (семантичне опрацювання вхідного сигналу системи), а також формування простору проміжних ознак (простору суттєво  $k$ -значних характеристичних функцій) як результату семантичного опрацювання. Проміжні ознаки дешифруються в керуючі сигнали вихідного комплексу *формувача  $k$ -значних функцій*, що здійснює аналітичні функціональні перетворення. Результат перетворень на функціональному рівні надходить на вихід комірки, а також надходить для оцінювання з точки зору семантичного змісту на стратегічний рівень.

*Тактичний рівень* у площинно-просторовій комірці реалізується з допомогою *аналізаторно-координаційного* процесора, у завдання якого входить керування комутатором вхідних сигналів, налагодження порогових пристроїв функціонального рівня, налагодження дешифраторів проміжних ознак із метою вибору виду здійснюваного функціонального перетворення та синхронізація роботи функціонального, тактичного і стратегічного рівні.

На *стратегічному рівні* процесор-супервізор дозволяє системам ШІ здійснювати остаточний семантичний аналіз із участю оператора, обмін даними з оператором, входом та виходом комірки, із базою знань обчислювальних систем новітніх поколінь, а також контролювати в автоматичному та діалоговому режимах процес вирішення задач на тактичному

рівні. Останнє дає можливість прослідкувати покроково процедури реалізації довільного алгоритму на всіх етапах його виконання і, тим самим, здійснити селекцію та нагромадження в базі знань ефективних алгоритмів із множини інших, менш ефективних.

Концептуальна структурно-функціональна модель комірки базується на концепції симбіозу (нерозривного зв'язку та взаємодії) дво- та  $k$ -значних засобів опрацювання інформації, тому на стратегічному рівні в ній містяться комплекси перетворювачів форми відображення даних – перетворювачі двозначного коду в  $k$ -значний ( $2 \rightarrow k$ ) та навпаки ( $k \rightarrow 2$ ). Застосування чи ні цих перетворювачів у обчислювальних системах новітніх поколінь визначає, на якому функціональному рівні вирішуються задачі, в якій логіці (у якому структурному алфавіті) та з якою швидкістю. Крім цього, використання таких перетворювачів, зокрема при  $k = 10$ , усуває необхідність роботи оператора з двозначними трансляторами під час уведення/виведення даних, у  $\log_2 k$  зменшує число входів на всіх рівнях.

Підсумовуючи процес логіко-інтуїтивного емпіричного системного синтезу структурно-функціональної моделі узагальненої комірки зауважимо, що така концепція є апіорною і сформована на основі аналізу широкого спектра власних наукових досліджень та досліджень учених, які працюють над створенням обчислювальних систем новітніх поколінь, а також тих, що вивчають природний інтелект. Із плином часу запропонована автором чотирирівнева ієрархічна модель інтелекту отримала визнання та набула значного поширення і використовується різними дослідниками у галузях цифрової обробки сигналів, інтелектуальних роботів, нейроінтелекту, теології, побудови мовних систем штучного інтелекту, психології інтелекту.

### Висновки

Сфери предметних галузей, де найдоцільніше працювати з даними і знаннями, поданими мовними моделями — це галузі з переважанням емпіричного знання, де складність фактів і їх описів виключає використання мови математики і зокрема інформаційно-інтелектуальних технологій та інтелектуальної підтримки суспільно-економічних процесів діяльності фінансово-кредитних установ.

Найбільш загальна проблема побудови системи управління семантичного чи семантико-прагматичного рівня взаємодії, пов'язана з вибором технології контекстно-залежного подання знань, побудовою інформаційних баз (даних і знань) про предметну галузь і механізму висновку для отримання необхідних рішень.

Єдино відомим нам, об'єктивним носієм знання та інтелекту є людина, а виразником, засобом до зовнішнього спілкування та носієм інтелекту є людська мова, що й складатиме об'єкт і напрямок досліджень у наступних частинах роботи.

Подання моделей декларативних мов здійснюється предикатами, малюнками, кресленнями, графами тощо.

Математична структура даних у декларативних мовах базується на системах предикатних рівнянь у алгебрі скінченних предикатів або на аксіоматичній теорії множин, у якій теорія множин інтерпретується як структура даних.

Проблема створення ШІ полягає не в тому, щоб «будувати штучних людей», а в тому, щоб пізнати природні організми настільки, щоб використовувати їх на рівні систем, тобто в керованому або коректованому режимі, зробити їх керованими.

Кожний атрибут системи ШІ сам може бути універсальним функціональним перетворювачем, але іншого ієрархічного рівня. З цього випливає, що універсальний функціональний перетворювач є системою рекурсивною.

УФП має свою мову і починає жити і діяти, реалізуючи свої властивості, після того, як процес програмування завершений і ЕОМ працює за певним керуючим алгоритмом налагодження.

УФП перетворюється (переходить) у базову комірку ШІ тільки в працюючому комп'ютері.

Вихідні реакції УФП наповнені змістом, тому і називаються власними уявленнями УФП про керуючий вплив, але оскільки джерелами зовнішніх по відношенню до УФП збурень можуть служити будь-які об'єкти і дані, то саме його (УФП) уявлення є новою інформацією, а сам УФП — джерелом цієї інформації.

Оскільки світ нелінійний, основні закономірності розвитку неживої і живої природи (від мікро- до макросвіту), зокрема соціальних і економічних структур, є нелінійними, тобто результат не прямо пропорційний причині, що його викликала.

Майбутнє неоднозначно визначається початковими умовами, його не можна передбачити, спираючись

тільки на попередній досвід. Таким чином, нелінійність і є по суті справжнім переходом від минулого до майбутнього. У зв'язку з цим можна зробити висновки, що процесом функціонування УФП необхідно управляти, щоб він став оптимальним за результативністю.

### Список літератури:

- [1] Бондаренко М.Ф., Коноплянко З.Д., Четвериков Г.Г. Основи теорії синтезу надшвидкодуючих структур мовних систем штучного інтелекту. — К.: ІЗМН, 1997. — 264 с.
- [2] Бондаренко М.Ф., Коноплянко З.Д., Четвериков Г.Г. Основи теорії багатозначних структур і кодування в системах штучного інтелекту. — Х.: Фактор-Друк, 2003. — 336 с.
- [3] Коноплянко З.Д., Четвериков Г.Г. Проблеми інтелектуалізації та українізації цифрових систем та мереж телекомунікацій // Труды УНИИРТ. — 1995. — № 4. — С. 72–81.
- [4] Коноплянко З.Д., Четвериков Г.Г. Аналіз лінгвістичних зв'язків елементів фонетичного рівня української мови // Праці УНДІРТ. — 1996. — № 1(5). — С. 70–77.
- [5] Четвериков Г.Г., Ляховець С.В., Коноплянко З.Д., Колесник А.С. Аналіз та дослідження букво-фонемних відношень української мови // Праці УНДІРТ. — 2001. — № 1(25). — С. 79–83.
- [6] Осыка А.Ф., Кравець О.А. О формализации семантики словосочетания с инструментальным значением // Проблемы бионики. — Вып. 45(90). — Харьков: Основа. — 1990. — С. 3–10.
- [7] Булкин В.И., Ситников Д.И., Шабанов-Кушнарченко Ю.П. Математические модели межморфемных связей на множестве полисемантических производящих основ и словообразовательных суффиксов // Проблемы бионики. — Вып. 47. — Харьков: Основа. — 1991. — С. 3–22.
- [8] Осыка А.Ф., Кравець О.А. К вопросу о формировании семантических признаков слов. // Проблемы бионики. — Вып. 47. — Харьков: Основа. — 1991. — С. 8–16.
- [9] Рябова Н.В. Моделирование семантики производных слов русского языка. // Проблемы бионики. — Вып. 45(90). — Харьков: Основа. — 1990. — С. 17–23.
- [10] Шабанов-Кушнарченко Ю.П. Теория интеллекта. Математические средства. — Харьков: Выща шк., 1984. — 144 с.
- [11] Шабанов-Кушнарченко Ю.П. Теория интеллекта. Технические средства. — Харьков: Вища шк., 1986. — 136 с.
- [12] Шабанов-Кушнарченко Ю.П. Теория интеллекта. Проблемы и перспективы. — Харьков: Вища шк., 1987. — 160 с.
- [13] Gorokhovatskyi V., Vechirska I., Chetverikov G. Method for Building of Logical Data Transform in the Problem of Establishing Links between the Objects in Intellectual Telecommunication Systems // Telecommunications and Radio Engineering, 2016. — Vol. 75. — P. 1645–1656.
- [14] G. Chetverykov, O. Puzik, and O. Tyshchenko. Analysis of the problem of homonyms in the hyperchains construction for lexical units of natural language, in 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), 2018, pp. 356–359, IEEE.
- [15] Повзун О.І., Козел Н.Б., Четвериков Г.Г. Концепції організації інформаційно-інтелектуальних технологій та інтелектуальної підтримки суспільно-економічних процесів: к-значні засоби. частина 1. [Текст] / О.І. Повзун, Н.Б. Козел, Г.Г. Четвериков // «Бионика интеллекта». — 2020. — №1(94). — С. 88–93.
- [16] Shliakhov, V., Chetverykov, G., Bozhko, I., Shliakhova, N. (2019). Predicate Data Model in the Form of a Linear Space. ECONTechMOD: An International Quarterly Journal on Economics of Technology and Modelling Processes, 8(2), 41–54.
- [17] Васильев Н.Р., Задорин Е.В. Математическая модель псевдоорганизма (КАУС) — путь к созданию искусственного интеллекта. — 2008. — <http://arupa-manas.narod.ru/BASE/nauka/index.html>

Надійшла до редакції 29.09.2020

Д.О. Золотарьов<sup>1</sup>

<sup>1</sup>Кандидат фізико-математичних наук, м. Харків, Україна,  
denis@zolotariov.org.ua, ORCID: 0000-0003-4907-7810

## МЕТОДИ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ІТЕРАЦІЙНИХ ОБЧИСЛЕНЬ У WOLFRAM MATHEMATICA

Розглянуті та досліджені методи скорочення обчислювальних ресурсних витрат ітераційних алгоритмів у математичному процесорі Wolfram Mathematica на основі попередньої підготовки вхідних даних, управління проміжними даними у процесі виконання, використанню та специфічному налаштуванню вбудованих функцій: експорту у платформи незалежні формати та у файли дампу, компілювання у код Wolfram Virtual Machine та машинний код за допомогою C-компілятора, операторів циклу. Показані їх неочевидні можливості та особливості. Доведено, що кожний з них навіть окремо здатен істотно впливати на скорочення використання таких обмежених ресурсів комп'ютера, як об'єм оперативної пам'яті та час центрального процесора при розрахунках.

ІТЕРАЦІЙНІ АЛГОРИТМИ, WOLFRAM MATHEMATICA, МЕТОДИ СКОРОЧЕННЯ ОБЧИСЛЮВАЛЬНИХ РЕСУРСНИХ ВИТРАТ

**Золотарьов Д.А. Методы повышения эффективности итерационных вычислений в Wolfram Mathematica.** Рассмотрены и исследованы методы сокращения вычислительных ресурсных затрат итерационных алгоритмов в математическом процессоре Wolfram Mathematica на основе предварительной подготовки исходных данных, управления промежуточными данными в процессе выполнения, использования и специфической настройки встроенных функций: экспорта в платформы независимые форматы и файлы дампа, компиляции в код Wolfram Virtual Machine и машинный код с помощью C-компилятора, операторов цикла. Показаны их неочевидные возможности и особенности. Доказано, что каждый из них даже по отдельности способен существенно влиять на сокращение таких ограниченных ресурсов компьютера, как объем оперативной памяти и время центрального процессора при вычислениях.

ИТЕРАЦИОННЫЕ АЛГОРИТМЫ, WOLFRAM MATHEMATICA, МЕТОДЫ СОКРАЩЕНИЯ ВЫЧИСЛИТЕЛЬНЫХ РЕСУРСНЫХ ЗАТРАТ

**Zolotariov D. The methods for iterative computations efficiency improving in Wolfram Mathematica.** In article considered and investigated methods for reducing computational resource costs of iterative algorithms in the mathematical processor Wolfram Mathematica that based on preliminary preparation of initial data, management of intermediate data in the execution time, use and specific tuning of built-in functions: export to platform independent formats and dump files, compilation into Wolfram Virtual Machine code and machine code using a C-compiler, loop operators. Their non-obvious capabilities and features are shown. It has been proven that each of them, even individually, can significantly affect the reduction of such limited computer resources as amount of RAM and CPU time during calculations.

ITERATIVE ALGORITHMS, WOLFRAM MATHEMATICA, METHODS FOR REDUCING CALCULATION COSTS

### Вступ

Сучасна прикладна наука майже повністю перейшла від пошуку точних аналітичних розв'язків задач до наближених чисельних широко розповсюдженими FDTD [1-5], МКЕ [6-10] та іншими. Такі методи, крім вагомих обмежень на вид розв'язуваної задачі, для одержання результату з досить високою точністю вимагають використання значних обчислювальних потужностей. Але використання великих комп'ютерних кластерів доступно далеко не всім дослідницьким групам, не говорячи вже про окремих вчених та інженерів. Тому над проблемою підвищення продуктивності чисельних методів працюють вже майже півстоліття.

До 90-х років зусилля наукової спільноти були переважно спрямовані у бік їх аналітичного вдосконалення, а після початку ери персональних комп'ютерів сфера інтересів перейшла до обчислювального поліпшення – такої зміни алгоритмів, завдяки якій вони найбільш оптимально використовують ресурси комп'ютера.

З моменту появи математичних процесорів відбувся новий поворот у напрямку досліджень – і сьогодні пошук шляхів підвищення ефективності чисельних методів та програм на їхній основі йде також у контексті використання матпакетів [11,12], що служать прошарком між алгоритмом програми та ресурсами комп'ютера і вносять істотні корективи у алгоритми через велику різницю у функціонуванні у порівнянні із компільованими мовами програмування та унікальні для кожного пакету особливості внутрішньої реалізації стандартних операцій: побудова матриць, чисельне інтегрування, побудова циклів та інше. На сьогодні одним із самих широко популярних у інженерів та дослідників різного напрямлення є математичний процесор Wolfram Mathematica [13], що використовується для моделювання фізичних, хімічних, біологічних процесів, нейронних мереж та інтелектуальної обробки даних.

Разом із універсальними методами розробляються та розвиваються також і спеціалізовані чисельно-аналітичні [14], що використовують ітераційний

підхід або метод вікна: на кожній ітерації обчислюється частина шуканих значень функції в деякій підобласті її аргументів (вікна), що є сусіднім до вікна попередньої ітерації. При цьому важливим припущенням є використання на поточній ітерації лише деякої незначної кількості результатів попередніх ітерацій. Їхнім недоліком у порівнянні із універсальними є дуже малий спектр розв'язуваних задач, але це компенсується такими перевагами як:

- отримання більш точного розв'язку своєї категорії задач,
- скорочення ресурсів комп'ютера, необхідних для обчислень,
- можливість контролювати точність результату через варіативність вікна без зростання використовуваних обчислювальних ресурсів комп'ютера.

До даного моменту були опубліковані роботи дослідників і програмістів, присвячені правилам та прийманням оптимальної розробки в пакеті Mathematica [15-18]. Але у публікаціях робиться акцент на процес виконання коду програми і його швидкодії, тому що автори виходять із неявного припущення, що основним обмеженим ресурсом комп'ютера є час центрального процесора (ЦП), що вірно для великих обчислювальних систем. І в той же час упускається з виду такий ресурс як оперативна пам'ять (ОП). Але недостатність останньої веде до затримки обчислень на кілька порядків [19,20] у випадку включеного swap-файлу [21], а якщо він виключений – робить такі розрахунки попросту неможливими [21]. Також жодної роботи не присвячено дослідженню використання комп'ютерних ресурсів та їх оптимізації саме для ітераційних алгоритмів, які мають своєю метою мінімізацію такого використання без зменшення точності обчислення.

Тому актуальною є задача дослідження та розробки рекомендацій щодо підвищення ресурсної обчислювальної ефективності ітераційних алгоритмів у математичному процесорі Wolfram Mathematica, якій присвячена дана робота.

### 1. Оптимізаційні підходи

З практики автора [22], найбільш перспективними представляються наступні методи підвищення продуктивності ітераційних алгоритмів програм у Wolfram Mathematica:

- попередня підготовка даних та функцій для мінімізації часу їх завантаження та обчислення,
- вивантаження на зовнішню пам'ять (ЗП) або видалення даних, що не використовуються у рамках поточної ітерації або її кроку,
- уникнення повторного обчислення виразів,
- визначення оптимального місця для супутніх обчислень,

- компілювання функцій,
- вибір правильного оператора циклу.

Дослідимо далі їх вплив на швидкодію (час виконання ЦП) та об'єм використовуваної ОП. Усі результати отримані на сервері з ОС Ubuntu 20.04 LTS x64, що розташований на потужностях одного із світових лідерів хмарних сервісів DigitalOcean, та ядром Wolfram Mathematica 11.3.0 (64-bit).

### 2. Попередня підготовка вхідних даних та функцій

До початку будь-яких обчислень, що представляють собою деякий набір операцій у циклі, необхідно зрозуміти, що може бути розраховане до головного циклу, а також після нього.

Дуже часто чисельні алгоритми оперують лише табличними значеннями функцій, при цьому її аналітичне вираження, обчислення якого може займати значний час, зовсім не потрібно. Варто звернути увагу також на наступний накопичувальний ефект: функція сама по собі може бути «легкою» у одноразовому обчисленні, але при тисячах або мільйонах обчислень стати «важкою» у загальному та «вузьким» місцем програми.

У такому випадку найкращим варіантом буде замість постійного циклічного виклику вихідної функції в аналітичному вигляді до початку роботи один раз побудувати її табличний вид та зберегти його у ЗП, а в циклі звертатися до готової таблиці. Такий підхід не тільки мінімізує використання ОП, але й на кілька порядків збільшує швидкодію додатка на «важких» функціях за рахунок того, що складний набір з арифметичних операцій із плаваючою комою замінюється простим читанням даних з певної комірки пам'яті. У разі використання у рамках ітерації тільки малої частини такої таблиці, рекомендується на початку ітерації завантажувати таку частину у ОП та видаляти її після зникнення потреби.

Ще більшої економії ОП можна досягти тільки через компроміс із часом обчислення – читати тільки потрібні значення із таблиці у ЗП. Але це також призведе до значного загального сповільнення роботи програми в силу малої швидкодії ЗП порівняно із ОП.

Якщо ж такий спосіб не підходить і вимагається тільки аналітичне вираження, то альтернативою може служити аналітичне спрощення функції: обчислення інтегралів і похідних, де це можливо, і так далі.

Проміжним варіантом може бути створення поліноміального аналога (наприклад, за допомогою сплайнів) вихідної функції із заданою точністю [14]. Часто такий варіант є дуже гарним компромісом між аналітичним завданням і табличним. До того ж обчислення поліномів займає набагато менше ОП та часу ЦП ніж більш складні операції.

Загальним правилом повинно стати: усе, що може бути спрощене до початку роботи програми, і тим паче основного циклу програми, повинно бути спрощене.

### 3. Вивантаження даних, що не використовуються у рамках поточної ітерації

Перед пошуком «зайвих» даних треба впевнитись, що вже видалені із ОП:

- дані, які потрібні лише на етапі запуску програми,
- тимчасові змінні.

Також безумовному вивантаженню підлягають результати попередніх ітерацій роботи алгоритму програми, які не потрібні для обчислення поточної.

Для досягнення максимального ефекту рекомендується постійно контролювати, які дані знаходяться у ОП у даний час, та критично визначати необхідність їх наявності там, керуючись питанням: чи беруть участь безпосередньо в обчисленнях на поточній ітерації алгоритму ці дані?

У математичному процесорі Mathematica для звільнення ОП служить функція ClearAll [23]. Розробникам на компільованих мовах програмування знайомі засоби очистки ОП від зайвих даних, що миттєво повертають визначений об'єм пам'яті до ОС. В силу того, що програма у Mathematica не є скомпільованою, виклик функції ClearAll[X] очистки змінної X не завжди призводить до повернення пам'яті, займаної X, до ОС. Це пов'язано із тим що за замовчуванням матпакет зберігає всю історію обчислень для можливості звернення до неї через якоря %, %2 і так далі. Встановлення глобального налаштування \$HistoryLength=0 дає змогу матпакету миттєво очищувати пам'ять за запитом користувача.

У разі коли дані потрібні в рамках поточної ітерації, але не на поточному її кроці, а значно далі, – рекомендується їх вивантажувати на ЗП. У Mathematica є два поширені способи експорту даних у:

- платформо незалежні формати функціями Export / Import [24,25];
- платформо залежний внутрішній формат функціями DumpSave / Get [26,27].

Перший спосіб зручний для використання експортованих даних у сторонніх додатках для аналізу та подальшої обробки. Другий – для максимально швидкого зворотного завантаження у Mathematica.

Доведемо, що експортування та імпортування за допомогою функції DumpSave є максимально ефективним, окрім тих випадків, де потрібні спеціальні формати для обробки у інших комп'ютерних додатках.

Розглянемо швидкість експортування та завантаження матриць за допомогою Export із використанням найпоширеніших форматів файлів: звичайного

txt, що оптимізований для редагування користувачем, та бінарного формату MATLAB для матриць. Та порівняємо її із аналогічною для DumpSave.

Порівняння швидкодії «експорт/завантаження» цих двох способів для 20 ітерацій для таблиці 200x500 чисел із плаваючою комою наведений на рис. 1, де: Txt, MATLAB та Dump означають формат текстовий, MATLAB та DumpSave відповідно; (e) та (i) – експорт та імпорт відповідно. Абсциса – номер ітерації, ордината – час виконання операції у секундах. Слід зазначити, що розміри експортованих файлів мало відрізняються: 76Мб для MATLAB та DumpSave, та 93Мб для Txt. Запис ведеться на SSD-накопичувач серверного типу.

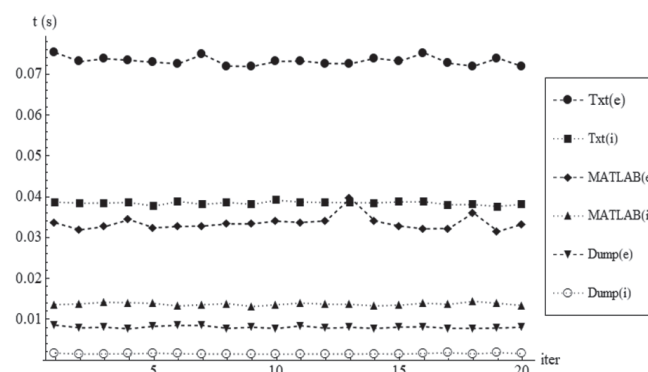


Рис. 1. Порівняння швидкодії експорту/імпорту у форматі txt, MATLAB та DumpSave

Як видно із рисунка вище, графіки функцій досить монотонні, і не має сенсу проводити експеримент для більшої кількості ітерацій.

Експортування у формати зручні для редагування користувачем (txt) мають найменшу швидкість, бінарні (MATLAB) – значно швидші, але без сумніву внутрішній бінарний формат Mathematica, що дає DumpSave, найшвидший і при експортуванні, і при імпортуванні. Це відбувається завдяки тому, що при експортуванні у дампи дані напряму переписуються із ядра Mathematica до файлу, а для будь-якого іншого формату – повинні пройти конвертацію із внутрішнього у потрібний.

Безумовно, інші текстові та бінарні формати будуть мати інші швидкості експорту/імпорту але результат залишиться тим самим у порівнянні з DumpSave, бо завжди будуть мати місце накладні витрати на конвертування у специфічний формат даних із внутрішнього, що знаходиться у ОП.

Рекомендується тому використовувати усюди експортування за допомогою функції DumpSave. Крім тих випадків, де явно потрібні специфічні формати для редагування або обробки у інших комп'ютерних додатках. Та, як наслідок, не рекомендується експортувати дані у будь-який формат окрім дампу усередині основного циклу алгоритму – а винести такі операції за його межі.

#### 4. Уникнення повторного обчислення

Wolfram Mathematica за замовчуванням усі обчислення здійснює та зберігає у максимально точному арифметичному форматі. Наприклад,  $5/8$  зберігається як арифметичний об'єкт Rational[5,8] (займає 56 байт у ОП) замість числа 0,625 (16 байт). Прямі вказівки використання чисельних розрахунків може не дати очікуваного результату через внутрішнє правило: якщо хоча б один компонент вираження має арифметичний вид – усе вираження обчислюється як арифметичне. А це в свою чергу може сповільнити весь розрахунок на один або більше порядків [15]. Тому уникнення повторного обчислення одних і тих самих виражень чи функцій – є важливою умовою ефективного використання у цьому матпакеті не тільки ресурсів ОП, а також ЦП. Навіть якщо вираження «легке» у однократному обчисленні, воно може істотно впливати на швидкість та використання ОП програмою, якщо обчислюється багато разів.

Для уникнення цієї проблеми необхідно виявляти усі дублікати обчислень на етапі програмування, зберігати та використовувати обчислені раніше значення змінних та константних для певного набору аргументів функцій замість повторного їх обчислення. За потреби, зберігати ці значення у ЗП для використання у віддалених ітераціях (розрахунках).

Продемонструємо це на прикладі двох функцій: «легкої»  $f1(x)$  та «важкої»  $f2(x)$ , що одиночно обчислюються за 0,000336 та 0,048106 секунди відповідно. Зміст функцій не важливий, має сенс лише час їх виконання.

Рис. 2 описує швидкість розрахунку цих функцій для 10 різних аргументів: що обчислюються повторно (Normal) та не обчислюються завдяки зберіганню після першого обчислення (Optimized) при 40, 160, 640 та 2560 звертаннях до кожної з них. Графік логарифмічний.

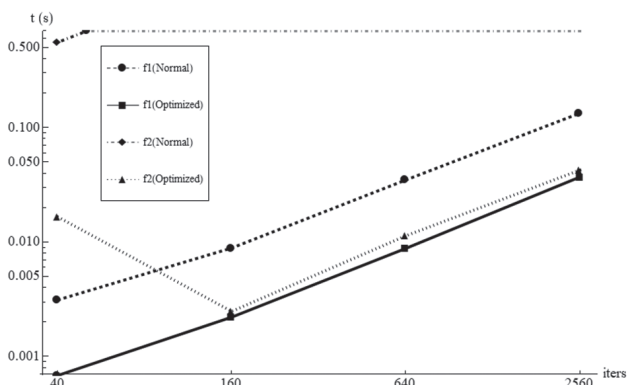


Рис. 2. Порівняння швидкості обчислення «легкої» та «важкої» функцій при повторному обчисленні та без нього

Вочевидь, при невеликій кількості звернень приріст швидкості майже не відчутний та він значно зростає із ростом звернень. Також не потребує доказів і те, що для більш складних функцій різниця буде відчутна вже на другому зверненні – графік

$f2(\text{Normal})$  при 40 повторних обчисленнях починається із 0.5с і зростає до 35с для 2560 ітерацій (для збереження масштабу – показаний пунктирною горизонтальною прямою зверху рисунка).

#### 5. Супутні обчислення

Окремо варто зупинитися на супутніх обчисленнях. Це необхідні у майбутньому (можливо, віддаленому) у рамках алгоритму розрахунку, які ресурсно вигідніше виконати в межах поточної ітерації по її остаточних або проміжних результатах. Наприклад, паралельно із точним значенням деякої функції обчислювати і її інтегральне значення, похибку, відхилення від заданої кривої і таке інше.

Виконання таких побічних розрахунків паралельно дуже зручно. Особливо, якщо потрібен вивід таких даних у реальному часі або витрати на їхнє обчислення поза межами основного циклу в разі перевищують такі всередині нього. Але треба мати на увазі – такі розрахунки завжди вимагають додаткових в рамках ітерації ОП та часу ЦП, тому найчастіше варто втриматися від пакетного виконання основних операцій ітерації алгоритму та побічних.

Рекомендується використовувати правило: відділяти та виносити після основного циклу програми максимальну кількість розрахунків. Такий підхід дає не тільки економію ресурсів ПК, але й виключає внесення додаткових джерел збоїв усередину основного циклу програми.

Варто зазначити, що за супутні обчислення не рахується побудова графіків у реальному часі, тому що у Mathematica один графік може займати від десятків Кб до декількох Мб даних, що потрібно запакувати на стороні ядра, передати клієнту, розпакувати та відобразити: наприклад, наведені у тексті вище графіки на Рисунку 1 та Рисунку 2 займають 32 та 62 Кб відповідно. При використанні віддаленого ядра Mathematica увесь процес виводу може займати від одиниць до десятків секунд, не рахуючи час на побудову графіку.

Рекомендується тому усіляким чином уникати побудови та виводу будь-яких графіків в рамках ітерації та перенести ці операції після закінчення основного циклу.

#### 6. Компілювання функцій

У Mathematica існують два типи компілювання, використовуючи функцію Compile [28]:

у байт-код для внутрішньої віртуальної машини WVM (Wolfram Virtual Machine),

у машинний код через C-компілятор, використовуючи опцію CompilationTarget -> «C».

У другому випадку математичний пакет створює зовнішню бібліотеку, що далі під'єднується до поточного ядра.

Також підтримується багатопотокове виконання скомпільованого коду через опцію Parallelization ->

True в обох типах. Але реальна кількість використаних під час обчислення процесорів залежить від здібності до паралельних обчислень конкретного алгоритму.

Треба мати на увазі, що Wolfram Mathematica не має вбудованого C-компілятора та використовує зовнішній, що має бути встановлений користувачем. А якщо такий не знайдений – за замовчуванням та без додаткового попередження використовується компілювання у код WVM.

Кожен з типів компілювання здатен істотно прискорити обчислення. Та якщо потрібно отримати максимум продуктивності – то рекомендується використовувати мову C, що з практики [14,28] на 20–500% швидше за WVM в залежності від функцій, що компілюються.

Емпіричне правило компілювання у C замість WVM, отримане при використанні компілятора gcc version 9.3.0 (Ubuntu 9.3.0-10ubuntu2), таке: функції, що складаються із арифметичних операцій дають значний приріст швидкодії, до декількох сотень відсотків; функції, що використовують складні вбудовані методи, наприклад, чисельне інтегрування, дають приріст до 15-25% через те що, по-перше, вони самі розроблені на мові C, по-друге, перед виконанням неявно компілюють свої аргументи у код WVM.

### 7. Вибір оператора циклу

Розробникам на компільованих мовах програмування знайомі три класичні типи операторів циклів: while, for та do – що відрізняються у таких мовах лише формою запису, бо компілюються у майже ідентичний за швидкістю машинний код. Але для Mathematica це три різних оператора [29,30,31], що всередині мають свої особливості внутрішньої реалізації.

Оператором циклу також може виступати функція Table [32] при нестандартному використанні. Її особливістю є те, що вона перед виконанням компілює у байт-код WVM функцію-аргумент [15]. Але вона до того ж є самим ресурсномістким варіантом циклу з точки зору ОП, бо разом із циклом формує дані – таблицю.

Рис. 3 описує результат експерименту розрахунку суми натуральних чисел на відрізку  $[1, N]$ , де  $N = [5000, 25000, 125000, 625000]$ , для кожного з операторів циклу. Для оператора Table побудова циклу сумування виглядає наступним чином:

```
sum = 0;
a = Table[sum+=i; 0, {i, sumN}];
Clear[a];
```

Тут формується таблиця з нулів, що у процесі своєї побудови також виконує корисну дію.

У експерименті навмисно не приймає участь спеціалізована функція Sum [33], що призначена для обчислення суми та виконується майже на два порядки

швидше, через те що метою експерименту є порівняння операторів циклу, а не пошук найоптимальнішого способу підрахунку суми.

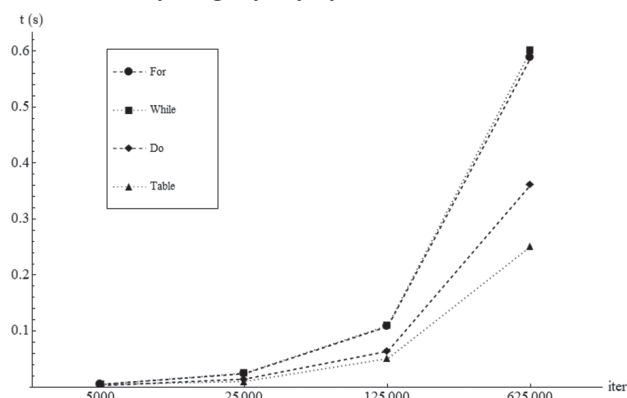


Рис. 3. Порівняння швидкодії операторів циклу While, For, Do та функції Table

Видно, що цикли For та While виконуються здебільшого з однаковою швидкістю, цикл Do – майже у два рази швидше за них, останній Table на 10-25% швидший за Do. Такий результат обумовлюється тим, що перші два є універсальними: можуть мати довільні значення параметра ітерування, функції його інкременту та умови закінчення. Цикл Do відрізняється від цих двох тим, що має стандартні функції інкременту та умови виходу і тому більш оптимізований. Таблиця, використовуючи неявне компілювання, ще більш оптимізована за оператор Do.

Рекомендується, з вище зазначеного, використовувати за можливості всюди оператор циклу Do, і тільки у випадках, коли потрібні можливості нестандартних операторів інкременту та умови виходу – два останні стандартні. Нестандартне використання оператору Table можна рекомендувати тільки якщо час ЦП є значно важливішою метою ніж економне використання ОП, та при достатній наявності останньої.

### Висновки

У рамках даної статті розроблені та описані методи підвищення обчислювальної ефективності ітераційних алгоритмів у математичному процесорі Wolfram Mathematica, що відносяться до двох категорій: попередні та часу виконання. До перших відноситься перетворення усіх важких або часто затребуваних функцій до табличного або поліноміального виду, винесення супутніх обчислень за межі основного ітеративного циклу програми, дотримання наданих у статті рекомендацій щодо вибору оператора циклу.

До досліджених методів підвищення ефективності обчислення часу виконання відносяться наступні: у процесі роботи програми очищувати усі непотрібні дані та експортувати у файли дампу ті, що потребують повторного завантаження; уникати повторного обчислення змінних та константних для певного набору аргументів функцій шляхом збереження результатів в оперативній або зовнішній пам'яті;

використовувати компілювання функцій у код WVM та C засобами матпакету.

Доведено, що кожний з них здатен істотно впливати на використання таких обмежених ресурсів комп'ютера, як об'єм оперативної пам'яті та час центрального процесора при розрахунках. Наведені приклади використання кожного методу, дані необхідні пояснення.

Дотримання викладених у статті рекомендацій щодо кожного дослідженого методу дозволить підвищити продуктивність ітераційних програм розроблених на базі математичного процесору Wolfram Mathematica.

#### Список літератури:

- [1] *Seyyedi, M., Rostami, A. and Matloub, S.* Comparative study of transparent display using aperiodic arrays of Si-SiO<sub>2</sub> core-shell nanoparticles. // *Optical and Quantum Electronics*. 2020. Vol. 52, No. 479.
- [2] *Syedfaraji, A.* Improving the Raman amplifier bandwidth and gain using multi-micro ring photonic crystal structures // *Optical and Quantum Electronics*. 2020. Vol. 52, No. 478.
- [3] *Mohammadi, K.N.* Design of XOR/NOT gate and power-phase comparator by gold air-gap composition in silicon wiring technology // *Optical and Quantum Electronics*. 2020. Vol. 52, No. 452.
- [4] *Mansuri, M., Mir, A. & Farmani, A.* Numerical analysis of tunable nonlinear plasmonic router based on nanoscale ring resonators // *Optical and Quantum Electronics*. 2020. Vol. 52, No. 448.
- [5] *Liu, D., Hu, S.* Silicon photonic crystal heterojunctions based dual-band unidirectional transmission with high transmissivity // *Optical and Quantum Electronics*. 2020. Vol. 52, No. 431.
- [6] *Mamtaz, R., Ahmed, K., Paul, B.K. et al.* Design and FEM analysis of pentagonal photonic crystal fiber for highly nonlinear applications // *Optical and Quantum Electronics*. 2020. Vol. 52, No. 455.
- [7] *Mahalakshmi, P., Prakash, S.A. & Rajan, M.S.M.* Design of germanium core with anisotropic metamaterial cladding optical fiber in mid-infrared range applications // *Optical and Quantum Electronics*. 2020. Vol. 52, No. 298.
- [8] *Li, J., Wang, J., Teng, Y. et al.* Broadband supercontinuum generation based on filled structural photonic crystal fibers with low incident optical power // *Optical and Quantum Electronics*. 2020. Vol. 52, No. 447.
- [9] *Jiang, L., Zhang, G., Zhang, Y. et al.* A miniaturized and high-bandwidth polarization filter based on a plasmonic and liquid crystal offset core photonic crystal fiber // *Optical and Quantum Electronics*. 2020. Vol. 52, No. 409.
- [10] *Zhou, S.* Twin photonic hooks generated from two adjacent dielectric cylinders // *Optical and Quantum Electronics*. 2020. Vol. 52, No. 389.
- [11] *Alves, R., Hildenbrand, D., Steinmetz, C. et al.* Efficient Development of Competitive Mathematica Solutions Based on Geometric Algebra with GAALOPWeb. // *Advances in Applied Clifford Algebras*. 2020. Vol. 30, No. 59.
- [12] *Athron, P., Balazs, C., Bringmann, T. et al.* GAMBIT: the global and modular beyond-the-standard-model inference tool. // *The European Physical Journal C*. 2018. Vol. 78, No. 98.
- [13] Wolfram Mathematica: Modern technical calculations [Електронний ресурс] // Wolfram. Режим доступу: <https://www.wolfram.com/mathematica/>
- [14] *Zolotariov D.* Extension of the approximation functions method for 2d nonlinear Volterra integral equations / D. Zolotariov, A. Nerukh // «Applied radioelectronics»: Sci.-Tech. journal. 2011. Mag. 10. № 1. P. 39–44.
- [15] 10 Tips for Writing Fast Mathematica Code [Електронний ресурс] // Wolfram. Режим доступу: <https://blog.wolfram.com/2011/12/07/10-tips-for-writing-fast-mathematica-code/>. – Назва з екрана.
- [16] Performance tuning in Wolfram Language [Електронний ресурс] // Wolfram Community. URL: <https://community.wolfram.com/groups/-/m/t/1037730>. – Назва з екрана.
- [17] Optimize Code [Електронний ресурс] // Wolfram Mathematica. URL: <https://www.wolfram.com/mathematica/new-in-10/inactive-objects/optimize-code.html>. – Назва з екрана.
- [18] A Package For Code Optimization Using Mathematica [Електронний ресурс] // Wolfram. URL: <https://library.wolfram.com/infocenter/MathSource/3947/> – Назва з екрана.
- [19] RAM Memory Speeds and Compatibility [Електронний ресурс] // Crucial. Режим доступу: <https://www.crucial.com/support/memory-speeds-compatibility>. – Назва з екрана.
- [20] SSD vs. HDD Speed [Електронний ресурс] // EnterpriseStorageForum. URL: <https://www.enterprisestorageforum.com/storage-hardware/ssd-vs-hdd-speed.html>. – Назва з екрана.
- [21] *Руссинович М., Соломон Л.* Внутреннее устройство Microsoft Windows. 6-е изд. Часть 1. СПб.: Питер, 2013. – 800 с.
- [22] *Золотарьов Д.О.* Автоматизація та оптимізація наукових та інженерних обчислень у Wolfram Mathematica. Харків: ФОП Панов А.М., 2020. – 107 с.
- [23] ClearAll – Wolfram Language Documentation [Електронний ресурс] // Wolfram. URL: <https://reference.wolfram.com/language/ref/ClearAll.html>. – Назва з екрана.
- [24] Export – Wolfram Language Documentation [Електронний ресурс] // Wolfram. URL: <https://reference.wolfram.com/language/ref/Export.html>. – Назва з екрана.
- [25] Import – Wolfram Language Documentation [Електронний ресурс] // Wolfram. URL: <https://reference.wolfram.com/language/ref/Import.html>. – Назва з екрана.
- [26] DumpSave – Wolfram Language Documentation [Електронний ресурс] // Wolfram. URL: <https://reference.wolfram.com/language/ref/DumpSave.html>. – Назва з екрана.
- [27] Get – Wolfram Language Documentation [Електронний ресурс] // Wolfram. URL: <https://reference.wolfram.com/language/ref/Get.html>. – Назва з екрана.
- [28] Compile – Wolfram Language Documentation [Електронний ресурс] // Wolfram. URL: <https://reference.wolfram.com/language/ref/Compile.html>. – Назва з екрана.
- [29] While – Wolfram Language Documentation [Електронний ресурс] // Wolfram. URL: <https://reference.wolfram.com/language/ref/While.html>. – Назва з екрана.
- [30] For – Wolfram Language Documentation [Електронний ресурс] // Wolfram. URL: <https://reference.wolfram.com/language/ref/For.html>. – Назва з екрана.
- [31] Do – Wolfram Language Documentation [Електронний ресурс] // Wolfram. URL: <https://reference.wolfram.com/language/ref/Do.html>. – Назва з екрана.
- [32] Table – Wolfram Language Documentation [Електронний ресурс] // Wolfram. URL: <https://reference.wolfram.com/language/ref/Table.html>. – Назва з екрана.
- [33] Sum – Wolfram Language Documentation [Електронний ресурс] // Wolfram. URL: <https://reference.wolfram.com/language/ref/Sum.html>. – Назва з екрана.



А.П. Сафоник<sup>1</sup>, О.П. Лакус<sup>2</sup>, І.М. Гришок<sup>3</sup>

<sup>1</sup>НУВГП, м. Рівне, Україна, a.p.safonyk@nuwm.edu.ua;

<sup>2</sup>НУВГП, м. Рівне, Україна, lakus\_ak15@nuwm.edu.ua;

<sup>3</sup>НУВГП, м. Рівне, Україна, i.m.hrytsiuk@nuwm.edu.ua

## РОЗРОБКА ТА ДОСЛІДЖЕННЯ ЕНЕРГОЗБЕРІГАЮЧОЇ СИСТЕМИ ОЧИЩЕННЯ ТЕХНОЛОГІЧНИХ ВОД ВІД ФЕРОМАГНІТНИХ ДОМІШОК ТА ЇЇ АВТОМАТИЗАЦІЯ

Побудовано математичну модель процесу очищення технологічних вод. Розроблено структурну схему зв'язків параметрів процесу вилучення феромагнітних домішок, що дало змогу знайти перехідні характеристики, провести комп'ютерний експеримент та в подальшому визначити основні конструктивні параметри процесу. Проведено дослідження відповідного процесу з використанням знайдених передаточних функцій об'єкта по каналах керування. Розроблено П-регулятор для енергоефективного керування концентрацією забруднення на виході з системи очищення. На основі експериментальних даних та розроблених перехідних характеристиках показано, що економія електроенергії з використанням запропонованого регулятора становить як мінімум 15%.

АВТОМАТИЗОВАНЕ КЕРУВАННЯ, МАГНІТНА ОЧИСТКА, ТЕХНОЛОГІЧНІ ВОДИ, ПЕРЕХІДНА ХАРАКТЕРИСТИКА, П-РЕГУЛЯТОР, ЕНЕРГОЕФЕКТИВНІСТЬ

**Сафоник А.П., Лакус П.О., Гришок І.М. Разработка и исследование энергосберегающей системы очистки технологических вод от ферромагнитных примесей и ее автоматизация.** Построена математическая модель процесса очистки технологических вод. Разработана структурная схема связей параметров процесса извлечения ферромагнитных примесей, что позволило найти переходные характеристики, провести компьютерный эксперимент и в дальнейшем определить основные конструктивные параметры процесса. Проведено исследование соответствующего процесса с использованием найденных передаточных функций объекта по каналам управления. Разработан П-регулятор для энергоэффективного управления концентрацией загрязнения на выходе из системы очистки. На основе экспериментальных данных и разработанных переходных характеристиках показано, что экономия электроэнергии с использованием предложенного регулятора составляет как минимум 15%.

АВТОМАТИЗИРОВАННОЕ УПРАВЛЕНИЕ, МАГНИТНАЯ ОЧИСТКА, ТЕХНОЛОГИЧЕСКИЕ ВОДЫ, ПЕРЕХОДНАЯ ХАРАКТЕРИСТИКА, П-РЕГУЛЯТОР, ЭНЕРГОЭФФЕКТИВНОСТЬ

**Safonyk A.P., Lakus P.O., Hrytsiuk I.M. Development and research of the energy saving system of purification technological waters from ferromagnetic impurities and its automation.** A mathematical model of the industrial water treatment process is built. The structural scheme of relationships of parameters of the process of extraction of ferromagnetic impurity is developed that has allowed to find the transitional characteristics, to carry out computer experiments and further to define the basic constructive parameters of the process. A study of the corresponding process was performed using the found transfer functions of the object through the control channels. A P-regulator has been developed for energy-efficient control of pollution concentration at the outlet of the treatment system. Based on experimental data and the developed transient characteristics, it is shown that the energy savings using the proposed regulator is at least 15%.

AUTOMATED CONTROL, MAGNETIC PURIFICATION, TECHNOLOGICAL WATER, TRANSIENT CHARACTERISTIC, P-REGULATOR, ENERGY EFFICIENCY

### Вступ

Вода є основою для існування людства, а забруднення гальмують або повністю виключають повноцінний розвиток інфраструктур країн. Прісна вода необхідна для отримання як і питної так і технологічної води. Технічна вода використовується в різноманітних галузях промисловості, сільського господарства, енергетики та інші.

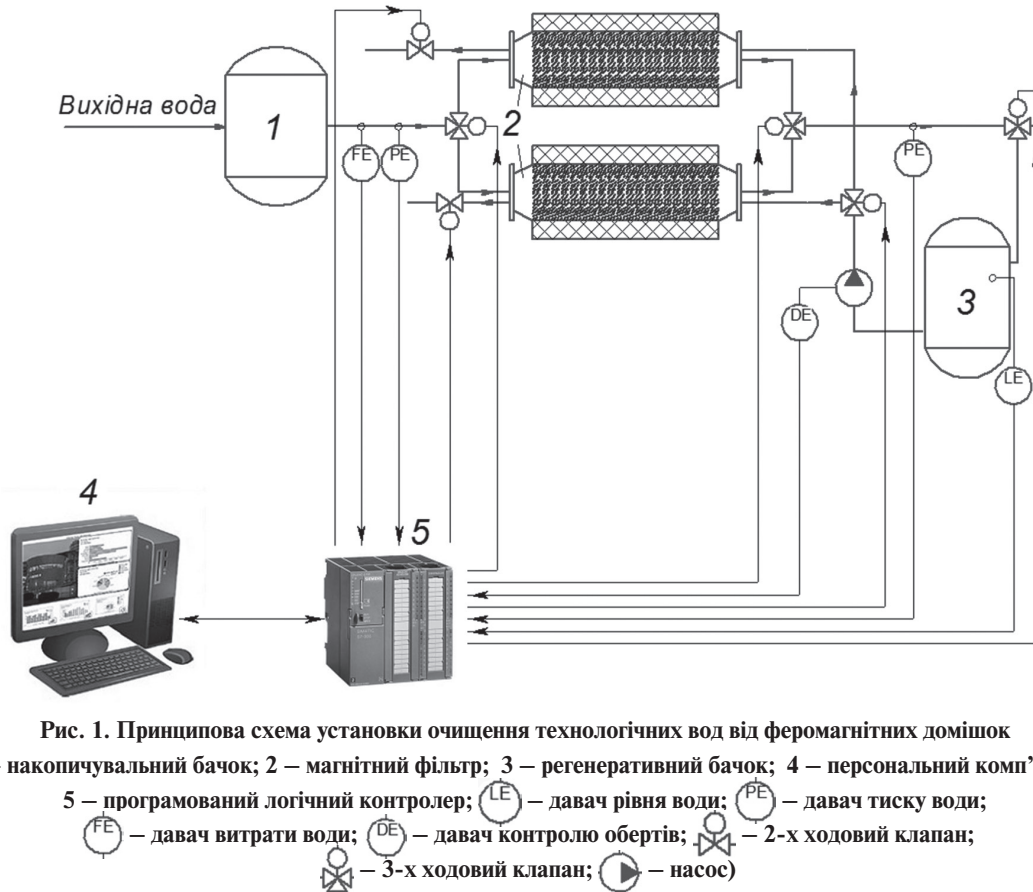
В останні часи ситуація з водами надзвичайно складна. Основними проблемами є:

- помітне скорочення запасів води в результаті збільшення кількості населення, розширення промисловості, розвитку сільського господарства;
- проблеми виробництв, які пов'язані з застарілими технологіями та інфраструктурами;
- погіршення якості води.

На даний час головними показниками якості технологічної води є ступінь очистки від різного виду домішок, які прискорюють зношення технологічного устаткування, утворення корозії та зниження якості продукту та в подальшому формування феромагнітних фракцій. Для видаленні такого виду домішок запропоновано застосовувати метод магнітного очищення водного середовища. Основною перевагою даного методу є відсутність використання хімічних реагентів, що є екологічно безпечним, а отже очищення такого роду завжди є актуальним.

### 1. Постановка задачі

Для комплексного знезалізнення технологічних вод була розроблена установка (рис. 1), яка дозволяє використовувати ряд параметрів для керування процесом вилучення феромагнітних домішок.



**Рис. 1. Принципова схема установки очищення технологічних вод від феромагнітних домішок**  
 (1 – накопичувальний бак; 2 – магнітний фільтр; 3 – регенеративний бак; 4 – персональний комп’ютер;  
 5 – програмований логічний контролер; LE – давач рівня води; PE – давач тиску води;  
 FE – давач витрати води; DE – давач контролю обертів; 2-х ходовий клапан;  
 3-х ходовий клапан; насос)

Довжина магнітного фільтра, діаметр гранульованих частинок, об’єм фільтруючої заправки є сталими параметрами, тому вони враховуються при розрахунках ефективності вилучення залізовмісних домішок із технологічних вод на стадії проектування. До керування параметрів процесу магнітного очищення стічних вод згідно з розробленою структурною схемою можна віднести: витрата вихідної води, сила струму в колі живлення електромагніта.

Керувати витратою вихідної води для забезпечення заданої концентрації залізовмісних домішок на виході із системи не є доцільним. Тому, що установка матиме змінну продуктивність і для розробки системи автоматизації необхідно використати ряд дороговартісних приладів: витратоміри, перетворювач частоти для насоса подачі забрудненої води. Також система буде досить інерційною через рух рідини по трубопроводах і електромагнітному фільтрі.

Керувати силою струму в колі живлення електромагніта значно простіше. Оскільки від напруженості магнітного поля змінюється здатність фільтруючої заправки притягувати забруднення. А сама напруженість прямопропорційна силі струму в колі живлення електромагніта. Важливим фактором також залишається мінімізація енерговитрат при протіканні процесу магнітного очищення, розробка економічно вигідного проекту для очистки стічних вод.

Розроблена у роботі система автоматизації процесу незалізнення технічної води забезпечує збір та

обробку даних про концентрацію залізовмісних домішок технічної води, що дозволяє зменшити корозію технологічного обладнання. У результаті це дозволить зменшити витрати на ремонт установок, підвищити якість роботи обладнання, зменшити кількість робочого персоналу.

## 2. Огляд літератури

Метод магнітної очистки не один рік є пріоритетним напрямком роботи багатьох науковців. В роботі [1] було досліджено новий підхід для отримання основних даних та складу феромагнітних частинок в процесі магнітної очистки. В свою чергу, в [2-3] автори розробляють автоматизовану систему керування процесом магнітного очищення та займаються підбором комплексу технічних засобів автоматизації, що допомагає в подальшому покращенні та інновації даних комплексів. Дослідження [4] містить експериментально отримані і функціонально описані залежності сили тяжіння ферочастинок в модулі від розміру, відстані та взаємодії, знайдені вирази для граничних значень прикладеної енергії захоплення ферочастинок, швидкості її входження в цю зону і висоти вільного падіння. У дослідженні [5] первинна увага приділяється насамперед флотаційним процесам, включаючи флотацію іонів. Було знайдено рішення для визначення концентрацій як вихідних забруднень, так і залишків після очищення води. Робота [6] описує розробку електромагнітного пристрою очистки води в якому агресивність фільтрату

знижується завдяки багатоступеневому процесу обробки. В [7] проведено моделюванням електромагнітного очищення води за умови швидкого розподілу в досліджуваній області. Для цього було використано метод подвійного потенціалу для моделювання потоків нестисливої в'язкої рідини. У [8-9] описано побудову новітніх системи магнітних фільтрів.

Проте, більшість робіт не враховують взаємний вплив зміни параметрів фільтруючого середовища на якість очистки стічних вод. Також деякі моделі враховують змінні параметри як константи, що значно спрощує обчислювальні процеси, але впливає на отримуваний результат. Майже відсутні роботи присвячені розробці алгоритмів автоматизованого керування технологічними процесами очистки технологічних вод від феромагнітних домішок. Важливими є також розробка математичних моделей, що мають підтвердження експериментальними даними, але і потребують обґрунтування математичним апаратом. Окремою важливою задачею є перевірка універсальності, точності та адекватності розроблених математичних моделей. Отже в теорії процесів фільтрування є ряд важливих задач, що потребують вирішення, зокрема: розробка універсальних математичних моделей, які описують технологічні процеси очистки стічних вод в цілому при змінних параметрах системи; врахуванні факторів, які несуть суттєвий вплив на якість очистки внесенням відповідних коефіцієнтів у залежностях; розробка методів розрахунку очисних споруд з використання комп'ютерного моделювання; розробка систем автоматизованого керування процесами фільтрування.

### 3. Матеріали і методи

Для автоматизації процесу магнітного очищення води розроблено принципову схему автоматизації представлену на рис. 1, згідно якої керування системою відбувається переведенням потоку фільтруючої рідини між двома фільтрами. Під час фільтрування робочим фільтром інший «забруднений» автоматично проводить регенерацію.

На основі проведених досліджень процесу було розроблено структурну схему зв'язків параметрів процесу вилучення феромагнітних домішок:

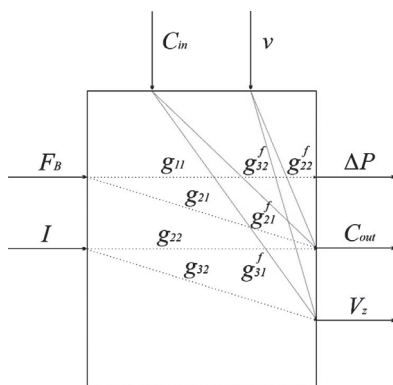


Рис. 2. Структурна схема зв'язків параметрів процесу

На основі структурної схеми зв'язків процесу вхідні і вихідні величини було описано наступною залежністю:

$$y = \begin{bmatrix} g_{11}(p) & 0 & 0 \\ g_{21}(p) & g_{22}(p) & 0 \\ 0 & g_{32}(p) & 0 \end{bmatrix} \cdot u + \begin{bmatrix} 0 & 0 \\ g_{21}^f(p) & g_{22}^f(p) \\ g_{31}^f(p) & g_{32}^f(p) \end{bmatrix} \cdot f, \quad (1)$$

де  $y = [\Delta P \ C_{out} \ V_z]^T$  — вектор вихідних параметрів, відповідно перепад тиску в електромагнітному фільтрі, залишкова концентрація домішок в очищеній воді (експериментальне вимірювання), об'єм осаду забруднювача;  $u = [F_B \ I \ 0]^T$  — вектор керування, відповідно витрата вихідної води і сила струму, яка протікає в колі живлення електромагніта для утворення заданої напруженості магнітного поля;  $f = [C_{in} \ v]^T$  — вектор неконтрольованих збурень, відповідно загальна концентрація забруднень в технологічній воді і швидкість руху рідини

### 4. Експерименти

Для визначення передаточних функцій  $g_{11}(p)$  об'єкта по каналу «витрата — перепад тиску»,  $g_{22}(p)$  об'єкта по каналу «сила струму — концентрація домішок на виході»,  $g_{21}^f(p)$  об'єкта по каналу «загальна концентрація забруднень в стічній воді-концентрація домішок на виході»,  $g_{22}^f(p)$  об'єкта по каналу «швидкість руху рідини - концентрація домішок на виході» було проведено комп'ютерний експеримент на основі розробленої в [16] математичної моделі.

При побудові перехідної характеристики об'єкта по каналу «витрата - перепад тиску», використовувалися наступні дані:  $H = 40 \text{ кА/м}$ ,  $c_{in} = 2 \text{ мг/дм}^3$ , характеристики об'єкта керування на відповідні впливи наведено на рис. 3.

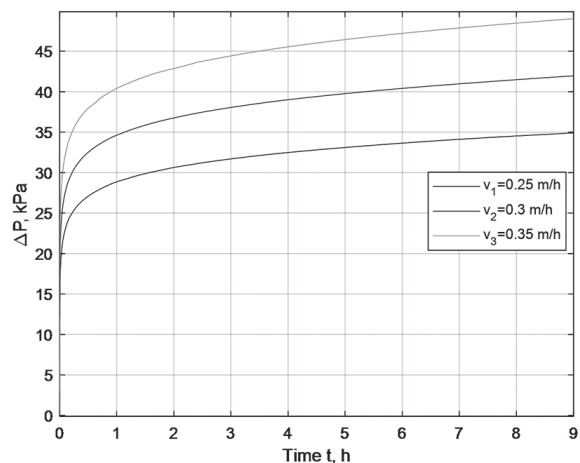


Рис. 3. Динаміка зміни втрати напору на виході магнітного фільтра при ступінчастій зміні вхідної швидкості

Для побудови перехідної характеристики об'єкта по каналу «сила струму - концентрація домішок на виході», використовувалися наступні дані:  $c_{in} = 2 \text{ мг/дм}^3$ ,  $v = 0.3 \text{ м/год}$ ,  $H_1 = 40 \text{ кА/м}$ ,  $H_2 = 60 \text{ кА/м}$ ,

$H_3 = 80$  кА/м. При експериментальних дослідженнях легше вимірювати та керувати силою струму, проте ефективність захоплення частинок фільтруючою загрузкою залежить від напруженості магнітного поля. Оскільки вони між собою прямо пропорційні згідно 1 закону Максвелла, тому згідно модельної задачі змінювали напруженість магнітного поля. Характеристики об'єкта управління на відповідні впливи наведено на рис. 4.

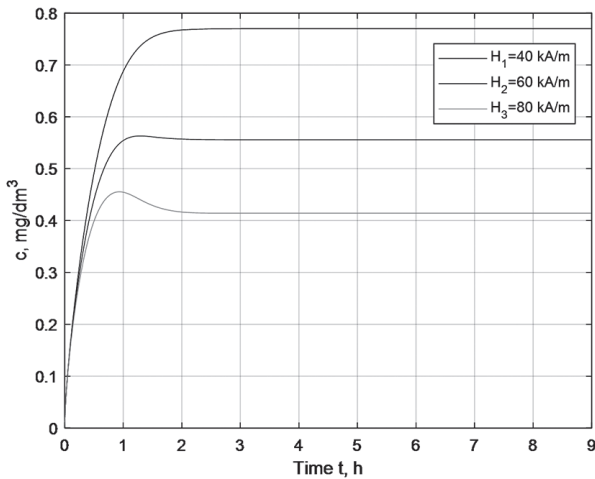


Рис. 4. Динаміка зміни концентрації забруднення на виході магнітного фільтра при ступінчастій зміні напруженості магнітного поля

При побудові перехідної характеристики об'єкта по каналу «загальна концентрація забруднень в стічній воді – концентрація домішок на виході», використовувалися наступні дані:  $H = 40$  кА/м,  $v = 0.3$  м/год,  $c_{in1} = 2$  мг/дм<sup>3</sup>,  $c_{in2} = 4$  мг/дм<sup>3</sup>,  $c_{in3} = 6$  мг/дм<sup>3</sup>, а характеристики об'єкта управління на відповідні впливи наведено на рис. 5.

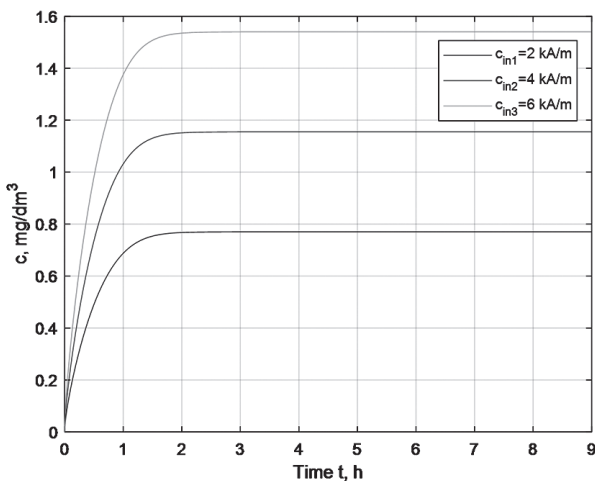


Рис. 5. Динаміка зміни концентрації забруднення на виході магнітного фільтра при ступінчастій зміні вхідної концентрації забруднення

При побудові перехідної характеристики об'єкта по каналу «швидкість руху рідини – концентрація домішок на виході», використовувалися наступні дані:  $H = 40$  кА/м,  $v = 0.3$  м/год,  $c_{in1} = 2$  мг/дм<sup>3</sup>,  $c_{in2} = 4$  мг/дм<sup>3</sup>,  $c_{in3} = 6$  мг/дм<sup>3</sup>, а характеристики об'єкта управління на відповідні впливи наведено на рис. 9.

дані:  $H = 40$  кА/м,  $c_{in} = 2$  мг/дм<sup>3</sup>,  $v_1 = 0.25$  м/год,  $v_2 = 0.3$  м/год,  $v_3 = 0.35$  м/год, а характеристики об'єкта управління на відповідні впливи наведено на рис. 6.

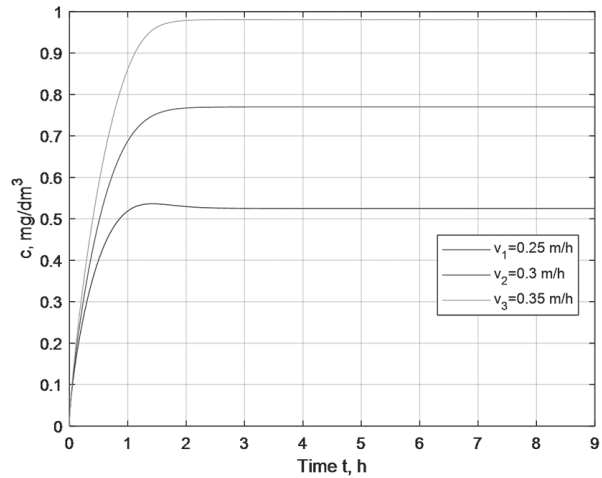


Рис. 6. Динаміка зміни концентрації забруднення на виході магнітного фільтра при ступінчастій зміні вхідної швидкості

При побудові перехідної характеристики об'єкта по каналу «сила струму – об'єм осаду забруднювача», використовувалися наступні дані:  $c_{in} = 2$  мг/дм<sup>3</sup>,  $v = 0.3$  м/год,  $H_1 = 40$  кА/м,  $H_2 = 60$  кА/м,  $H_3 = 80$  кА/м, а характеристики об'єкта управління на відповідні впливи наведено на рис. 7.

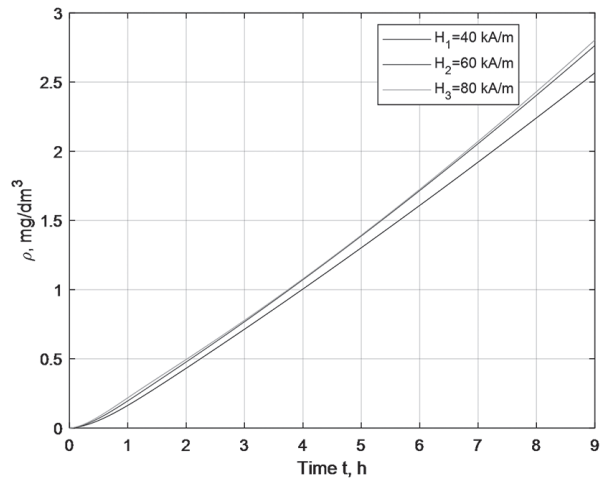


Рис. 7. Динаміка зміни концентрації осаду, що захопився у фільтрі при ступінчастій зміні напруженості магнітного поля

При побудові перехідної характеристики об'єкта по каналу «швидкість руху рідини – об'єм осаду забруднювача», використовувалися наступні дані:  $H = 40$  кА/м,  $c_{in} = 2$  мг/дм<sup>3</sup>,  $v_1 = 0.25$  м/год,  $v_2 = 0.3$  м/год,  $v_3 = 0.35$  м/год, а характеристики об'єкта управління на відповідні впливи наведено на рис. 8.

При побудові перехідної характеристики об'єкта по каналу «загальна концентрація забруднень в стічній воді – об'єм осаду забруднювача», використовувалися наступні дані:  $H = 40$  кА/м,  $v = 0.3$  м/год,  $c_{in1} = 2$  мг/дм<sup>3</sup>,  $c_{in2} = 4$  мг/дм<sup>3</sup>,  $c_{in3} = 6$  мг/дм<sup>3</sup>, а характеристики об'єкта управління на відповідні впливи наведено на рис. 9.

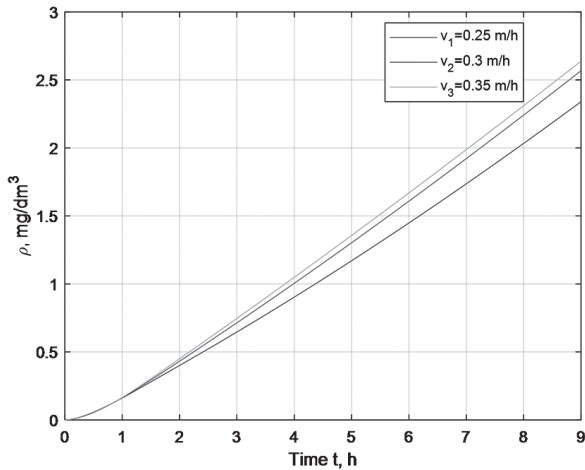


Рис. 8. Динаміка зміни концентрації осаду, що захопився у фільтрі при ступінчастій зміні вхідної швидкості

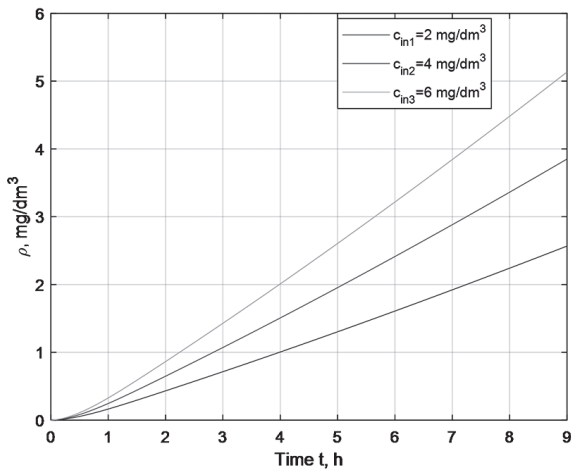


Рис. 9. Динаміка зміни концентрації осаду, що захопився у фільтрі при ступінчастій зміні вхідної концентрації забруднення

Згідно з результатами представленими на рис. 3 – рис. 9 було визначено передаточні функції  $g_{22}(p)$  об'єкта по каналу «сила струму - концентрація домішок на виході»,  $g_{21}^f(p)$  об'єкта по каналу «загальна концентрація забруднень в стічній воді-концентрація домішок на виході»,  $g_{22}^f(p)$  об'єкта по каналу «загальна концентрація забруднень в стічній воді-концентрація домішок на виході» описуються аперіодичною ланкою 1 порядку, а  $g_{11}(p)$  об'єкта по каналу «витрата-перепад тиску»,  $g_{32}(p)$  об'єкта по каналу «сила струму – об'єм осаду забруднювача»,  $g_{31}^f(p)$  об'єкта по каналу «загальна концентрація забруднень в стічній воді – об'єм осаду забруднювача» та  $g_{32}^f(p)$  об'єкта по каналу «швидкість руху рідини – об'єм осаду забруднювача» – реальною інтегруючою ланкою:

$$g_{11}(p) = \frac{k}{p(Tp+1)}, \quad (2)$$

$$g_{22}(p) = \frac{k_{22}}{T_{22}p+1}, \quad (3)$$

$$g_{32}(p) = \frac{k_4}{p(T_4p+1)}, \quad (4)$$

$$g_{21}^f(p) = \frac{k_{21}}{T_{21}p+1}, \quad (5)$$

$$g_{22}^f(p) = \frac{k_3}{T_3p+1}, \quad (6)$$

$$g_{31}^f(p) = \frac{k_5}{p(T_5p+1)}, \quad (7)$$

$$g_{32}^f(p) = \frac{k_6}{p(T_6p+1)}, \quad (8)$$

Дослідження показали, що об'єкт по каналах  $g_{11}(p)$ ,  $g_{22}(p)$ ,  $g_{21}^f(p)$ ,  $g_{22}^f(p)$ ,  $g_{32}(p)$ ,  $g_{31}^f(p)$ ,  $g_{32}^f(p)$  має змінні параметри, що залежать від режиму роботи установки (таблиця 1).

Таблиця 1

Параметри передаточних функцій на за відповідними каналами

Параметр	Експериментальні дані		
	0→0.25 м/хв	0→0.3 м/хв	0→0.35 м/хв
$k$	0.17	0.26	0.34
$T$	1.7	2.0	2.2
$k_3$	0.52	0.77	0.98
$T_3$	1.4	1.1	0.9
$k_6$	1.43	1.45	1.45
$T_6$	0.2	0.3	0.4
Параметр	Експериментальні дані		
	0→40 кА/м	0→60 кА/м	0→80 кА/м
$k_{22}$	0.77	0.56	0.41
$T_{22}$	0.49	0.33	0.22
$k_4$	1.45	1.46	1.46
$T_4$	0.4	0.2	0.15
Параметр	Експериментальні дані		
	0→2 мг/дм <sup>3</sup>	0→4 мг/дм <sup>3</sup>	0→6 мг/дм <sup>3</sup>
$k_{21}$	0.77	1.16	1.54
$T_{21}$	1.9	1.1	0.8
$k_5$	1.36	1.42	1.44
$T_5$	0.3	0.4	0.5

Експериментальними дослідженнями було встановлено, що на процес знезалізнення технічних вод витрата рідини не вносить істотного впливу, тому можна прийняти  $g_{21} \approx 0$ .

Таким чином, залежність (1) з урахуванням (2) - (8) має вигляд:

$$y = \begin{bmatrix} \frac{k}{p(Tp+1)} & 0 \\ 0 & \frac{k_{22}}{T_{22}p+1} \\ 0 & \frac{k_4}{p(T_4p+1)} \end{bmatrix} \cdot \begin{bmatrix} F_B \\ I \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{k_{21}}{T_{21}p+1} & \frac{k_3}{T_3p+1} \\ \frac{k_5}{p(T_5p+1)} & \frac{k_6}{p(T_6p+1)} \end{bmatrix} \cdot \begin{bmatrix} C_{in} \\ v \end{bmatrix}, \quad (9)$$

На основі отриманої передаточної функції  $g_{22}(p)$  об'єкта по каналу «сила струму – концентрація

домішок на виході» (3) проведено дослідження впливу величини струму на концентрацію феромагнітних домішок на виході. В якості вихідних даних використовували задану концентрацію домішок у технологічних водах, яка не повинна перевищувати  $0.5 \text{ мг/дм}^3$ . Отже при моделюванні керувалися наступним принципом: протягом 2 години система має ефективно чистити технологічні води і концентрація забруднення не має перевищувати  $0.5 \text{ мг/дм}^3$ . При зміні концентрації забруднення на вході отримували значення величини струму, яку необхідно прикласти, а результати наведено в таблиці 2, рис. 10.

Таблиця 2

Залежність величини струму від вхідної концентрації феромагнітних домішок

№	Концентрація вхідного забруднення, $\text{мг/дм}^3$	Величина струму, $A$
1	1.4	8.88
2	1.6	10.5
3	1.8	12
4	2	13.4
5	2.2	14.7
6	2.4	15.9
7	2.6	17.07

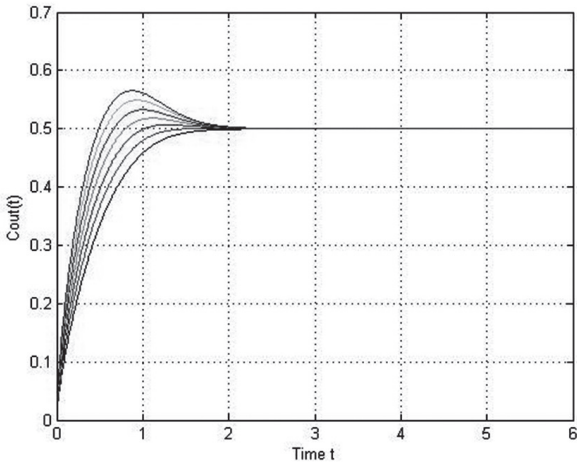


Рис. 10. Динаміка зміни концентрації феромагнітних домішок на виході при  $I=[8.88 \ 10.5 \ 12 \ 13.4 \ 14.7 \ 15.9 \ 17.07]$ .

### 5. Експерименти

З допомогою програми Matlab проведено апроксимацію отриманих даних для отримання коефіцієнтів поліноміального ряду. При цьому використовували функцію polyfit та написали наступний програмний код:

```
>> C=[1.4 1.6 1.8 2 2.2 2.4 2.6];
>> I=[8.88 10.5 12 13.4 14.7 15.9 17.07];
>> polyfit(C,I,1)
ans = 6.7982 -0.3893
```

В результаті отримали залежність зміни величини струму від вхідної концентрації забруднення, що представлено залежністю (9) та виступає в системі як П-регулятор

$$I = 6.7982 \cdot C_{in} - 0.3893 \quad (9)$$

Для опису магнітного фільтра, як об'єкта автоматизації за каналом «сила струму — концентрація домішок на виході» необхідно визначити залежності, які описують зміну коефіцієнта передачі і сталої часу від зміни сили струму. Для цього в таблиці 3 наведено результати проведеного експерименту з визначеними відповідними коефіцієнтами.

Таблиця 3

Залежність зміни коефіцієнта передачі і сталої часу передаточної функції об'єкта по каналу  $g_{22}(p)$  «сила струму — концентрація домішок на виході» від прикладеної сили струму

Параметр	Експериментальні дані		
	0→8 A	0→12 A	0→16 A
$k_{22}$	0.77	0.56	0.41
$T_{22}$	0.49	0.33	0.22

А також проведено апроксимацію отриманих даних для отримання коефіцієнтів відповідного поліноміального ряду з використанням функції polyfit програмного середовища Matlab.

```
>> I=[8 12 16];
>> k22=[0.77 0.56 0.41];
>> polyfit(I,k22,1)
ans = -0.0450 1.1200
>> T22=[0.49 0.33 0.22];
>> polyfit(I,T22,1)
ans = -0.0338 0.7517
```

В результаті отримали залежність зміни коефіцієнта передачі (10) та сталої часу (11) від величини струму:

$$k_{22} = -0.045 \cdot I + 1.12 \quad (10)$$

$$T_{22} = -0.0338 \cdot I + 0.7517 \quad (11)$$

Для перевірки роботи регулятора розроблено комп'ютерну модель, що представлена на рис 11, яка складається з двох субсистем «Р» рис. 12, «Magnetic Filter» рис. 13.

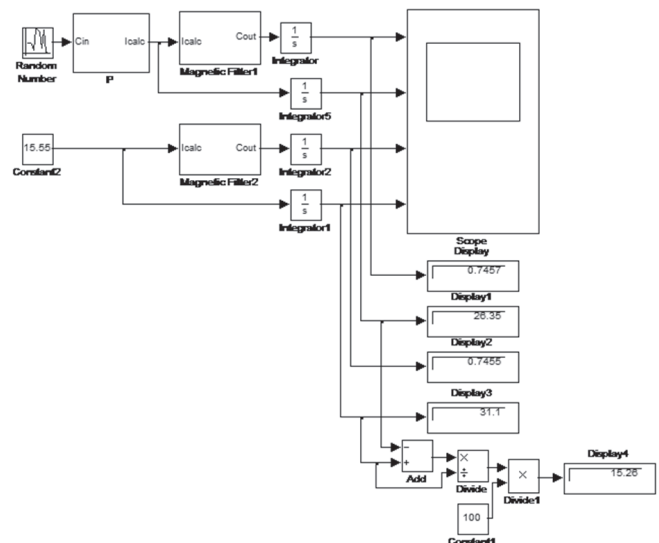


Рис. 11. Комп'ютерна модель системи очищення від феромагнітних домішок

В підсистемі «Р» реалізовано залежність (9), яка оптимізує величину струму, що протікає в колі живлення електромагніта та дозволяє економити витрати електроенергії.

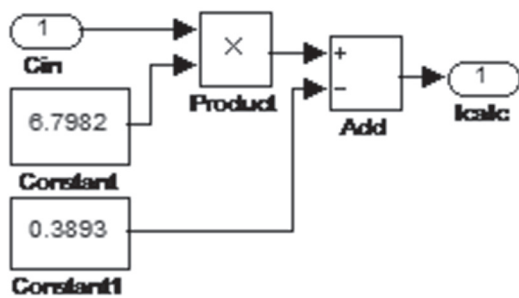


Рис. 12. Комп'ютерна модель регулятора

В підсистемі «Magnetic Filter» реалізовано передаточну функцію (3) об'єкта по каналу «сила струму — концентрація домішок на виході» із змінними коефіцієнтом передачі (10) та сталою часу (11).

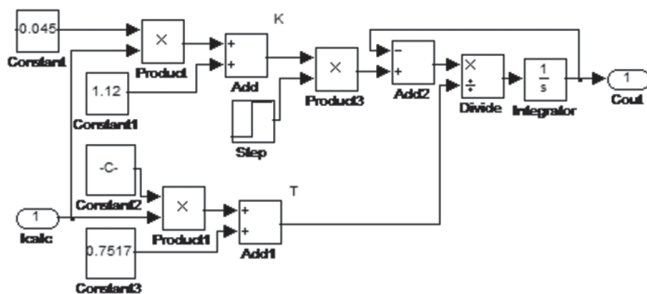


Рис. 13. Комп'ютерна модель «Magnetic Filter»

Після проведення моделювання отримано результати які представлено на рис. 14, рис. 15, де в якості вхідної концентрації забруднення використано випадковий сигнал, що формується блоком «Random number» (рис. 11), що відповідає значенню вхідної концентрації в заданому діапазоні 2,0 мг/дм<sup>3</sup> для П-регулятора.

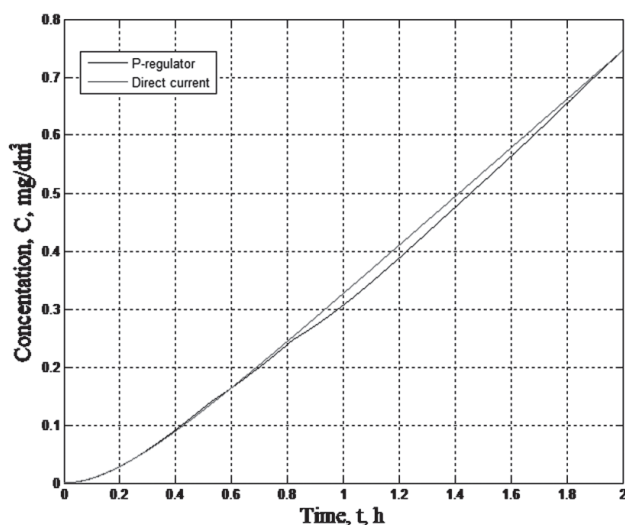


Рис. 14. Динаміка зміни концентрації забруднення на виході із використанням постійного струму, П-регулятора

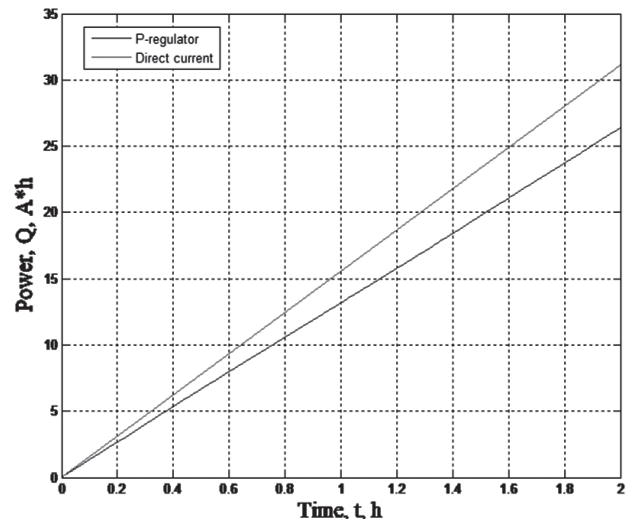


Рис. 15. Динаміка зміни споживаної потужності із використанням постійного струму, П-регулятора

Як видно на рис. 14 концентрація забруднення з часом на виході за 2 години вийде на усталене значення 0,75 мг/дм<sup>3</sup> при використанні сталою струму 15,55 А і вхідній концентрації 2,0 мг/дм<sup>3</sup>, а також при використанні П-регулятора і випадково змінній вхідній концентрації з усередненим значенням 2,0 мг/дм<sup>3</sup>, що відповідає реальним системам. При цьому показники споживаної енергії становитимуть 31,1 А·год та 26,35 А·год відповідно, як показано на рис 15. Тобто використання П-регулятора дозволяє економити 15% споживаної електроенергії за умови, що при розрахунку використовувався ідеальний випадок, де стала вхідна концентрація забруднення і струм в колі живлення максимально точно підібраний. Але як відомо в реальних системах дані умови не виконуються і ефективність економії використання П-регулятора відповідно росте.

## 6. Обговорення

В ході роботи було проведено комп'ютерний експеримент (рис. 3 – рис. 9), за допомогою якого визначено передаточні функції об'єкта по каналу «сила струму — концентрація домішок на виході», об'єкта по каналу «загальна концентрація забруднень в стічній воді-концентрація домішок на виході», об'єкта по каналу «загальна концентрація забруднень в стічній воді-концентрація домішок на виході» описуються аперіодичною ланкою 1 порядку, а об'єкта по каналу «витрата-перепад тиску», об'єкта по каналу «сила струму — об'єм осаду забруднювача», об'єкта по каналу «загальна концентрація забруднень в стічній воді — об'єм осаду забруднювача» та об'єкта по каналу «швидкість руху рідини — об'єм осаду забруднювача» - реальною інтегруючою ланкою. В результаті дослідження визначено, що об'єкт по вище зазначених каналах має змінні параметри, що залежать від режиму роботи установки та продемонстровано

в таблиці 1 та встановлено, що на процес витрата рідини не вносить істотний вплив.

Проведено моделювання на основі передаточної функції (3) у якому протягом двох годин система має ефективно чистити технологічні води і концентрація забруднення не має перевищувати  $0.5 \text{ мг/дм}^3$ . Змінюючи концентрацію забруднення на вході отримали значення наведені в таблиці 2 та на візуально продемонстрували на рис. 10.

Для перевірки роботи регулятора розроблено комп'ютерну модель, що представлена на рис. 11, яка складається з двох субсистем «Р» рис. 12, «Magnetic Filter» рис. 13. Після проведення моделювання отримано результати які представлено на рис. 14, рис. 15, де в якості вхідної концентрації забруднення використано випадковий сигнал.

### Висновки

Розглянуто процес очищення технологічних вод від залізовмісних домішок. Побудовано структуру схему об'єкта управління та визначено керуючі і збурюючі параметри. Проведено комп'ютерне моделювання перехідних процесів за каналами «швидкість руху рідини - концентрація домішок на виході», «сила струму - концентрація домішок на виході», «загальна концентрація забруднень в стічній воді-концентрація домішок на виході». Знайдено математичну модель у вигляді передавальних функцій по кожному каналу управління. Розроблено П-регулятор для енергоефективного керування концентрацією забруднення на виході з системи. Показано, що мінімальна економія електроенергії з використанням запропонованого регулятора становить як мінімум 15%. В перспективі є розробка автоматизованої системи керування відповідного процесу очищення технологічних вод від феромагнітних домішок.

### Список літератури:

- [1] Сандуляк А.В., Сандуляк Д.А., Полисмакова М.Н., Киселев Д.О., Сандуляк А.А. О проблеме изучения магнитных свойств железистых примесей технологических сред. IV Международная научно-практическая конференция «Новые технологии и проблемы технических наук», с. 77-85, 2017.
- [2] Сандуляк А.В., Сандуляк А.В., Гаращенко В.И. Электромагнитные фильтр-осадители: монография [Текст]. Львов: Выща школа, 70 с, 1982.
- [3] Сандуляк, А. В. Особенности магнитного фильтрования воды в циркуляционных системах электростанций. Изв. вузов СССР. Сер. Энергетика. № 3, с. 90-93, 1983
- [4] Сандуляк А.А., Сандуляк А.В., Ершова В.А. Экспериментально-расчетное определение силовых и энергетических характеристик рабочих зон магнитных сепараторов. Метрология: Всероссийский научно-исследовательский институт метрологической службы, №2, с. 32-39, 2012.
- [5] B. Ksenofontov. Models of complex process of water purification. Second international conference on material science, smart structures and applications, 2019. <https://doi.org/10.1063/1.5140154>
- [6] O. Terentiev, K. Tkachuk, O. Tverda, A. Kleshchov. Electromagnetic focusing of impurities in water purification, Vol 4, No 10(82), pp. 10-15, 2016. <https://doi.org/10.15587/1729-4061.2016.75251>.
- [7] N. Tarasov, S. Polyakov, T. Kudryashova. Computation of 3D Water Flows by the Double Potential Method for the Simulation of Electromagnetic Water Purification. The European Physical Journal Conferences. Vol. 226, 2020. <https://doi.org/10.1051/epjconf/202022602021>
- [8] Lin Li, Paul S. Greenberg, Kenneth W. Street Jr., Da Ren Chen. Study of a Magnetic Filter System for the Characterization of Particle Magnetic Property. Journal Aerosol Science and Technology, Vol. 45, Issue 3, pp. 327-335, 2011. <https://doi.org/10.1080/02786826.2010.534514>.
- [9] Roger M. Simonson. Magnetic filter and magnetic filtering assembly. Patent Application Publication. Pub. No.: US 2010/0294706 A1, Pub. Date: Nov. 25, 2010.
- [10] Steve McAllorum, Kevin Martin, Gary Liptrot. Magnetic filtration device. Patent Application Publication. Patent No.: US 9,669,414 B2, Date of Patent: Jun. 6, 2017.
- [11] Andrey Ya. Bomba. Identification of Mass Transfer Distribution Factor and Its Account for Magnetic Filtration Process Modeling / Andrey Ya. Bomba, Andrey P. Safonyk, Elena A. Fursachik // Journal of Automation and Information Sciences. – Volume 45, 2013 Issue 4. – pp. 16-22.
- [12] А.П. Сафоник, І.М. Таргоній, В.А. Пасічник. Математичний аналіз параметрів і моделі процесу магнітного осадження домішок. Вісник Інженерної академії України, №1, с. 105–110, 2018.
- [13] А.П. Сафоник. Комп'ютерне моделювання та автоматизація процесу фільтрування та регенерації в магнітному фільтрі. Вісник Інженерної академії України, №1, с. 202–208, 2015.
- [14] А.П. Сафоник, С.О Куницький. Модель процесу контактного знезалізнення води крізь пористе завантаження водоочисного фільтра. Науковий вісник будівництва, Т. 90, №4, с. 188-193, 2017.
- [15] A. Safonyk. Modelling and automation of process of water purification by magnetic filter. Informatyka Automatyka Pomiaru w Gospodarce i Ochronie Środowiska 1/2016, Poland, pp. 32-36, 2016.
- [16] А.П. Сафоник, І.М. Таргоній. Комп'ютерне моделювання та автоматизація процесу магнітного очищення води. Бионика интеллекта: науч.-техн. Журнал, №2(83), с. 117-122, 2014.

Надійшла до редколегії 28.10.2020

УДК 004.8:004.032.26

DOI 10.30837/bi.2020.2(95).10

Є.В. Бодяньський<sup>1</sup>, А.Ю. Шафроненко<sup>2</sup>, І.М. Климова<sup>3</sup><sup>1</sup>Доктор технічних наук, професор кафедри Штучного інтелекту.Харківський національний університет радіоелектроніки  
yevgeniy.bodyanskiy@nure.ua, ORCID 0000-0001-5418-2143<sup>2</sup>Кандидат технічних наук, доцент кафедри Інформатики.  
Харківський національний університет радіоелектроніки  
alina.shafronenko@nure.ua, ORCID 0000-0002-8040-0279<sup>3</sup>Асистент кафедри Системотехніки.Харківський національний університет радіоелектроніки  
iryna.klymova@nure.ua, ORCID 0000-0003-0455-6180

## РЕКУРЕНТНА ДОСТОВІРНА НЕЧІТКА КЛАСТЕРИЗАЦІЯ ВЕЛИКИХ ДАНИХ З ВИКОРИСТАННЯМ ФУНКЦІЇ НАЛЕЖНОСТІ СПЕЦІАЛЬНОГО ТИПУ

Запропоновано метод достовірної нечіткої кластеризації для задач, коли дані надходять на обробку або у послідовному онлайн режимі, або формують надвеликі масиви (Big Data). Введені процедури є за суттю градієнтними алгоритмами оптимізації цільової функції спеціального виду, та мають низку переваг перед відомими ймовірнісними та можливісними підходами і, перш за все, робастність до аномальних спостережень. В основі підходу лежить використання міри подібності, параметри якої визначаються автоматично у процесі самонавчання. Запропоновані процедури є узагальненням відомих методів, характеризуються високою швидкістю та є досить простими у чисельній реалізації.

НЕЧІТКА КЛАСТЕРИЗАЦІЯ, ДОСТОВІРНА НЕЧІТКА КЛАСТЕРИЗАЦІЯ, РІВЕНЬ НАЛЕЖНОСТІ, РІВЕНЬ ДОСТОВІРНОСТІ, МІРА ПОДІБНОСТІ

**Бодянский Е.В., Шафроненко А.Ю., Климова И.М. Рекуррентная достоверная нечеткая кластеризация больших данных с использованием функции принадлежности специального типа.** Предложен метод достоверной нечеткой кластеризации для задач, когда данные поступают на обработку или в последовательном онлайн режиме, или формируют большие массивы (Big Data). Введенные процедуры являются по сути градиентными алгоритмами оптимизации целевой функции специального вида, и имеют ряд преимуществ перед известными вероятностными и возможностными подходами и, прежде всего, робастность к аномальным наблюдениям. В основе подхода лежит использование меры сходства, параметры которых определяются автоматически в процессе самообучения. Предложенные процедуры являются обобщением более известных методов, характеризуются высокой скоростью и являются достаточно простыми в численной реализации.

НЕЧЕТКАЯ КЛАСТЕРИЗАЦИЯ, ДОСТОВЕРНАЯ НЕЧЕТКАЯ КЛАСТЕРИЗАЦИЯ, УРОВЕНЬ ПРИНАДЛЕЖНОСТИ, УРОВЕНЬ ДОСТОВЕРНОСТИ, МЕРА ПОДОБИЯ

**Bodyanskiy Ye.V., Shafronenko A.Yu., Klymova I. M. Recurrent authenticity of the clustering of great tribute to the function of special type.** A method of credibilistic fuzzy clustering is proposed for problems when data are fed sequentially, in online mode and forms large arrays (Big Data). The introduced procedures are essentially gradient algorithms for optimizing the objective function of a special type, and have a number of advantages over known probabilistic and possible approaches and, above all, robustness to anomalous observations. The approach is based on similarity measure, parameters of that are determined automatically in the process of self-learning. The proposed procedures are a generalization of the known methods, characterized by high speed and simple in numerical implementation.

FUZZY CLUSTERING, CREDIBILISTIC FUZZY CLUSTERING, MEMBERSHIP LEVEL, CREDIBILISTIC LEVEL, SIMILARITY MEASURE

### Вступ

Задача кластеризації (класифікації в режимі самонавчання) багатовимірних даних є важливою частиною інтелектуального аналізу даних (Data Mining), в рамках якої склався ряд напрямків і підходів [1, 2]. Один з таких напрямків утворюють методи нечіткої (фаззі-) кластеризації, в основі яких лежить припущення про те, що класи — кластери, що формуються взаємно перетинаються так, що кожне спостереження — вектор з різними рівнями належності — ймовірності — можливості може належати одночасно до кількох чи всіх класів.

Тут найбільш широкого поширення набули алгоритми ймовірнісної нечіткої кластеризації і, перш за все, метод нечітких С-середніх (FCM) [3,4]. Можливості цього підходу обмежуються ймовірнісними обмеженнями на рівні належності так, що «забруднені» збуреннями і викидами спостереження можуть бути віднесені до різних класів з практично однаковими рівнями належності.

У зв'язку з цим в [5] був запропонований можливісний підхід до нечіткої кластеризації (PCM) більш стійкий до шумів і збурень. Разом з тим PCM-алгоритми страждають від, так званої, проблеми

співпадіння, коли в процесі обробки інформації деякі кластери починають зливатися один з одним, що в результаті веде до невірної оцінки кількості цих кластерів.

Цих недоліків позбавлені алгоритми достовірної нечіткої кластеризації [6-8], засновані на апараті теорії вірогідності [9]. В рамках цього підходу в процесі розрахунків оцінюються як рівні нечіткої належності, так і рівні довіри, засновані на мірі належності спеціального вигляду [10]. Результати експериментів показали [7,8], що достовірний підхід забезпечує більш високу якість кластеризації в порівнянні з ймовірнісними і можливісними методами.

Вихідною інформацією для рішення задачі нечіткої кластеризації є масив  $n$ -вимірних векторів спостережень  $X = \{x_1, x_2, \dots, x_N\} \subset R^n$ ,  $x(k) \in X$ ,  $k = 1, 2, \dots, N$ , який повинен бути розбитий на  $m$  кластерів-кластерів з деяким рівнем належності-можливості-достовірності  $U_q(k)$   $k$ -го вектора  $x_k$  до  $q$ -го кластеру ( $1 < m < N, 1 \leq q \leq m$ ). Необхідно також відзначити, що вихідні дані попередньо предоброблені так, що  $-1 \leq x_{ki} \leq 1$  ( $1 \leq i \leq n$ ), де  $x_{ki}$  —  $i$ -та компонента вектора  $x_{ki}$ .

Таким чином, задача кластеризації вирішується в пакетному режимі, коли весь масив даних обробляється багаторазово на основі почергового оцінювання [8].

Якщо ж дані надходять на обробку у вигляді потоку або утворюють надвеликі масиви, що досить часто зустрічаються у реальних ситуаціях, пакетний режим не дозволяє ефективно вирішити розглянуту задачу.

У цій ситуації найбільш ефективними є рекурентні процедури нечіткої кластеризації, що дозволяють вирішувати завдання в online режимі і уточнювати отримані рішення по мірі надходження кожного нового спостереження. Так, у [9, 10] були запропоновані рекурентні варіанти FCM, які є за суттю градієнтними процедурами оптимізації прийнятої цільової функції, а в [11, 12] були введені рекурентні модифікації РСМ, що призначені для послідовної обробки даних.

У зв'язку з цим є доцільною розробка рекурентної модифікації методу достовірної нечіткої кластеризації, що дозволяє уточнювати шукані характеристики кластерів по мірі надходження кожного нового спостереження.

### 1. Рекурентний алгоритм достовірної нечіткої кластеризації

Найбільш популярний метод ймовірнісної нечіткої кластеризації пов'язаний з мінімізацією цільової функції [4]

$$E(U_q(k), w_q) = \sum_{k=1}^N \sum_{q=1}^m U_q^\beta(k) D^2(x_k, w_q) \quad (1)$$

за обмежень

$$\sum_{q=1}^m U_q(k) = 1, \\ 0 < \sum_{q=1}^m U_q(k) < N.$$

Вирішуючи задачу нелінійного програмування за допомогою методу невизначених множників Лагранжа, приходимо до відомого результату

$$U_q^{(\tau+1)}(k) = (D^2(x_k, w_q^{(\tau)}))^{-\frac{1}{1-\beta}} \times \quad (2)$$

$$\times \left( \sum_{l=1}^m (D^2(x_k, w_l^{(\tau)}))^{-\frac{1}{1-\beta}} \right)^{-1},$$

$$w_q^{(\tau+1)} = \sum_{k=1}^N (U_q^{(\tau+1)}(k))^\beta x_k \left( \sum_{k=1}^N (U_q^{(\tau+1)}(k))^\beta \right)^{-1}, \quad (3)$$

де  $U_q(k)$  — рівень нечіткої належності спостереження  $x_k$  до  $q$ -го кластера;  $Cl_q$  ( $1 \leq q \leq m$ ),  $w_q$  — прототип-центроїд  $q$ -го кластера;  $\beta > 1$  — параметр фазифікації, що задає «розмитість» границь кластерів;  $D(x_k, w_q)$  — відстань між  $x_k$  та  $w_q$  у прийнятій метриці;  $\tau = 0, 1, 2, \dots$  — індекс епохи обробки інформації в режимі почергового оцінювання.

При цьому процес обчислень триває до виконання умови  $w_q^{(\tau+1)} - w_q^{(\tau)} \leq \epsilon \forall 1 \leq q \leq m$ , де  $\epsilon$  — наперед заданий поріг точності обчислень.

У випадку  $\beta=2$  та евклідової метрики

$$D^2(x_k, w_q) = \|x_k - w_q\|_2^2$$

приходимо до популярного алгоритму нечітких С-середніх (FCM) [13] вигляду

$$U_q^{(\tau+1)}(k) = \|x_k - w_q^{(\tau)}\|^{-2} \left( \sum_{l=1}^m \|x_k - w_l^{(\tau)}\|^{-2} \right)^{-1}, \quad (4)$$

$$w_q^{(\tau+1)} = \sum_{k=1}^N (U_q^{(\tau+1)}(k))^2 x_k \left( \sum_{k=1}^N (U_q^{(\tau+1)}(k))^2 \right)^{-1}. \quad (5)$$

Якщо дані надходять на обробку послідовно в онлайн режимі, задача нелінійного програмування може бути вирішена за допомогою алгоритму Ерроу-Гурвіца-Удзави, що є за суттю градієнтною процедурою пошуку сідової точки функції Лагранжа на основі критерію (1) з обмеженнями на суму належностей.

При цьому співвідношення (2), (3) можуть бути переписані у формі

$$\begin{cases} U_q(k+1) = (D^2(x_{k+1}, w_q^{(k)}))^{-\frac{1}{1-\beta}} * \\ * \left( \sum_{l=1}^m (D^2(x_{k+1}, w_l^{(k)}))^{-\frac{1}{1-\beta}} \right)^{-1}, \\ w_q(k+1) = w_q(k) + \eta(k+1) U_q^\beta(k+1) (x_{k+1} - w_q(k)), \end{cases} \quad (6)$$

де  $\eta(k)$  — параметр кроку навчання, а (4), (5) —

$$\begin{cases} U_q(k+1) = \|x_k - w_q(k)\|^{-2} \times \\ \times \left( \sum_{l=1}^m \|x_k - w_l(k)\|^{-2} \right)^{-1}, \\ w_q(k+1) = w_q(k) + \eta(k+1)U_q^2(k+1) \times \\ \times (x_{k+1} - w_q(k)), \end{cases} \quad (7)$$

що є узагальненням рекурентних процедур Парка-Деггера [9] і Чанга-Лі [10].

Можливісні алгоритми нечіткої кластеризації засновані на мінімізації цільової функції [5]

$$\begin{aligned} E(U_q(k), w_q, \mu_q) &= \sum_{k=1}^m \sum_{q=1}^N U_q^\beta(K) D^2(x_k, w_q) + \\ &+ \sum_{q=1}^m \mu_q \sum_{k=1}^N (1 - U_q(k))^\beta, \end{aligned} \quad (8)$$

де  $\mu_q \geq 0$  визначає відстань, на якій рівень належності приймає значення 0,5, тобто  $U_q(k) = 0$ , якщо  $D^2(x_k, w_q) = \mu$ .

Мінімізація критерію (8) дозволяє отримати аналітичний розв'язок у вигляді

$$U_q^{(\tau+1)}(k) = \left( 1 + \left( \frac{D^2(x_k, w_q^{(\tau)})}{\mu_q^{(\tau)}} \right)^{\frac{1}{\beta-1}} \right)^{-1}, \quad (9)$$

$$w_q^{(\tau+1)} = \sum_{k=1}^N \left( U_q^{(\tau+1)}(k) \right)^\beta x_k \left( \sum_{k=1}^N \left( U_q^{(\tau+1)}(k) \right)^\beta \right)^{-1}, \quad (10)$$

$$\begin{aligned} \mu_q^{(\tau+1)} &= \sum_{k=1}^N \left( U_q^{(\tau+1)}(k) \right)^\beta D^2(x_k, w_q^{(\tau+1)}) \times \\ &\times \left( \sum_{k=1}^N \left( U_q^{(\tau+1)}(k) \right)^\beta \right)^{-1}, \end{aligned} \quad (11)$$

яке у квадратичному випадку набуває вигляду

$$U_q^{(\tau+1)}(k) = \left( 1 + \frac{\|x_k - w_q^{(\tau)}\|^2}{\mu_q^{(\tau)}} \right)^{-1}, \quad (12)$$

$$w_q^{(\tau+1)} = \sum_{k=1}^N \left( U_q^{(\tau+1)}(k) \right)^2 x_k \left( \sum_{k=1}^N \left( U_q^{(\tau+1)}(k) \right)^2 \right)^{-1}, \quad (13)$$

$$\mu_q^{(\tau+1)} = \sum_{k=1}^N \left( U_q^{(\tau+1)}(k) \right)^2 \|x_k - w_q^{(\tau+1)}\|^2 \left( \sum_{k=1}^N \left( U_q^{(\tau+1)}(k) \right)^2 \right)^{-1}. \quad (14)$$

Онлайн версії (9)-(14) при цьому мають вигляд [11, 12]

$$\begin{cases} U_q(k+1) = \left( 1 + \left( \frac{D^2(x_{k+1}, w_q(k))}{\mu_q(k)} \right)^{\frac{1}{\beta-1}} \right)^{-1}, \\ w_q(k+1) = w_q(k) + \eta(k+1)U_q^\beta(k+1) * (x_{k+1} - w_q(k)), \\ \mu_q(k+1) = \sum_{p=1}^{k+1} U_q^\beta(p) D^2(x_p, w_q(k+1)) * \left( \sum_{p=1}^{k+1} U_q^\beta(p) \right)^{-1} \end{cases} \quad (15)$$

та (при  $\beta = 2$ )

$$\begin{cases} U_q(k+1) = \left( 1 + \frac{\|x_{k+1} - w_q(k)\|^2}{\mu_q(k)} \right)^{-1}, \\ w_q(k+1) = w_q(k) + \eta(k+1)U_q^2(k+1) * (x_{k+1} - w_q(k)), \\ \mu_q(k+1) = \sum_{p=1}^{k+1} U_q^2(p) \|x_p - w_q(k+1)\|^2 * \left( \sum_{p=1}^{k+1} U_q^2(p) \right)^{-1}. \end{cases} \quad (16)$$

Достовірна нечітка кластеризація пов'язана з мінімізацією цільової функції

$$E(Cr_q(k), w_q) = \sum_{k=1}^m \sum_{q=1}^N Cr_q^\beta(k) D^2(x_k, w_q) \quad (17)$$

за обмежень

$$0 \leq Cr_q(k) \leq 1 \forall q, k;$$

$$\sup Cr_q(k) \geq 0,5 \forall k;$$

$$Cr_q(k) + \sup Cr_r(k) = 1$$

для будь яких  $q$  та  $k$ , для яких  $Cr_q(k) \geq 0,5$ . Тут  $Cr_q(k)$  — рівень нечіткої достовірності того, що спостереження  $x_k$  належить кластеру  $U_q$ . При цьому рівень достовірності розраховується на основі функції належності [14]

$$U_q(k) = \gamma_q(D(x_k, w_q)) \quad (18)$$

що задовольняє умовам:

- $\gamma_q(\cdot)$  монотонно зменшується на інтервалі  $[0, \infty]$ ,
- $\gamma_q(0) = 1$ ,
- $\gamma_q(\infty) \rightarrow 0$ .

Нескладно помітити, що функція (18) є за суттю мірою подібності, заснованій на відстані [15].

В якості такої функції у [14] було запропоновано використовувати вираз

$$U_q(k) = \left( 1 + D^2(x_k, w_q) \right)^{-1}, \quad (19)$$

що описує за суттю звичайну дзвонувату функцію належності, яка використовується в системах нечіткого висновування.

Цікаво зауважити, що вираз (2) може бути переписано у формі

$$\begin{aligned} U_q(k) &= \left( D^2(x_k, w_q(k)) \right)^{\frac{1}{1-\beta}} \times \\ &\times \left( \sum_{l=1}^m \left( D^2(x_k, w_l(k)) \right)^{\frac{1}{1-\beta}} \right)^{-1} = \\ &= \left( D^2(x_k, w_q(k)) \right)^{\frac{1}{1-\beta}} \left( D^2(x_k, w_q(k)) \right)^{\frac{1}{1-\beta}} + \\ &+ \sum_{\substack{l=1 \\ l \neq q}}^m \left( D^2(x_k, w_l(k)) \right)^{\frac{1}{1-\beta}} )^{-1} = \\ &= \left( 1 + \left( D^2(x_k, w_q(k)) \right)^{\frac{1}{1-\beta}} \sum_{\substack{l=1 \\ l \neq q}}^m \left( D^2(x_k, w_l(k)) \right)^{\frac{1}{1-\beta}} \right)^{-1}, \end{aligned} \quad (20)$$

яка для евклідової метрики  $\beta = 2$  приймає вигляд функції щільності розподілу Коші з параметром ширини  $\sigma_q^2$  [16]:

$$U_q(k) = \left( 1 + \frac{\|x_k - w_q(k)\|^2}{\sigma_q^2} \right)^{-1}, \quad (21)$$

$$\sigma_q^2 = \left( \sum_{\substack{l=1 \\ l \neq q}}^m \|x_k - w_l(k)\|^{-2} \right)^{-1}. \quad (22)$$

Нескладно бачити, що функція належності (19) є окремим випадком (21) при  $\sigma_q^2 = 1$ .

Остаточний пакетний алгоритм достовірної нечіткої кластеризації може бути записаний у вигляді [7,8]:

$$U_q^{(\tau+1)}(k) = \left( 1 + D^2(x_k, w_q^{(\tau)}) \right)^{-1}, \quad (23)$$

$$U_q^{*(\tau+1)}(k) = U_q^{(\tau+1)}(k) \left( \sup_l U_l^{(\tau+1)}(k) \right)^{-1}, \quad (24)$$

$$Cr_q^{(\tau+1)}(k) = \frac{1}{2} \left( U_q^{*(\tau+1)}(k) + 1 - \sup_{l \neq q} U_l^*(k) \right), \quad (25)$$

$$w_q^{(\tau+1)} = \sum_{k=1}^N \left( Cr_q^{(\tau+1)}(k) \right)^\beta x_k \left( \sum_{k=1}^N \left( Cr_q^{(\tau+1)}(k) \right)^\beta \right)^{-1}. \quad (26)$$

На підставі (17), (21) — (26) можна ввести у розгляд онлайн рекурентну версію алгоритму достовірної нечіткої кластеризації у вигляді

$$\left\{ \begin{aligned} \sigma_q^2(k+1) &= \left( \sum_{\substack{l=1 \\ l \neq q}}^m \|x_{k+1} - w_l(k)\|^{-2} \right)^{-1}, \\ U_q(k+1) &= \left( 1 + \frac{\|x_{k+1} - w_q(k)\|^2}{\sigma_q^2(k+1)} \right)^{-1}, \\ U_{(k+1)}^* &= U_q(k+1) \left( \sup_l U_l(k+1) \right)^{-1}, \\ Cr_q(k+1) &= \frac{1}{2} \left( U_q^*(k+1) + 1 - \sup_{l \neq q} U_l^*(k+1) \right), \\ w_q(k+1) &= w_q(k) + \eta(k+1) Cr_q^\beta(k+1) * \\ & * (x_{k+1} - w_q(k)). \end{aligned} \right. \quad (27)$$

Як можна бачити, з обчислювальної точки зору online алгоритм достовірної нечіткої кластеризації не складніший рекурентних версій FCM і РСМ, зберігаючи при цьому переваги достовірного підходу.

## 2. Результати обчислювального експерименту

Щоб перевірити ефективність, оцінити працездатність та спроможність якісно кластеризувати велику кількість даних розробленого методу, а також довести його перевагу над аналогами, було проведено експериментальне дослідження за допомогою двох різних баз даних.

Проведено порівняльний аналіз якості кластеризації даних за основними характеристиками

рейтингів якості таких, як: коефіцієнт розподілу (PC) який визначає «перекриття» між кластерами; індекс розподілу (SC), що кількісно визначає співвідношення суми компактності та розділення кластерів; індекс Ксі — Бені (XB) визначає співвідношення загальної мінливості всередині кластерів та їх поділу до існуючих методів кластеризації та запропонованого методу.

Для порівняння було обрано найбільш відомі методи кластеризації, такі як: нечіткі С-середні (FCM), алгоритм Густафсона-Кесселя (GK), алгоритм Гата-Геви (GG), метод адаптивної ймовірнісної нечіткої кластеризації, метод адаптивної нечіткої можливісної кластеризації даних та метод адаптивної нечіткої кластеризації даних.

Результати експериментальних досліджень та порівняльного аналізу із зазначеними характеристиками наведені в табл. 1 та табл. 2, для першої та другої вибірок відповідно.

Таблиця 1

Порівняльна оцінка якості кластеризації нечітких методів кластеризації з використанням першого набору даних

Методи кластеризації даних	Перша вибірка		
	PC	SC	XB
FCM	0,50	1,62	0,19
Густафсон-Кессель	0,27	1,66	1,62
Гат-Гева	0,25	1,54	1,35
Адаптивна ймовірнісна нечітка кластеризація	0,25	1,44	<b>0,01</b>
Адаптивна нечітка можливісна кластеризація	0,26	1,22	<b>0,01</b>
Рекурентна нечітка достовірна кластеризація	<b>0,21</b>	<b>1,13</b>	<b>0,01</b>

Таблиця 2

Порівняльна оцінка якості кластеризації нечітких методів кластеризації з використанням другого набору даних

Методи кластеризації даних	Друга вибірка		
	PC	PC	PC
FCM	0,48	1,60	0,19
Густафсон-Кессель	0,26	1,64	1,62
Гат-Гева	0,26	1,50	1,35
Адаптивна ймовірнісна нечітка кластеризація	0,25	1,42	<b>0,01</b>
Рекурентна нечітка можливісна кластеризація	0,37	<b>1,11</b>	0,18
Адаптивна нечітка достовірна кластеризація	<b>0,23</b>	1,22	<b>0,01</b>

Аналізуючи та оцінюючи отримані результати експериментальних досліджень, можна зробити висновки, що запропонований метод рекурентної достовірної нечіткої кластеризації великих даних з використанням функції належності спеціального типу має достатньо якісні результати кластеризації, що підтверджується експериментально.

Увагу привертає ще те, що запропонований підхід достатньо простий в реалізації та практично по всім критеріям аналізу кластерування даних, не поступається більш відомим нечітким алгоритмам.

На рис. 1 та 2 більш наочно продемонстрована робота запропонованого методу рекурентної достовірної нечіткої кластеризації великих даних з використанням функції належності спеціального типу.

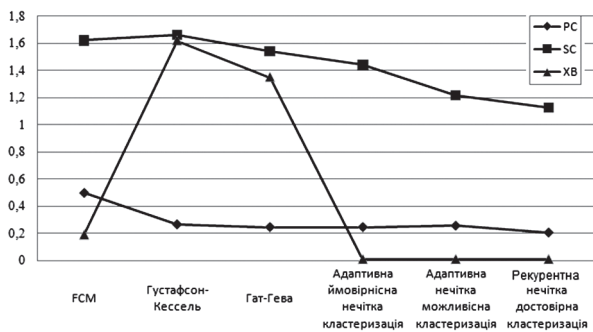


Рис. 1. Порівняльний аналіз роботи методів кластеризації для першої вибірки

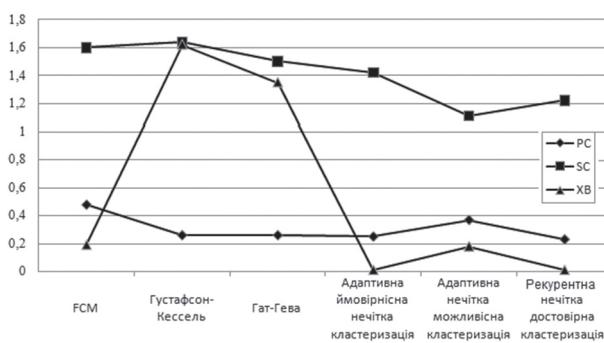


Рис. 2. Порівняльний аналіз роботи методів кластеризації для другої вибірки

### Висновки

Розглянуто задачу нечіткої кластеризації на основі ймовірнісного, можливісного і достовірного підходів на основі пакетного і online режимів надходження і обробки інформації. Введена рекурентна версія достовірного алгоритму, що є за суттю процедурою градієнтної оптимізації прийнятого критерія нечіткої достовірної кластеризації. Введена модифікація функції належності, що є по суті мірою подібності та узагальненням відомих раніше функцій. Запропонований метод рекурентної достовірної нечіткої кластеризації великих даних з використанням функції належності спеціального типу є доволі простим в чисельній реалізації і призначений для вирішення завдань, що виникають у рамках інтелектуального аналізу великих даних.

### Список літератури:

- [1] Xu R., Wunsch D.C. II. Clustering— Hoboken, N.J.: John Wiley & Sons, Inc., 2009.
- [2] Aggarwal C.C. Data Mining: Text Book. Springer, 2015.
- [3] Bezdek J.C. Pattern Recognition with Fuzzy Objective Function Algorithms.-N.Y.:Plenum Press, 1981.

- [4] Höppner F., Klawonn F., Kruse R., Runkler T. Fuzzy Clustering Analysis: Methods for Classification, Data Analysis and Image Recognition.-Chichester: John Wiley & Sons, 1999.-289p.
- [5] R. Krishnapuram, J.M. Keller. A possibilistic approach to clustering. Fuzzy Systems, 1993, 1, №2, P.98-110.
- [6] Chintalapudi K. K. and M. Kam, "A noise resistant fuzzy c-means algorithm for clustering," IEEE Conference on Fuzzy Systems Proceedings, vol. 2, May 1998, pp. 1458-1463.
- [7] Zhou J., Wang Q., Hung C.-C., Yi X. Credibilistic clustering: the model and algorithms. Int.J. of Uncertainty, Fuzziness and Knowledge-Based Systems- 2015-23-№4-P.545-564.
- [8] Zhou, J., Wang, Q., Hung, C. C. Credibilistic clustering algorithms via alternating cluster estimation.- J. Intell. Manuf.-2017-28-P.727-738.
- [9] Liu, B., & Liu, Y. Expected value of fuzzy variable and fuzzy expected value models. IEEE Transactions on Fuzzy Systems, — 2002-10-№ 4-P. 445–450.
- [10] Liu, B. A survey of credibility theory. Fuzzy Optimization and Decision Making-2006-5-№4-P. 387–408.
- [11] D.C. Park, I. Dagher. Gradient based fuzzy c-means (GBFCM) algorithm. Proc. IEEE Int. Conf. on Neural Networks, 1984, P.1626-1631.
- [12] F.L. Chung, T. Lee. Fuzzy competitive learning. Neural Networks, 1994, 7, №3, P.539-552.
- [13] Bodyanskiy Ye, Kolodyazhniy V., Stephan A. Recursive fuzzy clustering algorithms. —Proc 10th East West Fuzzy Coll. 2002, -Zittau- Görlitz, HS, 2002-P.276-283.
- [14] Bodyanskiy, Ye. Computational intelligence techniques for data analysis / Ye. Bodyanskiy // Lecture Notes in Informatics.-Bonn: V-P- 72, GI, 2005. — P.15-36.
- [15] Zhou, J., & Hung, C.-C. (2007). A generalized approach to possibilistic clustering algorithms. Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems. — 2007 — 15. — P. 117–138.
- [16] Young F.W., Hamer R.M. Theory and Applications of Multidimensional Scaling-Hillsdale, N.J.: Erlbaum, 1994.
- [17] Hu Zh., Bodyanskiy Ye, Tyshchenko O., Shafronenko A. Fuzzy clustering of incomplete data by means of similarity measures- Proc.2019 IEEE 2nd Ukr. Conf. on Electrical and Computer Engineering (UKRCON), Track 6.-Lviv, Ukraine, 2019.-P.149-152.
- [18] Bezdek J.C. A convergence theorem for the fuzzy ISODATA clustering algorithms. — IEEE Trans. Pattern Anal. Mach. Intell. — 1980 — 2. — P. 1- 8.
- [19] Bodyanskiy Ye., Gorshkov Ye., Kokshenev I., Kolodyazhniy V. Outlier resistant recursive fuzzy clustering algorithms. Ed. by B. Reusch «Computational Intelligence Theory and Applications» — Advances in Soft Computing-Vol.38.-Berlin Heidelberg, Springer Verlag, 2006-P.647-652.
- [20] Bodyanskiy Ye., Gorshkov Ye., Kokshenev I., Kolodyazhniy V. Robust recursive fuzzy clustering algorithms- Proc. 12th East West Fuzzy Coll 2005 — Zittau- Görlitz, FH,2005-P.301-308.
- [21] Bodyanskiy Ye, Shafronenko A., Mashtalir S., Online robust fuzzy clustering of data with omissions using similarity measure of special type — Lecture Notes in Computational Intelligence and Decision Making-Cham: Springer, 2020-P.637-646.

Надійшла до редколегії 18.11.2020



І.Ю. Шубін<sup>1</sup>, Г.Г. Четвериков<sup>2</sup>, В.А. Ляшик<sup>3</sup>, Н.О. Шанидзе<sup>4</sup>

<sup>1</sup>Кандидат технічних наук, професор кафедри програмної інженерії,  
Харківський національний університет радіоелектроніки,  
igor.shubin@nure.ua, ORCID iD: 0000-0002-1073-023X

<sup>2</sup>Доктор технічних наук, професор кафедри програмної інженерії,  
Харківський національний університет радіоелектроніки,  
grirorij.chetverykov@nure.ua, ORCID iD: 0000-0001-5293-5842

<sup>3</sup>Аспірант кафедри програмної інженерії,  
Харківський національний університет радіоелектроніки,  
volodymyr.liashyk@nure.ua, ORCID iD: 0000-0001-7326-0813

<sup>4</sup>Кандидат соціологічних наук, доцент кафедри соціології та політології,  
Національний технічний університет «Харківський політехнічний інститут»,  
nashanidze@ukr.net ORCID iD 0000-0002-9613-186X

## АДАПТИВНЕ ТЕСТУВАННЯ ЗНАТЬ МЕТОДАМИ ЛОГІЧНИХ МЕРЕЖ

Під адаптивним тестовим контролем розуміють комп'ютеризовану систему науково обґрунтованої перевірки й оцінювання результатів навчання, що має високу ефективність за рахунок оптимізації процедур генерації, пред'явлення й оцінки результатів виконання адаптивних тестів, що заснована на методах побудови та оптимізації логічних мереж. Алгоритми підбору й пред'явлення завдань будуються за принципом зворотного зв'язку, коли при правильній відповіді суб'єкта навчання чергове завдання вибирається більш важким, а невірна відповідь спричиняє пред'явлення наступного більш легкого завдання, ніж те, на яке суб'єктом навчання була дана невірна відповідь. Також є можливість завдання додаткових питань по темах, які суб'єкт навчання знає не дуже добре для більш точного з'ясування рівня знань у даних областях. Вибір алгоритмів тестування наразі фактично обмежений формами представлення тестових завдань і алгоритмами оцінювання результатів тестування. Досягнення більш високих результатів і підвищення мотивації навчання в остаточному підсумку є основною метою тестування знань. Для визначення базового алгоритму, необхідно навести сценарій роботи системи. У його основі лежить модель приймання іспиту викладачем у студента, як модель адаптивного тестування. Такий вибір сценарію роботи системи обумовлений тим, що, по-перше, дана процедура історично добре формалізована, по-друге, при проектуванні тестів, їх розробнику необхідно спиратися на загальноприйняті, відомі й використовувані їм методи з мінімальною модифікацією.

ДИСТАНЦІЙНА ОСВІТА, ЛОГІЧНА МЕРЕЖА, АЛГЕБРА СКІНЧЕННИХ ПРЕДИКАТІВ, ДИСТАНЦІЙНЕ ТЕСТУВАННЯ ЗНАТЬ, МОДЕЛЬ СУБ'ЄКТА НАВЧАННЯ

**Шубин И.Ю., Четвериков Г.Г., Ляшик В.А., Шанидзе Н.А. Адаптивное тестирование знаний методами логических сетей.** Под адаптивным тестовым контролем понимают компьютеризированную систему научно обоснованной проверки и оценки результатов обучения, имеет высокую эффективность за счет оптимизации процедур генерации, предъявления и оценки результатов выполнения адаптивных тестов, основанная на методах построения и оптимизации логических сетей. Алгоритмы подбора и предъявления задач строятся по принципу обратной связи, когда при правильном ответе субъекта обучения очередное задание выбирается более тяжелым, а неверный ответ вызывает предъявления следующего более легкого задания, чем то, на которое субъектом обучения была дана неверная ответ. Также имеется возможность задания дополнительных вопросов по темам, которые субъект обучения знает не очень хорошо для более точного выяснения уровня знаний в данных областях. Выбор алгоритмов тестирования пока фактически ограничен формами представления тестовых заданий и алгоритмами оценки результатов тестирования. Достижения более высоких результатов и повышения мотивации обучения в конечном итоге является основной целью тестирования знаний. Для определения базового алгоритма, необходимо привести сценарий работы системы. В его основе лежит модель принятия экзамена преподавателем у студента, как модель адаптивного тестирования. Такой выбор сценария работы системы обусловлен тем, что, во-первых, данная процедура исторически хорошо формализована, во-вторых, при проектировании тестов, их разработчику необходимо опираться на общепринятые, известные и используемые им методы с минимальной модификацией

ДИСТАНЦИОННОЕ ОБРАЗОВАНИЕ, ЛОГИЧЕСКАЯ СЕТЬ, АЛГЕБРА КОНЕЧНЫХ ПРЕДИКАТОВ, ДИСТАНЦИОННОЕ ТЕСТИРОВАНИЕ ЗНАНИЙ, МОДЕЛЬ ОБУЧАЕМОГО

**Shubin I.Yu., Chetverikov G.G., Liashyk V.A., Shanidze N.A. Adaptive Testing Of Knowledge by Methods of Logical Networks.** Adaptive test control is a computerized system of scientifically based verification and evaluation of learning outcomes, which is highly effective by optimizing the procedures for generating, presenting and evaluating the results of adaptive tests, based on methods of building and optimizing logical networks. Algorithms for selection and presentation of tasks are based on the principle of feedback, when the correct answer of the subject of training is the next difficult task, and the wrong answer causes the presentation of the next easier task than that to which the subject of training the wrong answer was given. It is also possible to ask additional questions on topics that the subject does not know very well to clarify the level of knowledge in these areas. The choice of testing algorithms is currently actually limited by the forms of

presentation of test tasks and algorithms for evaluating test results. Achieving higher results and increasing the motivation to learn is ultimately the main goal of testing knowledge. To determine the basic algorithm, it is necessary to provide a scenario of the system. It is based on the model of taking the exam by a teacher as a model of adaptive testing. This choice of the scenario of the system is due to the fact that, firstly, this procedure is historically well formalized, and secondly, when designing tests, their developer must rely on common, known and used methods with minimal modification.

DISTANCE EDUCATION, LOGICAL NETWORK, ALGEBRA OF FINITE PREDICATES, DISTANCE TESTING OF KNOWLEDGE, MODEL OF SUBJECT OF LEARNING

### Вступ

Наразі серед проблем дистанційної освіти [1] особливе місце займає проблема автоматизації контролю знань. Питання комп'ютерного контролю становлять великий інтерес для викладачів вузів і розробників програмних засобів реалізації такого контролю. Тим часом, питання комп'ютерного контролю знань недостатньо широко освітлені в теоретичному плані, і інтерес до них зазвичай реалізується в більшості випадків шляхом створення чергової програми комп'ютерного контролю із задалегідь складеним набором контрольних завдань. Засоби адаптивного тестування недостатньо висвітлені й опрацьовані недостатньо прозоро.

Контроль знань або тестування – процес, що проводиться з метою визначення рівня знань суб'єкта навчання [2]. Це найбільш стандартизований і об'єктивний метод контролю й оцінювання знань, умінь і навичок суб'єкта навчання, який позбавлений таких традиційних недоліків інших методів контролю знань, як неоднорідність вимог, суб'єктивність екзаменаторів, невизначеність системи оцінювання і т.ін. Рівні знань зазвичай дискретизуються. При такому підході, тестування може розглядатися як деякий діагностичний процес, а стани, що характеризують оцінки знань суб'єкта навчання, – як діагностичні стани. Тести є ефективним засобом перевірки якості знань, одержуваних студентами, і оперативного контролю ходу навчання [3]. Інформаційні освітні ресурси (ІОР), що містять тестові матеріали, можна розбити на дві категорії:

- такі, що орієнтовані на проходження студентами тестів у письмовій формі з подальшою перевіркою вручну викладачем (як варіант – сканування результатів тестування з метою їх подальшої автоматизованої перевірки);

- системи комп'ютерного тестування (СКС) з відповідним наповненням тестовими матеріалами.

Переваги другої категорії тестових ІОР очевидні. Вони дозволяють звільнити викладача від рутинної роботи при проведенні іспитів і проміжній оцінці знань у традиційному навчальному процесі, а при навчанні з використанням дистанційних технологій стають основним засобом контролю; надають можливість автоматизації обробки результатів, об'єктивність контролю й швидкість перевірки якості підготовки великого числа тестуємих по широкім колу питань. Це дозволяє визначити розділи, які представляють найбільшу складність у вивченні,

і, можливо, коректувати процес навчання залежно від результатів тестування. Надають можливість реалізації навчальної функції та дозволяють впровадити методи індивідуалізації процесу засвоєння знань суб'єктами навчання [4].

Функції контролю знань носять не тільки контролюючий характер, але також навчально-виховний і розвиваючий. Тестування є важливим елементом не тільки контролю знань, але й навчання. При навчальній організації процесу тестування користувачеві після проходження тесту надаються посилання на ті розділи навчального матеріалу, на запитання по яких він відповів невірно. Для досягнення зазначених результатів необхідно швидкими темпами розбудувати дистанційну форму навчання, впровадження якої в Україні передбачене Національною програмою інформатизації.

Контролююча функція [5] полягає у виявленні стану знань і вмінь учнів, рівня їх розвитку, у вивченні ступеня засвоєння приймань пізнавальної діяльності, навичок раціональної навчальної праці. За допомогою контролю визначається вихідний рівень для подальшого опанування знаннями, уміннями й навичками, вивчається глибина й обсяг їх засвоєння. Порівнюються планований рівень з дійсними результатами, встановлюється ефективність використовуваних викладачем методів, форм і засобів навчання.

Навчальна функція [6] контролю полягає в удосконалюванні знань і вмінь, їх систематизації. У процесі перевірки учні повторюють і закріплюють вивчений матеріал. Вони не тільки відтворюють раніше вивчене, але й застосовують знання й уміння в новій ситуації. Перевірка допомагає виділити головне, основне в досліджуваному матеріалі, зробити, що перевіряються знання й уміння більш ясними й точними. Контроль сприяє також узагальненню й систематизації знань.

Сутність діагностичної функції контролю [7] – в одержанні інформації про помилки, недоліки й про біли у знаннях і вмінях суб'єкта навчання, причинах, що їх породжують, та аналізу утруднень учнів в оволодінні навчальним матеріалом при чисельних помилках. Результати діагностичних перевірок допомагають вибрати найбільш інтенсивну методику навчання, а також уточнити напрями подальшого вдосконалювання змісту, методів та засобів навчання.

Прогностична функція [8] перевірки служить одержанню випереджальної інформації про навчально-виховний процес. У результаті перевірки знань

одержують підстави для прогнозу про хід певного відрізка навчального процесу: чи досить сформовані конкретні знання, уміння й навички для засвоєння наступної порції навчального матеріалу (розділу, теми). Результати прогнозу використовують для створення моделі подальшої поведінки, суб'єкта навчання, допускає сьогодні помилки даного типу або, що він має певні пробіли в системі приймань пізнавальної діяльності. Прогноз допомагає одержати вірні висновки для подальшого планування.

### 1. Постановка задачі

Під адаптивним тестовим контролем розуміють комп'ютеризовану систему науково обгрунтованої перевірки й оцінювання результатів навчання, що має високу ефективність за рахунок оптимізації процедур генерації, пред'явлення й оцінки результатів виконання адаптивних тестів [9].

Ефективність контрольних оцінних процедур підвищується при використанні багатокрокової стратегії відбору й пред'явлення завдань, заснованої на алгоритмах з повною контекстною залежністю, у яких черговий крок відбувається тільки після оцінки результатів виконання попереднього кроку (або кроків). Після виконання випробуванням чергового завдання шораз виникає потреба в ухваленні рішення про вибір рівня складності наступного завдання залежно від того, вірною або невірною була попередня відповідь [10].

Алгоритми відбору й пред'явлення завдань будуються за принципом зворотного зв'язку, коли при правильній відповіді суб'єкта навчання чергове завдання вибирається більш важким, а невірна відповідь спричиняє пред'явлення наступного більш легкого завдання, ніж те, на яке суб'єктом навчання була дана невірна відповідь. Також є можливість завдання додаткових питань по темах, які суб'єкт навчання знає не дуже добре для більш точного з'ясування рівня знань у даних областях. Таким чином, можна сказати, що адаптивна модель нагадує викладача на іспиті — якщо суб'єкт навчання відповідає на питання, що задаються, впевнено й правильно, викладач досить швидко ставить йому позитивну оцінку. Якщо той, кого навчають, починає давати неправильні відповіді, то викладач задає йому додаткові або навідні запитання того ж рівня складності або по тій же темі. І, нарешті, якщо суб'єкт навчання із самого початку відповідає погано, оцінку викладач теж ставить досить швидко, але негативну.

В літературі введено два поняття: «логіт рівня знань» і «логіт рівня труднощів завдання» [11]. Першу він визначив як натуральний логарифм відношення частки правильних відповідей випробуваного, на всі завдання тесту, до частки неправильних відповідей, а другу — як натуральний логарифм іншого відношення — частки неправильних відповідей на

завдання тесту до частки правильних відповідей на тіж самі завдання, по множині випробуваних. Єдина логарифмічна шкала дозволяє встановити необхідну відповідність між рівнем підготовки й труднощами завдання й, більше того, зробити корекцію результатів тестування при тестах різної складності [12].

Адаптивне тестування визначається як «сукупність процесів генерації, пред'явлення й оцінки результатів виконання адаптивних тестів, що забезпечує приріст ефективності вимірів у порівнянні із традиційним тестуванням завдяки оптимізації добору характеристик завдань, їх кількості, послідовності й швидкості пред'явлення стосовно до особливостей підготовки тестуємих» [13-14].

При адаптивному тестуванні в процесі проходження тесту (або набору тестів) будується модель суб'єкта навчання, яка використовується для генерації або вибору наступних завдань тестування залежно від рівня підготовки суб'єкта навчання. У комплексних системах отримана модель також може використовуватися в процесі навчання. У наш час адаптивне тестування реалізується в основному у вигляді алгоритмів комп'ютерного тестування.

Адаптивне тестування повинне задовольняти наступним вимогам:

- можливість регулювати пропорції підготовлених легких, середніх і важких завдань в залежності від числа правильних відповідей суб'єкта навчання;
- можливість регулювати пропорцій запропонованих різних тематичних розділів навчальної програми в тесті;
- можливість регулювання рівнів складності запропонованих тестів з урахуванням семантичної компетенції тестуємого;
- включення адаптивного механізму переводу на більш високий рівень завдань на тому ж самому рівні запропонованих завдань;
- кожне завдання більш високого рівня оцінюється більш високими балами.

Вибір алгоритмів тестування наразі фактично обмежений формами представлення тестових завдань і алгоритмами оцінювання результатів тестування. Досягнення більш високих результатів і підвищення мотивації навчання в остаточному підсумку є основною метою тестування знань [15]. Сам процес тестування обумовлений алгоритмом тестування повинен бути максимально формалізований і, у той же час, гнучкий для того щоб одержувати адекватні оцінки знань тестуємих. Крім того, можливість регулювати пропорції запропонованих легких, середніх і трудних завдань залежно від числа правильних відповідей тестуємого є нетривіальною вимогою. Це пов'язане з тим, що, в остаточному підсумку, при оцінці здатностей суб'єкта навчання так чи інакше використовуються статистичні методи для валідної апроксимації функції успіху в розумінні моделі Раша.

**2. Опис проведених теоретичних досліджень**

Самі набори тестових завдань, що розробляються, повинні бути адекватним відбиттям предметної галузі, а сама процедура адаптивного тестування (включаючи розроблені тести, алгоритми тестування, алгоритми оцінювання знань) повинна надійно контролюватися в процесі розробки й тестуватися як продукт.

Таким чином, вибір такого алгоритму тестування, структури тестів, які б задовольняли викладеним вимогам і характеристикам не є тривіальним завданням і надалі визначається як завдання дослідження.

Моделі лінійного програмування для конструювання систем багатоступінчастого тестування заснована на критерії максимуму інформації, що дозволяє брати до уваги різноманітні обмеження для різних тестових структур. Можливість включення таких обмежень у системи східчастого адаптивного тестування привернула увагу до цього методу. Одним з видів багатоступінчастого підходу є пірамідальне тестування, що представлено на рис. 1.

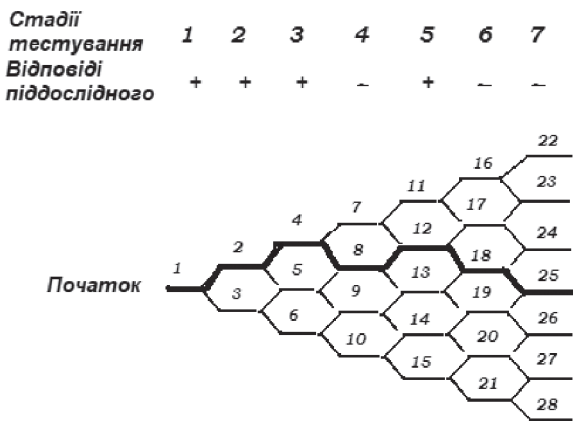


Рис. 1. Пірамідальна модель адаптивного тестування

На рис. 1 жирною лінією показаний маршрут одного суб'єкта навчання (траєкторія відповідей на запитання).

Якщо зміна порядку пред'явлення тестового завдання відбувається на кожному кроці тестування (постійна адаптація), то траєкторія відповідей на запитання прийме вид представлений на рис. 2.

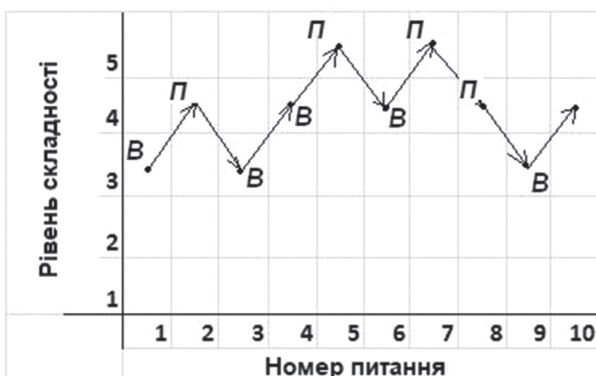


Рис. 2. Схема проходження завдань при постійній адаптації (П – помилкова відповідь, В – вірна відповідь)

Якщо ухвалення рішення про зміну порядку проходження завдань здійснюється після аналізу результатів звітів випробуваного на спеціальному блоку завдань (блокова адаптація), то схема проходження завдань має такий вигляд (див. рис. 3).

На протизагу дискретним завданням, множини завдань встановлюють так звані блоки, що являють собою зв'язані завдання, які об'єднані загальною тематикою (або за іншою ознакою), притому ці зв'язки розглядаються як неподільні одиниці.

Блоки питань об'єднані таким чином, що припускають фіксований порядок видачі тестових завдань.



Рис. 3. Схема проходження завдань при блоковій адаптації: (О – помилкова відповідь, В – вірна відповідь)

Методи лінійного програмування вводять обмеження через алгоритм вибору завдання. Останні два підходи вбудовують усі обмеження безпосередньо в групу з множині тестів з якої потім проводиться керування самим процесом тестування. Відмінності між наведеними методами призводять до наступних аспектів:

- до рівня адаптації, який можливий під час тестування;
- до розширення опису самого завдання;
- до можливості експертній оцінці дійсного змісту тестів;
- до природи реалізації обмежень, керованих процесом тестування;
- до можливості порушення цих обмежень.

Методи передбачають відновлення оцінки знань після кожного тестового завдання – у такий спосіб вони припускають можливість максимального рівня адаптації. Однак, для того щоб процедура тестування була успішною, обоє методи враховують усі стосовні до завдання закодовані атрибути. Якщо потенційно важливий атрибут пропущений, зміст тесту може стати розбалансованим. Також виконується мінімізація тільки зважених сум відхилень від обмежень, тому деякі із цих обмежень можуть бути втрачені навіть тоді, коли повністю виконане кодування атрибутів тестового завдання.

В послідовних підходах вибір завдань і реалізація обмежень є зв'язаними. Хоча послідовний вибір тестових завдань припускає оптимальну адаптацію,

послідовна реалізація обмежень не є ідеальною. Алгоритми з такими особливостями мають схильність до вибору завдань із найбільшою кількістю зв'язків з іншими темами на початку тестування. Однак вибір деяких із цих питань може виявитися не оптимальним при подальшому проходженні тесту. У цьому випадку це призведе до того, що результат оцінки знань буде менш адекватним, ніж при оптимальному адаптивному тестуванні та/або до неможливості завершення тесту без порушення обмежень.

В експертних методах є опція експертного оцінювання всього тестового матеріалу перед процедурою тестування. Явне кодування істотних для завдань атрибутів не завжди є необхідним. Також ці підходи не мають повної адаптації при виборі завдання. Ці методи дозволяють часткову адаптацію з ієрархічними переходами. Завдання угруповання блоків тестових завдань і створення багатоступінчастої тестової системи, таким чином, щоб будь-яка частина системи задовольняла всім обмеженням, призводить до великої комбінаторної проблеми, яка у свою чергу може призвести до неоптимальної системи з ієрархічними переходами та/або до порушення обмежень. Однак, як було відзначено, формальні методи для створення багатоступінчастих систем тестування існують.

Використання формальних методів для створення багатоступінчастих систем тестування дозволяє одержати явне кодування всіх важливих для завдання атрибутів.

Аналіз здійснювався за наступними критеріями:

- інформативність – застосування принципу максимуму інформації для кінцевої кількості заданих тестових завдань (теорія Г.Раша);

- статистична обробка – у процесі тестування здійснюється оцінювання здатностей суб'єкта тестування методами математичної статистики (наприклад  $j$ -та ймовірність правильної відповіді на завдання  $i$ -го рівня складності), крім цього статистична обробка дозволяє визначити момент закінчення процесу тестування, що відбувається, коли функція розподілу  $j$ -х ймовірностей правильних відповідей на завдання  $i$ -го рівня складності змінюється несуттєво.

- імовірність помилок – це ймовірність того, що тестування буде проводитися неадекватно знанням суб'єкта тестування або відбудеться порушення обмежень системи тестування;

- складність проектування;

- адекватність оцінки знань суб'єкта тестування;

- рівень адаптації, що пов'язаний із критерієм інформативності – чим вище рівень адаптації, тим вище інформативність системи;

- експертна оцінка – результативний висновок, що дозволяє робити висновки про ті або інші властивості тестових завдань;

- формалізація процедур створення тестових завдань і обмежень, що припускає застосування

рутинного алгоритму для досягнення завдання мети при певному наборі початкових умов.

Оцінка існуючих методів комп'ютерного адаптивного тестування призводить до дуже важливої дилеми. Алгоритм із оптимальними властивостями повинен був би вибирати завдання в такій послідовності, яка дозволяла досягати оптимальної адаптації й одночасно враховувати всі обмеження для того, щоб запобігти порушенням деяких з них або не одержати неоптимальну адаптацію при подальшій тестуванні.

Можливі наступні вирішення цієї дилеми.

- реалізація алгоритму з можливістю повернення назад, для того щоб поліпшити наступні рішення;

- реалізація маршового алгоритму, який би враховував наслідки прийнятих у майбутньому рішень. При адаптивному тестуванні відстеження у зворотному порядку неможливо, алгоритм застосовується в масштабі реального часу і попередній вибір не може бути відмінний. Таким чином, залишається тільки одна можливість – використовувати алгоритм, який щораз вибирає нове завдання. Це винятково новий клас алгоритмів.

### 3. Опис методу адаптивного тестування із застосуванням логічних мереж

3.1. *Моделі рішення завдання адаптивного тестування.*

Для визначення базового алгоритму, необхідно навести сценарій роботи системи. У його основі лежить парадигма приймання іспиту викладачем у студента, як модель адаптивного тестування. Такий вибір сценарію роботи системи обумовлений тим, що, по-перше, дана процедура історично склалася дуже давно й добре формалізована, по-друге, при проектуванні тестів, їх розробникам необхідно спиратися на загальноприйняті, відомі й використовувані їмі методи з мінімальною модифікацією. Сам алгоритм адаптивного тестування повинен супроводжуватися на кожному кроці (при переході від одного завдання до іншого) максимальною інформативністю (надавати максимум інформації про відповіді студента на кожне питання, що було запропоновано). У той же час, не можна повністю відмовлятися від обліку суб'єктивних властивостей суб'єкта навчання, які можуть виражатися в нерозумінні очевидно поставленого питання або завдання.

Додаткові питання є невід'ємною частиною тестів за певних умов, однак потрібно врахувати:

- не всі додаткові питання є тотожними як з погляду складності, так і з погляду повноти відповідності основному питанню;

- ланцюжки додаткових питань є логічно зв'язаними послідовностями;

- питання (як основні, так і додаткові) задаються послідовно, тобто неможливо задати два або більше питань одночасно;

- частота пред'явлення додаткових питань різна.

За базову, модель системи адаптивного тестування, обрано представлення у вигляді логічних мереж та математичні моделі алгебри скінченних предикатів, яке має наступні властивості:

- основною одиницею є питання певної складності, у якого можуть бути ланцюги додаткових питань;

- вибір додаткового питання визначається виходячи з імовірності появи кожного додаткового питання як потоку найпростіших подій з шляхом логічного висновку на вибір невірної відповіді;

- система тестів є замкненою в розумінні логічних мереж, тобто якщо питання (основне або додаткове) являє собою стан  $s_i$ , що має предикат виконання тестового завдання  $P(s_i)$ , то логічна сума предикатів тестових відповідей та перебування системи в стані  $S$  тотожно визначається як:

$$P = \vee P(s_i) = 1,$$

- зазначений стан – тестове завдання як процес оцінювання знань і вмінь суб'єкта навчання, повинне виконуватися у випадку вдалої відповіді на одне з питань та/або додаткового питання (або їх ланцюга), тобто видалення одного з питань не повинне призвести до присвоєння стану нульового значення.

Остання вимога дозволяє обходити питання нескінченно, тобто завершення тесту можливо в наступних випадках:

- вичерпані всі питання в банку тестових завдань;

- досягнуто кінець тесту;

- рівень знань оцінений з достатньою точністю;

- рівень знань суб'єкта навчання визначений як недостатній для досягнення критерію проходження тесту;

- суб'єкт навчання демонструє свою неспроможність при відповідях на запитання тесту.

Подання алгоритму тестування у вигляді предикатного опису логічних мереж не є вичерпним. Як відзначалося раніше, як при правильній, так і при невірній відповіді на запитання повинно ухвалюватися одне з наступних рішень:

- перехід до наступного основного питання з вибором рівня його складності;

- перехід до додаткових питань (до їхніх дерев), при цьому потрібно відкинути вже задані додаткові питання (як окремі, так і їх дерева);

- повернення до основного питання, якщо отримана відповідь на додатковий (або додаткові) питання;

- закінчення тестування.

Тестове завдання як суб'єкт у розглянутому процесі тестування припускає виконання правил, що закладені екзаменатором. Це відповідає й традиційному процесу приймання іспиту, тобто в залежності від відповіді на поставлені питання екзаменатор ухвалює зазначені вище рішення. При цьому він бере до уваги як фрагментарно розділені відповіді (оцінка за

кожне питання й висновок середнього), так і за певними логічними правилами весь ланцюжок відповідей суб'єкта тестування.

Загальним підходом у прийнятих методах, крім загальних правил для всіх завдань, є компонування завдань із різних частин (рисуноків, таблиць, мультимедійного контенту), обумовлених як стимули. Це дозволяє, заощаджувати ресурси по розміщенню стимулів і виділяти програму обробки відповідей, загальні правила в різних частинах логічної мережі, що розробляється.

Даний підхід не дозволяє в достатній степені індивідуалізувати тестові питання – насамперед, це пов'язане з тим, що кожне питання поєднує як безпосереднє завдання, так і рішення, що пов'язане з виконанням цього завдання з відповідями на додаткові питання.

### 3.2. Модифікований алгоритм адаптивного тестування

Враховуючи викладене, для забезпечення гнучкості в прийнятті рішень, простоти створення питань і логічних правил, що визначають прийняття рішень для конкретного питання, доцільним є об'єднання питання й процедури ухвалення рішення, пов'язаної з ним. Такий підхід спрощує як процедуру тестування, так і саму систему тестування з погляду задоволення вимогам мінімальної складності застосовуваних алгоритмів.

Прийняття загальних для всього процесу тестування рішень, вимагає загальних підходів в одному сеансі тестування. Ці підходи визначаються:

- застосуванням загального методу (або методів), що визначає той крок у процесі тестування, коли додаткова інформація про знання суб'єкта тестування буде надлишковою;

- процедурою початку роботи системи тестування (вибір першого питання) і стратегією переходу від одного питання до наступного;

- наданням докладних результатів тестування, як у природній, так і обробленої тими або іншими методами формі.

Це вимагає використання протоколу тестування й застосування:

- алгоритмів логічного операційного й статистичного аналізу результатів тестування з погляду надмірності або недостатності інформації;

- алгоритмів, що визначають рівень підготовки суб'єкта тестування;

- алгоритмів, що забезпечують стохастичні переходи по мережах тестових завдань.

Фактично, модель суб'єкта навчання в окремій сесії визначається:

- протоколом опитування;

- результатами оцінювання знань.

Таким чином, даний підхід формує парадигму тестування, що є природною для викладача, має аналог у класичним розумінні іспиту й визначається як модель

суб'єкта навчання. Слід відзначити, що безпосереднє використання представлення логічних мереж не забезпечує рішення поставленого завдання, тому що є цілий ряд обмежень логічного характеру, що визначають як перехід зі стану в стан, видалення станів, так і обмеження кількості переходів (тестування не може проводитися нескінченно й питання не повинні повторюватися).

Процедуру додаткових питань пропонується виконувати двома способами. Перший спосіб полягає в тому, що до закінчення множини додаткових питань одного основного (або єдиного питання) не здійснюється ніяких дій, крім фіксування відповідей на запитання з наступної передачі протоколу основному питанню, де ухвалюється одне з наступних рішень:

- зарахувати відповідь на основне питання як правильний з можливим коректуванням складності;
- вважати питання незадовільним і перейти до групи питань більш низького рівня;
- повторно поставити додаткове запитання із залученням, що залишилися додаткових питань;
- повторно поставити основне запитання без залучення додаткових питань.

Другий спосіб: функції прийняття рішень про правильності відповіді та/або перехід до інших основних, додаткових питань або про закінчення тестування передається додатковому питанню.

Останнє є загальною формою, тобто псевдо-інтерактивна процедура може бути зведена до маршової шляхом видалення логічного аналізу й прийняття рішень (логічний перехід у розумінні теорії логічних мереж).

Наявність логічного аналізу й ухвалення рішення в тестовому завданні моделює правило поведінки екзаменатора. Прийнятий ситуативний підхід на підставі правил для конкретного питання про поведінку (передача керування іншому питанню) у рамках окремо взятого завдання.

Додаткові питання можуть бути обрані з основних з будь-яким рівнем складності та/або створені окремо (для конкретного тесту). Як і основним питанням, їм повинен бути привласнений рівень складності. Таким чином, можна узагальнити питання у наборі тестових завдань як структуру (клас) даних і визначити його поведінку. Питання повинні мати:

- ознаку, що визначає зв'язки (такі питання, у які з нього можна перейти), обумовлені логічними висновками;
- ознаку, що визначає зв'язки (ті питання, у які з нього можна перейти звичайним для логічних мереж способом);
- ознака «останову» питання в логічних мережах, тобто випадок, коли питання вже було задано;
- ознака, що визначає чи є питання унікальним для даної мережі;
- ознака, що визначає яке або які наступні питання можуть бути викликані цим питанням, тобто яким питанням може бути передане керування.

Питання повинні забезпечувати наступні положення:

- забезпечити приймання керування від попереднього питання з одержанням поточного протоколу опитування;
- забезпечити видачу тестового завдання й варіантів відповідей до них;
- забезпечити початок відліку власного часу в момент видачі завдання;
- підтримувати логічну функцію, що визначає зниження рівня складності, у випадку виходу за межі часу, що допускається, встановленого для відповіді на дане питання та/або прийняти (за певних умов) для цього питання рішення про правильні/неправильні відповіді;
- передати протокол опитування наступному питанню і контролерові мережі питань;
- передати ознаки, що визначають стан питання контролерові мережі питань;
- перевірити роботу питання (передбачається передача керування контролерові мережі та/або наступному питанню).

На рис. 4 представлений фрагмент мережі питань (основне питання з додатковими).

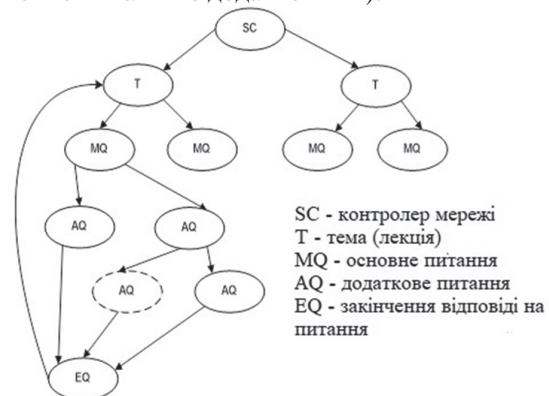


Рис. 4. Фрагмент мережі питань

### 3.3. Реалізація модифікованого алгоритму вибору тестових завдань

Використання логічних мереж на основі алгебри скінчених предикатів та предикатних операцій, припускає наявність моделі системи як абстракції, яка має сукупність станів  $S=\{s_1, \dots, s_n\}$ . Ці стани є взаємовиключними й перехід з одного стану в інший здійснюється різними для мережі шляхами, через появу події, що є в загальному випадку предикатом подій. Кожний стан системи  $s_i$  повинен мати як мінімум один вхід і один вихід, тобто так звані «глухі» або кінцеві стани відсутні. Фактично будь-який стан системи може бути обраний як початковий.

Однак на практиці вибір початкового стану визначається апріорною інформацією моделі суб'єкта навчання. Після ініціалізації деякого початкового стану робота моделі, побудованої в зазначених рамках алгебраїчної системи логічної мережі полягає в нескінченному обході мережі станів системи, у яких система перебуває впродовж деякого певного часу.

Останній метод виконує перевірку предикату чи сформувався множина такої ознаки, як складність питань для всієї групи суб'єктів навчання і одночасно чи сформувався множина предикатів розподілу відповідей залежно від їхньої складності для конкретного суб'єкта тестування. Також збільшення складності алгоритму тестування не призведе до істотного збільшення витрат ресурсів.

### Висновки

Ідея адаптивного тестування, заснована на блоках питань, має безпосереднє відношення до одного із самих розповсюджених форматів багатоступінчастого тестування, при якому суб'єкт навчання проходить через послідовність тестів, рухаючись у бік більш складних питань при успішних відповідях або до більш простих, якщо його відповіді невірні. При цьому перехід від питання до питання відбувається за певними правилами. У більш ранній літературі обговорювалася паперова версія цього методу без статистичної обробки знань суб'єкта тестування після кожного тестового завдання, однак автоматизація у практиці тестування надає можливість статистичної оцінки знань на кожному кроці тестування.

Визначено ряд обмежень логічного характеру, що визначають як перехід зі стану в стан, видалення станів, так і обмеження кількості переходів (тестування не може проводитися нескінченно й питання не повинні повторюватися). Однак, при невеликій різниці між зазначеними передбачуваними й дійсними властивостями групи випробуваних застосування зазначеної методики дає позитивний ефект, із чого можна зробити висновок, що застосування запропонованого методу є ефективним для будь-яких обсягів і інших властивостей, що визначають тестування як процес.

Збільшення складності алгоритму тестування не призведе до яких або істотним витрат ресурсів ПК.

Очікуваний результат по використанню розроблених моделей стосовно до 3-х параметричної моделі тестування має бути досягнутий при апроксимації функції успіху, що припускає визначення уточнених значень дискримінаторів завдань шляхом рішення завдання оптимізації при використанні відомих алгоритмів. Це пов'язане насамперед з тим, що застосовані алгоритми оптимізації є універсальними, а рішення виниклої проблеми вимагає додаткового дослідження метою якого є часткове заміщення цих алгоритмів операційними методами, що використовують як властивість поставленого завдання тестування так і іншої апріорної інформації.

Різниця в рівнях складності основних і додаткових питань і запропонований зв'язок між основними питаннями і гілками додаткових питань дозволяє в процесі тестування як мінімізувати кількість необхідних відповідей суб'єкта навчання для визначення рівня його знань, так і суттєво поліпшити адаптаційні властивості тестування.

На підставі проведення тестів можна стверджувати, що створений прототип виконує всі функції, зазначені у вихідних даних на розробку й відповідає всім вимогам, що пред'явлені до нього.

Застосування методів побудови логічних мереж для реалізації етапу перевірки знань в системах адаптивного навчання є ефективним для будь-яких обсягів і інших властивостей, що визначають тестування як процес.

### Список літератури:

- [1] Шабанов-Кушнарченко Ю.П. Про теорію інтелекту // Проблеми біоніки // Х.: Вища школа, 1979. – Вип. 22. – с. 3–11.
- [2] Шабанов-Кушнарченко Ю.П. Теорія інтелекту. Проблеми й перспективи // Х.: Вища школа, 1987. – 160 с.
- [3] Логічні мережі та їх використання для вирішення морфологічних завдань / І.Ю. Шубін, М.О. Пітюкова // Матеріали III Міжнародної конференції «Інноваційні технології в науці та освіті». м. Амстердам, Нідерланди, 2019. – С. 402-405
- [4] Алгебра предикатов и предикатных операций./ М.Ф. Бондаренко, З.В. Дударь, Н.Т. Процай, В.В. Черкашин, В.А. Чикина, Ю.П. Шабанов-Кушнарченко // Радиоелектроника и информатика. – 2004.–№ 1. – С. 5-22
- [5] Learning management system // URL: [https://en.wikipedia.org/wiki/Learning\\_management\\_system](https://en.wikipedia.org/wiki/Learning_management_system) (дата звернення: 26.04.2020).
- [6] Бейкер, Р. Educational data mining and learning analytics/ Р. Бейкер, Г. Сіменс – The Cambridge handbook of the learning sciences, 2019. – 274 с.
- [7] Fourier J. Un modele d'indexation relationnel pour les graphes conceptuels fondee sur une interpretation logique: Phd thesis // Universitee. – Grenoble, 1998.- 302 p.
- [8] Климушев В.Б. Моделирование корреляционного алгоритму розпізнавання образів на базі формальної граматики // Автоматизовані системи керування й прилади автоматки. – 2001. – №115.– С. 54-74.
- [9] Brainlike computing / Shabanov-Kushnarenko Yu., Klimushev V., Lukashenko O. // Proceedings of the East-West Design and Test Workshop. Odessa, Ukraine, September 15 – 19, 2005, p. 274 - 279.
- [10] Бондаренко М.Ф., Осика А.Ф. Автоматична обробка інформації природною мовою. // Київ: УМК В, 1991. – 143 с.
- [11] Бондаренко М.Ф., Дудар З.В., Процай Н.Т., Черкашин В.В., Чикина В.А., Шабанов-Кушнарченко Ю.П. Алгебра предикатів і предикатних операцій // Радиоелектроника й информатика. – 2004.–№ 1. – С. 5-22.
- [12] Четвериков Г.Г. Многозначные структуры (анализ, сравнение, синтез, обобщение) . ч.1. - К.: ИСМО, 1997. - 192.
- [13] Chetverikov G.G., Vechirska I.D., Tanyanskiy S.S. The methods of algebra finite predicates in the intellectual system of complex calculations of telecommunication companies // International Conference Proceedings Crimean Microwave and Telecommunication Technology (CriMiCo).-2014, 6959425. - pp. 346-347.
- [14] Chetverikov G., Puzik O., Vechirska I. Multiple-valued structures of intellectual systems //Proceedings of the with International Computer Sciences and Information Technologies (CSIT). 2016, 7589907. -pp. 204-207.
- [15] A Guide to the SCRUM BODY OF KNOWLEDGE (SBOK™ GUIDE) URL: <https://www.scrumstudy.com/SBOK/SCRUMstudy-SBOK-Guide-3rd.pdf>.

Надійшла до редколегії 15.12.2020

---

## RESUME

---

Ivan Bozhko, Grygoriy Chetverykov, Aleksandr Karataiev

### THE MODERN QUANTUM COMPUTING TOOLS INVESTIGATION

**Background.** Since the quantum computing studies started in 1982, it has been proved that this type of computations provides new possibilities comparatively to the classic computations. It has been shown that quantum computing is useful not only for scientific research, but could impact the modern cryptography by making the existing asymmetric algorithms unreliable. Since the stable and publicly available version of quantum computer has not been created yet and considering the research proving the possible impact of this, there is a high demand on the quantum computing research nowadays and it is done by many technological companies, scientists and software developers. Considering above, there are many tools for describing, implementing and testing the quantum programs (mostly using emulators) appearing nowadays, which allow to learn quantum programming and arrange quantum computing research using them.

**Materials and methods.** In order to retrieve the information about the current state of the quantum computing research, the existing tools and their peculiarities, the literature analysis was used as the main research method. After determining the list of different quantum computing tools to be investigated and compared, modeling was used. In order to perform the modeling using computer, the respective libraries and frameworks were used (Microsoft Quantum Development Kit, QCL and Quipper languages, etc.).

**Results.** In scope of the research, a list of different quantum computing tools was investigated. It included different languages (imperative (QCL), functional (Quipper), multi-paradigm (Q#)), quantum pseudocode and frameworks used with the existing languages (e.g. ProjectQ for Python). According to the retrieved data, the accurate performance cannot be determined using the given tools because of the absence of a fully functional quantum computer prototype (the results obtained for emulators cannot be considered precise). The tools cover most of the basic quantum computing units including qubits and operations with them, but differ in terms of the used syntax and the standard library.

**Conclusion.** According to the initial objective, a high-level overview of the currently existing tools was provided in this work. Since it is not possible to measure the performance of the used tools, the main criteria for the selection are existence of the necessary functions in the standard library and personal preference (e.g. primary programming language, used platform, etc.). Considering the rapidly changing research state and new studies appearing, the research requires continuation and needs to be conducted again later once the situation changes.

---

S. Chalyi, V. Leshchynskyi

### TEMPORAL PATTERNS OF USER PREFERENCES IN THE TASKS OF FORMING EXPLANATIONS IN THE RECOMMENDER SYSTEM

**Background:** Considered the problem of taking into account changes in user behavior of the recommendation system when constructing explanations for recommendations. This problem occurs as a result of cyclical changes in user requirements. Its solution is associated with the construction of an explanation comparing the alternative choices of the user of the recommendation system.

The existing approaches to the construction of temporally oriented explanations do not consider the alternative of consumer choice. To solve the problem of constructing explanations in this case, it is necessary to single out subsets of temporal dependencies that characterize these alternatives. Such dependencies are part of temporal patterns reflecting the cyclical nature of user choices over time.

**Materials and methods:** methods for constructing temporal dependencies when constructing explanations in recommender systems.

**Results:** Models of temporal patterns of alternative choice and sequential alternative choice of the consumer are proposed.

An approach to identifying temporal patterns using the technology of intelligent analysis of processes is proposed.

An experimental test of the developed approach was performed using a set of data on sales in a supermarket chain.

**Conclusion:** The developed models of temporal patterns consist of a set of temporal relationships between events of user choice of goods and services. Temporal relationships are used for pairs of events. The first pattern contains an alternative in the form of sequential selection in time of several objects or the selection of only a pair — the first and the last object. The second pattern, sequential-alternative choice, consists of a sequence of choices over time, which ends with the first pattern.

The proposed approach to the formation of patterns is based on the construction of data sets containing temporal dependencies between a group of user choices for a given level of time detail. The temporal dataset is used to construct a temporal graph of the recommender system user selection process. The latter includes a set of temporal patterns with an indication of the time of their beginning and end, which makes it possible to determine the duration of the implementation of these patterns. Taking into account the duration of the patterns, subsets of temporal relationships are formed to construct explanations for the recommended list of goods and services.

Experimental verification of the developed approach using the “Online Retail” sales data set has shown the possibility of identifying temporal patterns even on short initial samples.

---

Kyrychenko I., Nazarenko A., Popov R.

### OPTIMIZATION AND SCALING NODE.JS APPS

**Background:** The relevance of this work is due to the fact that Node.js is fast becoming one of the most popular platforms for creating fast, scalable web and mobile applications. The problem of scaling and optimization of Node.js applications are considered in the work. Several approaches to developing Node.js applications have been proposed to help optimize code execution speed and develop solutions that are easy to scale.

**Materials and methods:** Currently, there is a demanding attitude to technical solutions, especially to the issue of speed and performance of the application, including the need to take into account financial costs. The Node.js platform is revolutionary because it allows you to create a fast, easily scalable application for a small fee, and allows you to serve a large number of users that can withstand extreme loads.

**Results:** The paper analyzes and demonstrates existing ways to speed up and increase the performance of Node.js applications. Also, the speed of different versions of Node.js, solutions developed on different versions of Node.js, and methods built into the platform were clearly compared. We considered ways to speed up Node.js servers.

**Conclusion:** The best ways to speed up applications and increase their productivity were chosen. Using these methods, you can increase the number of requests processed, on average, several times.

---

V.V. Bilotserkovskyy, S.G. Udovenko, L.E. Chala

### METHOD OF NEURAL NETWORK RECOGNITION OF FALSIFIED IMAGES

**Background:** The task of detecting fake content in analyzed static and video images covers various components (media, legal and technological) and is non-trivial. The methods for generating images falsified using Deepfake technologies and methods for their detection use very similar principles. In most of these methods, various versions of the so-called convolutional generative-adversarial artificial neural networks are used, consisting of two main elements — a generator and a discriminator. Both of these elements learn together based on opposing goals. The purpose of the generator is to create a fake image that is difficult to distinguish from the real one, and the purpose of the discriminator is to identify a synthetically generated image. Thus, the urgent task is to separate falsified static or dynamic images (in particular, images of human faces), created using Deepfake technologies, from the original ones. A promising direction for solving this problem is the use of a set of EfficientNet models in combination with the Attention mechanism and the Siamese learning procedure.

**Materials and methods:** The experiments were carried out on FF ++ and DFDC datasets created to classify videos generated using DeepFake technology and containing both real and fake videos. To create an ensemble of convolutional models, the XceptionNet, EfficientNetB4, EfficientNetB4Att models and the Attention mechanism were used.

**Results:** The analysis shows that the best results of recognition of falsified images are achieved using combinations of two or more convolutional neural networks. Networking helps to increase both detection accuracy (as measured by the AUC measure) and detection quality (as assessed by the LogLoss measure).

**Conclusion:** The proposed method helps to identify fake video images generated using DeepFake technologies. The method uses a set of EfficientNet models trained using the Attention mechanism and the Siamese training strategy of the convolutional neural network.

---

O.B. Dudinova, S.G. Udovenko, L.E. Chala.

### INTELLECTUAL PROCESSING OF SPATIAL DATA IN THE GIS OF LANDSCAPE-ECOLOGICAL MONITORING

**Background:** The ecological monitoring of the environment and natural resources is carried out with the help of specialized systems of observations and further processing of the obtained data. Computer processing of realistic cartographic images of landscape objects in geographic information systems (GIS) of environmental monitoring usually involves the implementation of stages of pre-processing, segmentation, recognition and interpretation. Of particular practical interest are the tasks of object recognition by satellite images and aerial images. Today there are no universal methods and technologies of intellectual data processing in geographic information systems of ecological monitoring of regional and national levels. Thus, the development and study of a set of methods, models and algorithms for digital image processing in GIS environmental monitoring, based on the application of modern approaches combined into information technology for intelligent digital image processing in GIS environmental monitoring of landscape objects (otherwise, GISLEM — landscape ecological monitoring) and industrial zones, is an urgent scientific and practical task.

**Materials and methods:** Spatial data and images from the GIS database of environmental control, presented by the raster model, were used to test the module of image segmentation and contour selection. In the testing process, simulation of the procedure of pre-neural network processing of noisy images and the procedure of neural network selection of contours on the image by the previous procedure was carried out.

**Results:** The results obtained during the experiments indicate a reduction in computation time for some of the proposed data processing procedures and improve the quality of spatial image processing.

**Conclusion:** An approach to the creation of modular subsystems for intelligent processing and compression of spatial data as part of GISLEM is proposed. This approach improves the efficiency of solving problems of ecological monitoring of landscape objects and predicting changes in their state.

---

**A.I. Povzun, N.B. Kozel, O.A. Karataiev, G.G. Chetverikov**

**CONCEPTIONS OF ORGANIZATION OF INFORMATIVELY-INTELLECTUAL TECHNOLOGIES AND INTELLECTUAL SUPPORT OF SOCIAL-ECONOMIC PROCESSES: MULTIPLE-VALUED TOOLS. PART 2**

**Background:** The review condition of modelling functions intelligence person problem at level of its language behaviour is spent. The comparative syntactic and semantic analysis of morphology word forms is given. The hardware way realization natural language models in the form of k-unit structures (AKP-STRUCTURES) is offered. The concept matter in the field of the further development artificial intelligence systems is stated.

**Materials and methods:** The paper proposes a method for analyzing large data in the presence of different data sources and different methods of processing these data. The concept of associative dependence is introduced, the method of finding dependences is developed, efficiency and possibility of its parallelization are determined.

**Results:** Changing the concentration of contaminants at the input obtained the values of the dependence of the current on the input concentration of ferromagnetic impurities and visually demonstrated the results. After the simulation, the results of the dynamics of changes in the concentration of pollution at the output using direct current and the dynamics of changes in power consumption using direct current, where the input concentration of pollution used a random signal.

**Conclusions:** The process of purification of process waters from iron-containing impurities is considered. The structure of the scheme of the control object is constructed and the control and forgetting parameters are defined. In perspective there is the development of an automated control system for the process of purification of process water from ferromagnetic impurities.

---

**Zolotariov D.**

**THE METHODS FOR ITERATIVE COMPUTATIONS EFFICIENCY IMPROVING IN WOLFRAM MATHEMATICA**

**Background:** The article is devoted to research, development and substantiation of methods to increase the iterative algorithms computational efficiency in the mathematical processor Wolfram Mathematica. The subject of the study is the practical principles for reducing the use of Mathematica-programs such limited computer resources as RAM and CPU time. The problem places of Mathematica-programs built on iterative algorithms are revealed in the work, their influence on the speed of calculations is analyzed, solutions for each of them are developed.

**Materials and methods:** system and functional analysis, bottleneck theory.

**Results:** The study found that a significant impact on reducing the use of computer resources by iterative programs has a preliminary conversion of all heavy or frequently requested functions to a tabular or polynomial form, making related calculations outside the main iterative cycle of the program and compliance with the recommendations provided in the article for the cycle operator choice. At calculations time it needs clear unnecessary data and export to dump files those that need to be reloaded, avoid recalculating variables and constants for a certain set of function arguments by saving the results in RAM or external memory, use compiling functions into WVM and C code by math-processor. Examples of each approach use and the necessary explanations are given.

**Conclusion:** Methods for increasing the computational efficiency of iterative algorithms in the Wolfram Mathematica mathematical processor have been developed and described. It has been proven that each of them can significantly affect the use of such limited computer resources as the amount of RAM and CPU time in calculations. The results are substantiated in absolute terms and graphically.

---

**A.P. Safonyk, P.O. Lakus, I.M. Hrytsiuk**

**DEVELOPMENT AND RESEARCH OF THE ENERGY SAVING SYSTEM OF PURIFICATION TECHNOLOGICAL WATERS FROM FERROMAGNETIC IMPURITIES AND ITS AUTOMATION**

**Background:** Currently, the main indicator of the technical water quality is the degree of purification from various types of impurities, which accelerate the wear of process equipment, the formation of corrosion and reduce the quality of the product and the subsequent formation of ferromagnetic fractions. To remove this type of impurities it is proposed to use the method of magnetic purification of the aqueous environment. The main advantage of this method is no use of chemical reagents which is environmentally friendly, and therefore purification of this type is always relevant.

**Materials and methods:** To remove ferromagnetic impurities from technical water proposed method of magnetic purification of the aqueous environment. The main advantage of this method is no use of chemical reagents which is an environmentally friendly process. To automate the process of magnetic purification technical water the schematic diagram of automation has been developed, according to which the control of the purification system is the transfer of the flow of filter fluid between the two filters.

**Results:** During the work, a computer experiment was conducted to determine the transfer functions of the object on the channel "electric current- the concentration of impurities at the output", object on the channel "total concentration of contaminants in wastewater — the concentration of impurities at the output", object on the channel "total concentration of contaminants

in wastewater — the concentration of impurities at the output” described by an aperiodic link of the first order and the object on the channel “flow — pressure drop”, the object on the channel “electric current — the amount of sediment of the pollutant”, the object on the channel “total concentration of contaminants in wastewater — the volume of pollutant sediment” and the object on the channel “fluid velocity — the volume of the contaminant sediment” — real integrating link. As a result of the research, it was determined that the object on the above channels has variable parameters that depend on the operating mode of the device. The modeling is carried out on the basis of the transfer function in which within two hours the system must effectively purify process water and the concentration of pollution should not exceed  $0.5 \text{ mg/dm}^3$ . Changing the concentration of contaminants at the input obtained the values of the dependence of the current on the input concentration of ferromagnetic impurities and visually demonstrated the results. To check the operation of the regulator has been developed the computer model, which consists of two subsystems "Regulator", "Magnetic Filter". After the simulation, the results of the dynamics of changes in the concentration of pollution at the output using direct current and the dynamics of changes in power consumption using direct current, where the input concentration of pollution used a random signal.

**Conclusions:** The process of purification of process waters from iron-containing impurities is considered. The structure of the scheme of the control object is constructed and the control and forgetting parameters are defined. The computer simulation was performed on the channels “fluid velocity — the concentration of impurities at the output”, “current — the concentration of impurities at the output” and “total concentration of contaminants in wastewater — the concentration of impurities at the output”. A mathematical model in the form of transfer functions for each control channel is found. The P-regulator for energy-efficient control of pollution concentration at the output of the system is developed. It is shown that the minimum energy savings using the proposed regulator is at least 15%. In perspective there is the development of an automated control system for the process of purification of process water from ferromagnetic impurities.

---

**Ye. Bodyanskiy, A.Yu. Shafronenko, I. M. Klymova**

#### **RECURRENT AUTHENTICITY OF THE CLUSTERING OF GREAT TRIBUTE TO THE FUNCTION OF SPECIAL TYPE**

**Background:** The task of clustering multidimensional data is an important part of data mining. Here the most widespread methods of probabilistic fuzzy clustering and, first of all, the method of fuzzy C-means (FCM).

**Materials and methods:** Introduced a recurrent version of the reliable method, which is essentially a procedure for gradient optimization of the adopted criterion of fuzzy reliable clustering. Also introduced is a modification of the membership function, which is essentially a degree of similarity and generalization of previously known functions.

**Results:** Experimental results have shown that the reliable approach provides a higher quality clustering compared to probabilistic and feasible methods.

**Conclusion:** The article discusses the problem of fuzzy clustering based on probabilistic, feasible and reliable approaches based on batch and online modes of information receipt and processing. The considered recurrent procedures are simple in numerical implementation and are designed to solve problems arising in the framework of mining big data.

---

**Shubin I.Yu., Chetverikov G.G., Liashyk V.A., Shanidze N.A.**

#### **ADAPTIVE TESTING OF KNOWLEDGE BY METHODS OF LOGICAL NETWORKS**

**Background Information:** Distance learning has become a leader among other industries in Ukraine. The main factor in the further development of this field of knowledge control or testing is the process carried out in order to determine the level of knowledge of the subject. This is the most standardized and objective method of control and evaluation of knowledge, skills and abilities of the subject, which is devoid of such traditional shortcomings of other methods of knowledge control as heterogeneity of requirements, subjectivity of examiners, uncertainty of the assessment system, etc. In this approach, testing is seen as a diagnostic process, and the conditions that characterize the assessment of the subject's knowledge — as diagnostic conditions.

**Materials and methods:** Adaptive testing in the process of passing a test (or set of tests) builds a model of the subject, which is used to generate or select the next test tasks depending on the level of preparation of the subject. In complex systems, the resulting model can also be used in the learning process. Nowadays, adaptive testing is implemented mainly in the form of computer testing algorithms.

**Results:** To ensure flexibility in decision-making, ease of creating questions and logical rules that determine decision-making for a particular issue, it is advisable to combine the issue and the decision-making procedure associated with it by methods of building logical networks. This approach simplifies both the testing procedure and the testing system itself in terms of meeting the requirements of the minimum complexity of the applied algorithms.

**Conclusion:** The expected result of using the developed models in relation to the 3 parametric testing model should be achieved by approximating the success function, which involves determining the refined values of task discriminators by solving the optimization problem using known algorithms. This is primarily due to the fact that the applied optimization algorithms are universal. The use of methods for building logical networks to implement the stage of testing knowledge in adaptive learning systems is effective for any volume and other properties that define testing as a process.

# ПРАВИЛА оформлення рукописів для авторів науково-технічного журналу «БІОНІКА ІНТЕЛЕКТУ»

Науково-технічний журнал «Біоніка інтелекту» приймає до друку написані спеціально для нього оригінальні рукописи, які раніше ніде не друкувались. Структура рукопису повинна бути такою: індекс УДК, відомості про авторів, заголовок, анотації (на трьох мовах), ключові слова, вступ, основний текст статті, висновки, список використаної літератури, резюме.

Відповідно до Постанови ВАК України від 15.01.2003 №7-05/1 (Бюлетень ВАК, №1, 2003, с. 2), стаття повинна мати такі необхідні елементи: постановка проблеми у загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями; аналіз останніх досліджень і публікацій і виділення не вирішених раніше частин загальної проблеми в даній області; формулювання цілей та завдань дослідження; виклад основного матеріалу досліджень з повним обґрунтуванням отриманих наукових результатів; висновки з даного дослідження та перспективи подальших досліджень у даному напрямку.

Статті мають бути виконані в редакторі Microsoft Word. Формат сторінки – А4 (210×297 мм), поля: верхнє – 25 мм, нижнє – 20 мм, ліве, праве – 17 мм. Кількість колонок – 2, з інтервалом між ними 5 мм, основний шрифт Times New Roman, кегль основного тексту – 10 пунктів, міжрядковий інтервал – множник (1,1), абзацний відступ – 6 мм. Обсяг рукопису – від 6 до 12 сторінок (мови: українська, англійська, російська та мовою оригінала).

УДК друкується з першого рядка, без відступів, вирівнювання по лівому краю.

*ПІБ автора* (-ів), назва статті, назва та адреса учбового закладу необхідно надати повністю російською, українською та англійською мовами.

*Назва статті* друкується прописними літерами; шрифт прямий, напівжирний, кегль 12.

*Назви розділів* нумерують арабськими цифрами, виділяють жирним шрифтом. Відступи для назви статті, ініціалів та прізвищ авторів, відомостей про авторів, назв розділів, вступу та висновків, списку літератури: зверху – 6 пт, знизу – 3 пт.

*Анотації* (мовою статті, абзац 6–12 рядків, кегль 9) розміщують на початку статті, в ній має бути розміщена інформація про очікувані результати описаних досліджень (на трьох мовах).

*Ключові слова* (4–10 слів з тексту статті, які з точки зору інформаційного пошуку несуть змістовне навантаження) наводять мовою рукопису, через кому в називному відмінку, кегль 9.

*Рисунки та таблиці* (чорно-білі, контрастні) розміщуються у тексті після першого посилання у вигляді окремих об'єктів і нумерують арабськими цифрами наскрізною нумерацією за наявності більше ніж одного об'єкта. Невеликі схеми, що складаються з 3–4 елементів виконують, використовуючи вставку об'єкта Рисунок Microsoft Word. Більш складні виконують у графічних редакторах у вигляді чорно-білих графічних файлів форматів .tif, .jpg, .wmf, .cdr із

розділенням 300 dpi. Рисунки мають міститися у текстовому файлі й обов'язково подаватися окремими файлами з відповідними назвами (наприклад, рис1.jpg).

Усі елементи рисунка, включаючи написи, повинні бути згруповані. Усі написи в рисунках і таблицях мають бути виконані шрифтом Times New Roman, кегль у рисунках – 10, у таблицях – 9.

Рисунок повинен мати центрований підпис (поза рисунком), шрифт 9, відступи зверху і знизу по 6 пт. Ширина рисунка має відповідати ширині колонки (або ширині сторінки).

*Формули, символи, змінні* повинні бути набрані в редакторі формул **MathType**. Формули розміщують посередині рядка й нумерують за наявності посилань на них у рукописі. Шрифт – Times New Roman. Висота змінної – 10 пунктів, великих і малих індексів – 8 пт, основний математичний символ – 12 (10) пт. Змінні, позначені латинськими літерами, набирають курсивом, грецькі літери, скорочення російських слів і цифри – прямим написанням. Змінні, які є в тексті, також набирають у редакторі формул.

*Список літератури* вміщує опубліковані джерела, на які є посилання в тексті, укладені у квадратні дужки, друкують без абзацного відступу, кегль 9 пт, відступ зверху – 6 пт.

Після списку літератури з відступом зверху 6 пт зазначають *дату подання статті до редколегії*. Число та місяць задають двозначними числами через крапку. Розмір шрифту – 9 пт, курсив, вирівнювання по правому краю.

*Резюме* (Times New Roman, кегль – 10 пунктів,) подають англійською мовою: обсяг резюме до 2000 знаків (бажаний переклад). *Структура резюме*: **Background, Materials and methods, Results, Conclusion**.

Разом із рукописом (на аркушах білого паперу формату А4 щільністю 80-90 г/м<sup>2</sup>, надрукований на лазерному принтері) необхідно подати такі документи:

1. Заяву, яку повинні підписати всі автори.
2. Акт експертизи про можливість опублікування матеріалів у відкритому друці (якщо потрібно).
3. Рецензію, підписану доктором чи кандидатом наук.
4. Відомості про авторів.
5. Електронний варіант рукопису, резюме та відомостей про авторів.
6. Зробити оплату публікації.

Необхідно також зазначити один з наступних тематичних розділів, якому відповідає рукопис:

1. Теоретичні основи інформатики та кібернетики. Теорія інтелекту.
2. Математичне моделювання. Системний аналіз. Прийняття рішень.
3. Інтелектуальна обробка інформації. Розпізнавання образів.
4. Інформаційні технології та програмно-технічні комплекси.
5. Структурна, прикладна та математична лінгвістика.
6. Дискусійні повідомлення.

---

## СОДЕРЖАНИЕ

### **СТРУКТУРНАЯ, ПРИКЛАДНАЯ И МАТЕМАТИЧЕСКАЯ ЛИНГВИСТИКА**

<i>Широков В. А., Надутенко М. В., Стрижак О. Є., Юценко С. С.</i> Технологічні засади логіко-лінгвістичних досліджень законодавства .....	3
<i>Bozhko Ivan, Chetverykov Grygoriy, Karataiev Oleksandr.</i> The modern quantum computing tools investigation .....	15

### **ОБЪЕКТНОЕ МОДЕЛИРОВАНИЕ. ЭКСПЕРТНЫЕ СИСТЕМЫ**

<i>Чалий С. Ф., Лецинський В. О.</i> Темпоральні патерни вподобань користувачів в задачах формування пояснень в рекомендаційній системі .....	21
<i>Кириченко І. В., Назаренко А. В., Попов Р. О.</i> Оптимізація та масштабування node.js додатків .....	28

### **ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ. ЛОГИЧЕСКИЕ СЕТИ**

<i>Білоцерковський В. В., Удовенко С. Г., Чала Л. Е.</i> Метод нейромережевого розпізнавання фальсифікованих зображень .....	32
<i>Дудінова О. Б., Удовенко С. Г., Чала Л. Е.</i> Інтелектуальна обробка просторових даних в ГІС ландшафтно-екологічного моніторингу .....	43

### **ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ. МЕТОДОЛОГИЯ ИНФОРМАТИЗАЦИИ. ОНТОЛОГИЧЕСКИЕ СИСТЕМЫ**

<i>Повзун О. І., Козел Н. Б., Каратаєв О. А., Четвериков Г. Г.</i> Концепції організації інформаційно-інтелектуальних технологій та інтелектуальної підтримки суспільно-економічних процесів: k-значні засоби. Частина 2 .....	51
<i>Золотарьов Д.О.</i> Методи підвищення ефективності ітераційних обчислень у Wolfram Mathematica .....	63
<i>Сафоник А.П., Лакус О.П., Грицюк І.М.</i> Розробка та дослідження енергозберігаючої системи очищення технологічних вод від феромагнітних домішок та її автоматизація .....	69
<i>Бодянський Є.В., Шафроненко А.Ю., Климова І.М.</i> Рекурентна достовірна нечітка кластеризація великих даних з використанням функції належності спеціального типу .....	77
<i>Шубін І.Ю., Четвериков Г.Г., Ляшик В.А., Шанідзе Н.О.</i> Адаптивне тестування знань методами логічних мереж.....	82
Resume .....	90
ПРАВИЛА оформлення рукописів для авторів науково-технічного журналу «БІОНІКА ІНТЕЛЕКТУ» .....	94

*Наукове видання*

**БІОНІКА ІНТЕЛЕКТУ**  
**інформація, мова, інтелект**

**Науково-технічний журнал**

**№ 2 (95)**

**2020**

Головний редактор — *Г. Г. Четвериков*  
Відповідальний редактор — *І. Д. Вечірська*

Комп'ютерна верстка — *О. Б. Ісаєва*

Рекомендовано Вченою Радою  
Харківського національного університету радіоелектроніки  
(протокол № 05/26 от 00.12.2020)

Адреса редакції:  
Україна, 61166, Харків-166, просп. Науки, 14,  
Харківський національний університет радіоелектроніки, к. 127  
тел. 702-14-77, факс 702-10-13,  
e-mail: [bionics@nure.ua](mailto:bionics@nure.ua)

---

Підписано до друку 03.07.2020. Формат 60 × 84 <sup>1</sup>/<sub>8</sub>. Друк ризографічний.  
Папір офсетний. Гарнітура Newton. Умов. друк. арк. 15,4. Обл.-вид. арк. 15,0.  
Тираж 100 прим.

Віддруковано в редакційно-видавничому відділі ХНУРЕ  
61166, Харків, просп. Науки, 14.