ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СУСПІЛЬСТВО

INFORMATION TECHNOLOGY AND SOCIETY

МАУП

ISSN 2786-5460 (Print)
ISSN 2786-5479 (Online)

МІЖРЕГІОНАЛЬНА АКАДЕМІЯ УПРАВЛІННЯ ПЕРСОНАЛОМ

INTERREGIONAL ACADEMY OF PERSONNEL MANAGEMENT

МАУП

НАУКОВІ ПРАЦІ МІЖРЕГІОНАЛЬНОЇ АКАДЕМІЇ УПРАВЛІННЯ ПЕРСОНАЛОМ

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СУСПІЛЬСТВО

SCIENTIFIC WORKS OF INTERREGIONAL ACADEMY OF PERSONNEL MANAGEMENT

INFORMATION TECHNOLOGY AND SOCIETY

Випуск 3 (5), 2022
Issue 3 (5), 2022

Видавничий дім
«Гельветика»
2022

Рекомендовано до друку Вченою радою
Міжрегіональної Академії управління персоналом
(протокол № 6 від 26 жовтня 2022 року)

Інформаційні технології та суспільство / [головний редактор О. Попов]. – Київ : Міжрегіональна Академія управління персоналом, 2022. – Випуск 3 (5). – 56 с.

Журнал «Інформаційні технології та суспільство» є науковим рецензованим виданням, в якому здійснюється публікація матеріалів науковців різних рівнів у вигляді наукових статей з метою їх поширення як серед вітчизняних дослідників, так і за кордоном.

Редакційна колегія не обов'язково поділяє позицію, висловлену авторами у статтях, та не несе відповідальності за достовірність наведених даних і посилань.

Свідоцтво про державну реєстрацію друкованого засобу масової інформації
«Інформаційні технології та суспільство» Серія КВ № 24815-14755Р від 27.04.2021 р.

Відповідно до Наказу МОН України № 1290 від 30 листопада 2021 року (додаток 3) журнал включено до Переліку наукових фахових видань України (категорія Б) зі спеціальностей 121 – Інженерія програмного забезпечення, 122 – Комп'ютерні науки, 123 – Комп'ютерна інженерія, 124 – Системний аналіз, 125 – Кібербезпека, 126 – Інформаційні системи та технології.

Усі електронні версії статей журналу оприлюднюються на офіційній сторінці видання
http://journals.maup.com.ua/index.php/it

Статті у виданні перевірені на наявність плагіату за допомогою програмного забезпечення StrikePlagiarism.com від польської компанії Plagiat.pl.

Recommended for publication
by Interregional Academy of Personnel Management
(Minutes No. 6 dated 26 October 2022)

Information Technology and Society / [chief editor Oleksandr Popov]. – Kyiv : Interregional Academy of Personnel Management, 2022. – Issue 3 (5). – 56 p.

Journal «Information Technology and Society» is a peer-reviewed scientific edition, which publishes materials of scientists of various levels in the form of scientific articles for the purpose of their dissemination both among domestic researchers and abroad.

Editorial board do not necessarily reflect the position expressed by the authors of articles, and are not responsible for the accuracy of the data and references.

Print media registration certifi cate «Information Technology and Society»
series KV No. 24815-14755P dated 27.04.2021

According to the Decree of MES No. 1290 (Annex 3) dated November 30, 2021, the journal was included in the List of scientific professional publications of Ukraine (category B) in specialties 121 – Software engineering, 122 – Computer sciences, 123 – Computer engineering, 124 – Systems analysis, 125 – Cybersecurity, 126 – Information systems and technologies.

All electronic versions of articles in the collection are available on the official website edition
http://journals.maup.com.ua/index.php/it

The articles were checked for plagiarism using the software
StrikePlagiarism.com developed by the Polish company Plagiat.pl.

ЗМІСТ

CONTENTS

УДК 35.077:347.97/99:004.8
DOI https://doi.org/10.32689/maup.it.2022.3.1

Микола ВАСИЛЕНКО
доктор фізико-математичних наук, доктор юридичних наук, професор, професор кафедри кібербезпеки, Національний університет «Одеська юридична академія», вул. Рішельєвська, 28, Одеса, Україна, індекс 65011 (vasylenko.it@journals.maup.kiev.ua)

ORCID: http://orcid.org/0000-0002-8555-5712

Валерія СЛАТВІНСЬКА
викладач кафедри кібербезпеки, Національний університет «Одеська юридична академія», вул. Рішельєвська, 28, Одеса, Україна, індекс 65011 (slatvinskaya_valeriya@ukr.net)

ORCID: https://orcid.org/0000-0002-6082-981X

Nikolai VASILENKO
Doctor of Physical and Mathematical Sciences, Doctor of Law, Professor, Professor of the Department of cybersecurity, National University «Odessa Law Academy», 28 Richelevskaya str., Odessa, Ukraine, postal code 65011 (vasylenko.it@journals.maup.kiev.ua)

Valeriia SLATVINSKA
Assistant Professor of the Department of cybersecurity, National University «Odessa Law Academy», 28 Richelevskaya str., Odessa, Ukraine, postal code 65011 (slatvinskaya_valeriya@ukr.net)

Бібліографічний опис статті: Василенко, М., Слатвінська, В. Кібернетичний захист операційних систем (аналітичне оглядове дослідження). *Інформаційні технології та суспільство*. 2022. Вип. 3 (5), 6–13. DOI:

Bibliographic description of the article: Vasilenko, M., Slatvinska, V. (2022). Kibernetychnyi zakhyst operatsiinykh system (analitychne ohliadove doslidzhennia) [Cyber security of operating systems (analytical and review research)]. *Informatsiini tekhnolohii ta suspilstvo – Information technology and society*, 3 (5), 6–13. DOI:

КІБЕРНЕТИЧНИЙ ЗАХИСТ ОПЕРАЦІЙНИХ СИСТЕМ (АНАЛІТИЧНЕ ОГЛЯДОВЕ ДОСЛІДЖЕННЯ)

Кібернетичний захист операційних систем (ОС) обговорюється як оглядове дослідження. Оскільки ОС складається з ядра ОС та базового набору прикладних програм йдеться про їх захист в процесі функціонування системи. Однак організація ефективного та надійного захисту ОС неможлива за допомогою одних лише програмно-апаратних засобів, а потребує цілий комплекс заходів. Будь-який комп'ютер складається з чотирьох основних компонентів: центрального процесора; оперативної пам'яті, яка тимчасово зберігає оброблювану інформацію, програмні коди та результати обробки; пристрою введення/виводу, системної магістралі, що визначає механізм взаємодії зазначених вище компонентів. При цьому важко усвідомити усю безліч дій, що виконують і фіксують ОС, яка потребує захисту від внутрішніх та зовнішніх небезпек. Існують певні послуги, що надають ОС в захищеному (незахищеному) режимі. Сформульовані послуги, що надаються в цих умовах типовими ОС, до яких відносять наступні: розробка програм, виконання програм, доступ до пристроїв вводу-виводу; контрольований доступ до файлів; системний доступ; виявлення помилок та їх обробка; облік використання ресурсів; структура системи команд; інтерфейс прикладного програмування; а також бінарний інтерфейс програми. Можливість доступу до об'єктів ОС визначається не тільки архітектурою ОС, а й поточної безпековою ситуацією доступу до об'єкта. Показано, як у практично значущих ситуаціях захищена ОС зазвичай містить засоби управління доступом користувачів до різних ресурсів, засоби перевірки справжності користувача, що починає роботу з ОС, а також застосовувати засоби реєстрації дій користувачів потенційно небезпечних з точки зору безпеки. Розглянемо типові загрози безпеці ОС мобільного пристрою, які суттєво відрізняються від аналогічних загроз для ОС персонального комп'ютера або мережевого сервера. Констатується в результаті обговорень, що основною проблемою забезпечення безпеки ОС все ж таки залишається контроль доступу до ресурсів системи.

Ключові слова: *обчислювальна машина, мобільний пристрій, операційна система, доступ, захист, безпека.*

CYBER SECURITY OF OPERATING SYSTEMS (ANALYTICAL AND REVIEW RESEARCH)

Cyber security of operating systems (OS) is discussed as a survey research. Since the OS consists of the OS kernel and the basic set of applications, it is about their protection in the process of system operation. However, the organization of effective and reliable OS protection is impossible with the help of software and hardware alone, but requires a whole range of

measures. Any computer consists of four main components: the central processor; RAM, which temporarily stores the processed information, program codes and processing results; input/output device, system backbone, which determines the mechanism of interaction of the above components. At the same time, it is difficult to realize all the many actions performed and recorded by the OS, which needs protection from internal and external dangers. There are certain services that provide OS in protected (unprotected) mode. The services provided in these conditions by typical operating systems are formulated, which include the following: program development, program execution, access to input/output devices; controlled access to files; system access; error detection and processing; resource usage accounting; command system structure; application programming interface; and binary program interface. The possibility of access to OS objects is determined not only by the OS architecture, but also by the current security situation of access to the object. It is shown how, in practically significant situations, a secure OS usually contains means of controlling user access to various resources, means of authenticating a user who starts working with the OS, as well as applying means of registering user actions that are potentially dangerous from a security point of view. Consider typical security threats to the OS of a mobile device, which are significantly different from similar threats to the OS of a personal computer or network server. It is stated as a result of the discussion that the main problem of OS security is still the control of access to system resources.

Key words: *computer, mobile device, operating system, access, protection, security.*

Актуальність проблеми. Виходячи із загальних положень кібербезпеки, операційну систему слід вважати захищеною, якщо вона передбачає засоби захисту від основних загроз конфіденційності, цілісності та доступності інформації, актуалізованих з урахуванням особливостей експлуатації конкретного екземпляра операційної системи (ОС, англ. OS).

Адекватною безпековою політикою можна вважати таку політику безпеки, яка забезпечує достатній рівень захищеності ОС. Слід особливо відзначити, що адекватна безпекова політика не обов'язково є тією політикою безпеки, при якій досягається максимально можлива захищеність системи.

З іншого боку, користувачами ОС виступають особи, в інтересах яких здійснюється обробка даних обчислювальних систем (виконання складний розрахунків, управління виробничими процесами і т. д.). Основним технічним засобом для цього стали електронно-обчислювальні машини, включаючи комп'ютери, які здатні ефективно обробляти будь-які види інформації (числову, текстову, табличну, графічну, відео, звукову), заздалегідь перетворивши її в цифрову форму. Програмні засоби керуються наборами цифрових кодів (інструкцій), що управляють технічними засобами залежно від алгоритмів рішення конкретних інформаційний завдань користувача. Отже, для роботи такої системи потрібен комплекс програм, що виконує керування апаратною складовою комп'ютера та забезпечує керування обчислювальним процесом і організовує взаємодію з користувачем. Таким комплексом програм стала ОС, яка звичайно складається з ядра ОС та базового набору прикладних програм. Однак організація ефективного та надійного захисту ОС неможлива за допомогою одних лише програмно-апаратних засобів. Ці засоби обов'язково мусять входити в комплексну систему захисту ОС, надійно їх захищаючи. При цьому комплексний оглядовий підхід до цієї проблеми в періодичній літературі майже не висвітлений, хоча спеціалісти одностайні в тому, що без постійної кваліфікованої підтримки з боку адміністратора навіть найнадійніший програмно-апаратний захист обертається фікцією.

Аналіз останніх досліджень і публікацій. ОС за останні півстоліття змінилися до невпізнання: мінилася оболонка, манера програмування ОС, функціональність і можливість використовувати як для особистих потреб, так і для систем управління на всіляких виробництвах. При цьому ОС залишаються невід'ємною частиною обчислювальних систем. Звичайний користувач навіть не уявляє, які складні процеси відбуваються одночасно у системі, дозволяючи, наприклад читати новини в Інтернеті. Тому, варто враховувати всю складність створення нової та неповторної ОС. Очевидно, що змінилися підходи щодо захищеності таких систем.

Слід наголосити на тому, що ОС представляють собою громіздке програмне забезпечення, що складається з мільйонів рядків. Над розробкою нової системи працюють цілі команди фахівців, проводячи за роботою багато часу, оскільки ОС здебільшого визначається як величезний інженерний проект, який можна порівняти з будівництвом чогось дуже значущого.. Величезні блоки програмного коду часто не можуть бути відокремленими одиницями та постійно взаємодіють з іншими блоками інформації, ускладнюючи завдання, що стоять перед розробниками. Існує досить велика кількість різних операційних систем: від довгожителів, таких як UNIX та його клони, до дуже нових і маловідомих систем. В той же час робіт, присвячених захисту та безпеці операційний систем, небагато. Деякий базовий матеріал викладено в підручниках (див., наприклад [1]). Однак сьогодення пред'являє більш високі вимоги та перспективніші методи захисту ОС ніж це було раніше [2]. В книзі [3] розглядаються загальні концепції безпеки, включаючи принципи інформаційної безпеки, стандарти, регулювання та дотримання; автентифікація, авторизація та облік; та контроль доступу. Аналіз звітів і публікацій про виявлені вразливості у системі Windows і патчів для їх усунення, висвітлені, наприклад на офіційних сайті Microsoft, дають зрозуміти напрями підвищення забезпечення безпеки операційної системи [4-6]. В статті [7]

визначено, що базовим підходом щодо безпеки операційних систем виступає процес «загартовування операційної системи». Показано приклади реалізації Блокчейн для перевірки сертифікатів, враховуючи деякі із варіацій перевірок, які реалізовано на Python 3.0. Автори [7] стверджують: «Використання загартовування системи, є одним з дієвих та комплексних підходів щодо забезпечення інформації безпеки, що дозволить своєчасно виявляти вразливості та своєчасно реагувати на порушення базових властивості операційної системи». В роботі [8] зазначено складність і невичерпність проголошених завдань, які потребують значного наукового опрацювання. Система виявлення вторгнень з використанням технології зв'язаних списків (блокчейн) в процесі реалізації програмного забезпечення обговорювалася також в [9-12]. При обговоренні захисту ОС різні автори фактично розглядають окремі питання щодо проблеми, не виходячи за межі конкретного обговорення. В свою чергу, це ще більше ускладнює і так досить складні питання генези їх захисту. Тому, враховуючи величезну широту і глибину проблеми, а також її багатогранність, наступні дослідження потребують всебічного оглядового аналізу, який і пропонується в наведеній статті.

Метою статті є загальний аналітичний огляд стану захисту ОС в динаміці кібернетичного розуміння їх безпеки.

Виклад основного матеріалу. Важко не погодитися з тим, що будь-який комп'ютер складається з чотирьох основних компонентів: центрального процесора, що керує іншими компонентами та викидає функції обробки інформації; оперативної пам'яті, яка тимчасово зберігає оброблювану інформацію, програмні коди та результати обробки; пристрою введення/виводу, що переміщує інформацію від користувача; системної магістралі, що визначає механізм взаємодії зазначених вище компонентів. При цьому важко усвідомити усю безліч дій, що виконують і фіксують ОС. Однак очевидним для фахівців є і те, що кожен користувач системи має доступ тільки до тих об'єктів ОС, до яких йому надано доступ відповідно до поточної політики безпеки. Доступ користувачів до інших суб'єктів доступу може бути довільно обмежений. Ключовим словом в даному визначенні є слово «довільно». Однак, говорячи про доступ до ОС, треба знати саме ті послуги, що надають ті ж самі ОС. До послуг, що надаються типовими ОС можна віднести ті, які наводяться нами нижче.

• **Розробка програм.** Сприяючи програмісту при розробці програм, операційна система надає йому різноманітні інструменти та служби, наприклад, редактори або відладчики. Зазвичай ці служби реалізовані у вигляді програм утиліт, які підтримуються операційною системою, хоч і не входять до її ядра.

Такі програми називають інструментами розробки прикладних програм.

• **Виконання програм.** Для виконання програми потрібно здійснити ряд дій. Слід завантажити в основну пам'ять команди та дані, ініціалізувати пристрої вводу-виводу та файли, а також підготувати інші ресурси. ОС виконує всю цю рутинну роботу замість користувача.

• **Доступ до пристроїв вводу-виводу.** Для керування роботою кожного пристрою вводу-виводу потрібен особливий набір команд або контрольних сигналів. ОС надає користувачеві одноманітний інтерфейс, який приховує всі ці деталі та забезпечує програмісту доступ до пристроїв введення-виведення за допомогою простих команд читання та запису.

• **Контрольований доступ до файлів.** Працюючи з файлами управління з боку операційної системи передбачає як глибоке розуміння природи пристроїв вводу-виводу (дисковода, стрічкопротяжного пристрою), а також знання структур даних, записаних у файлах Багатокористувацькі ОС, крім того, можуть забезпечувати роботу механізмів захисту під час звернення до файлів.

• **Системний доступ.** Операційна система керує доступом до спільно використовуваної або загальнодоступної обчислювальної системи в цілому, а також окремих системних ресурсів. Вона повинна забезпечувати захист ресурсів та даних від несанкціонованого використання, а також вирішувати конфліктні ситуації.

• **Виявлення помилок та їх обробка.** При роботі комп'ютерної системи можуть відбуватися різноманітні збої. До них належать внутрішні та зовнішні помилки, що виникли в апаратному забезпеченні (наприклад, помилки пам'яті,

відмова або збій пристроїв). Можливі (різні) програмні помилки (такі, як арифметичне переповнення, розподіл на нуль, спроба звернутися до осередку пам'яті, доступ до якого заборонено, або неможливість виконання запиту програми). У кожному з цих випадків операційна система має виконати дії, що мінімізують вплив помилки на роботу програми. Реакція ОС на помилку може бути різною – від простого повідомлення про помилку до аварійної зупинки програми, що її викликала.

• **Облік використання ресурсів.** Хороша операційна система повинна мати засоби обліку використання різних ресурсів та відображення параметрів продуктивність. Ця інформація вкрай важлива у будь-якій системі, особливо у зв'язку з необхідністю подальших поліпшень та налаштування обчислювальної системи підвищення її продуктивності. У розрахованих на багато користувачів системах ця інформація може застосовуватися для виставлення рахунків.

• **Структура системи команд** (instruction set architecture – ISA). Визначає набір команд машинної мови, які можуть виконувати комп'ютер. Цей інтерфейс є межею між апаратним та програмним забезпеченням. Зверніть увагу, що і прикладні програми, і утиліти можуть отримати безпосередній доступ до ISA. Для цих програм доступне підмножина команд (користувацька ISA). ОС має доступ до додаткових команд машинної мови, що належать до керування ресурсами системи (системна ISA).

• **Бінарний інтерфейс програми** (application binary interface – ABI). ABI визначає стандарт бінарної переносимості між програмами. ABI визначає інтерфейс системних викликів операційної системи та апаратних ресурсів та служб, доступних у системі через користувальницьку ISA.

• **Інтерфейс прикладного програмування** (application programming interface – API). API забезпечує програмі доступ до апаратних ресурсів та служб, доступних у системі через користувальницьку ISA з бібліотечними викликами мовою високого рівня. Зазвичай будь-які системні дзвінки виконуються через бібліотеки. Застосування API забезпечує легку переносимість прикладного програмного забезпечення інші системи, що підтримують той же API шляхом перекомпіляції.

Якщо правила обмежують доступ суб'єктів до деякого елементу ОС, вони визначені жорстко і не допускають зміни з плином часу. Іншими словами, можливість доступу до об'єктів ОС визначається не тільки архітектурою ОС, а й поточної безпековою ситуацією (методом) доступу до об'єкта. Наприклад, для файлів можуть бути визначені: методи доступу «читання», «запис» і «додавання» (дописування інформації в кінець файлу). Суб'єктом доступу тут слід називати будь-яку сутність, здатну ініціювати виконання операцій над об'єктами (звертатися до об'єктів за деякими методами доступу).

У практично значущих ситуаціях захищена ОС зазвичай містить засоби управління доступом користувачів до різних ресурсів, засоби перевірки справжності користувача, що починає роботу з ОС, а також застосовувати засоби реєстрації дій користувачів потенційно небезпечних з точки зору безпеки. Крім того, захищена операційна система повинна містити засоби протидії випадковому або навмисному виведенню операційної системи з ладу.

Вважається визнаною політикою безпеки набір норм, правил і практичних прийомів, які регламентують порядок зберігання та обробки цінної інформації. У застосуванні до операційної системи політика безпеки визначає те, які користувачі можуть працювати з операційною системою, які мають доступ до конкретних об'єктів операційної системи, які події повинні реєструватися в системних журналах і т.д.

Важливим ланцюжком з ОС стає ідентифікація та аутентифікація. Жоден користувач не може почати роботу з ОС, який не ідентифікував себе і не надавши системі аутентифікуючу інформацію, яка підтверджує, що користувач дійсно є тим, за кого він себе видає. (В захищеної операційної системи будь-який суб'єкт доступу, перед тим як почати роботу з системою, повинен пройти ідентифікацію, аутентифікацію і авторизацію. Ідентифікація суб'єкта доступу полягає у тому, що суб'єкт повідомляє операційній системі ідентифікаційну інформацію про себе (ім'я, обліковий номер і т. д.) і таким чином ідентифікує себе. Так,конкретна ОС може надавати захист різного ступеню для різних об'єктів, користувачів та додатків. У захищеній операційній системі будь-який суб'єкт доступу, перед тим як розпочати роботу з системою, повинен пройти ідентифікацію, автентифікацію та авторизацію.

Аутентифікація суб'єкта доступу полягає в тому, що суб'єкт надає системі крім ідентифікаційної інформації ще й аутентифікаційну інформацію, що підтверджує, що він дійсно є тим суб'єктом доступу, до якого належить ідентифікаційна інформація.

Авторизація суб'єкта доступу відбувається після успішної ідентифікації та аутентифікації. При авторизації суб'єкта операційна система виконує дії, необхідні для того, щоб суб'єкт міг почати роботу в системі – завантажує індивідуальні налаштування користувача, запускає програму-оболонку і т.п.

Авторизація суб'єкта не відноситься безпосередньо до захисту операційної системи. У процесі авторизації вирішуються суто технічні завдання, пов'язані з організацією початку роботи в системі вже ідентифікованого та автентифікованого суб'єкта доступу. Зауважимо, що у ряді джерел термін «авторизація» використовується як синонім терміна «управління доступом». Це вносить певну плутанину в термінологію у цій галузі.

З точки зору забезпечення безпеки комп'ютерної системи процедура аутентифікації є дуже відповідальною. Якщо порушник зумів увійти в систему від імені іншого користувача, тим самим порушник легко отримує доступ до всіх об'єктів системи, до яких має доступ цей користувач. Якщо при цьому процесі роботи порушника з операційною системою підсистема аудиту генерує повідомлення про потенційно небезпечних подіях, то в журнал аудиту буде внесено не ім'я порушника, а ім'я користувача, від імені якого порушник працює у системі.

Хоча аутентифікація може здійснюватися як фізичних користувачів, так псевдокористувачів, найбільший інтерес з погляду забезпечення інформаційної безпеки представляє аутентифікація фізичних

користувачів. Якщо в системі реалізується адекватна безпекова політика, фізичний користувач просто не може увійти в систему від імені псевдокористувача. Якщо псевдокористувач має великі повноваження, вхід порушника в систему від імені псевдокористувача дає порушнику великі можливості для здійснення несанкціонованого доступу, проте на практиці здійснити таку атаку зазвичай вкрай важко. Тому надалі ми розглядатимемо аутентифікацію тільки звичайних користувачів.

Зазвичай підсистема аутентифікації операційної системи будується за однією з трьох схем: парольна автентифікація; аутентифікація з використанням зовнішніх носіїв інформації; біометрична автентифікація.

Можливе використання комбінацій двох або навіть усіх трьох схем аутентифікації в одній системі.

В зв'язку з наведеними вище варіантами захищеності ОС виникає питання про їх характеристики і створення підходів до них. Відомі два основні підходи до створення захищених ОС – фрагментарний та комплексний. При фрагментарному підході спочатку створюється захист від однієї загрози, потім від іншої і т.д. Прикладом фрагментарного підходу може бути с ситуація, коли за основу береться незахищена операційна система, на неї встановлюються антивірусний пакет, потім система шифрування, система реєстрації дій користувачів і т.д.

Основний недолік фрагментарного підходу очевидний – при застосуванні цього підходу підсистема захисту операційної системи є набір розрізнених програмних продуктів, як правило, вироблених різними виробниками. Ці програмні засоби працюють незалежно один від одного, організувати їхню тісну взаємодію практично неможливо. Крім того, окремі елементи такої підсистеми захисту можуть некоректно працювати в присутності один одного, що призводить до різкого зниження загальної надійності системи. Оскільки підсистема захисту, створена з урахуванням фрагментарного підходу, перестав бути невід'ємною компонентою операційної системи, при відключенні окремих захисних функцій у результаті несанкціонованих дій користувача-порушника інші елементи операційної системи продовжують нормально працювати, що ще більше знижує надійність захисту.

При комплексному підході до організації захисту захисні функції вносяться в операційну систему ще на етапі проектування архітектури операційної системи та є її невід'ємною частиною. Окремі елементи підсистеми захисту, створеної на основі комплексного підходу, тісно взаємодіють один з одним при вирішенні різних завдань, пов'язаних з організацією захисту інформації. Оскільки вся підсистема захисту розробляється та тестується в сукупності, конфлікти між її окремими компонентами практично неможливі. Підсистема захисту, створена на основі комплексного підходу, може бути влаштована так, що при фатальних збоях у функціонуванні її ключових елементів підсистеми захисту вона викликає аварійне завершення роботи операційної системи, що не дозволяє порушнику відключати захисні функції системи. З використанням фрагментарного підходу така організація підсистеми захисту неможлива.

Як правило, підсистему захисту ОС, створену на основі комплексного підходу, проєктують так, що окремі її елементи є замінними і відповідні програмні модулі можуть бути замінені іншими модулями, що реалізують передбачений і належним чином документований інтерфейс взаємодії відповідного програмного модуля коїться з іншими елементами підсистеми захисту. Відома і така ситуація, коли існує відсутність захисту. Цей варіант підходить, коли відповідні процедури виконуються за часом окремо.

Існує також така ситуація як ізоляція. Кожний процес працює окремо від інших процесів, не використовуючи сумісно ресурси і не обмінюючись інформацією. Кожний процес має свій адресний простір, свої файли та інші об'єкти.

Можливими є повний розподіл або повна його відсутність. Власник об'єкту (файлу, сегмента пам'яті) об'являє його відкритим або закритим. У першому випадку доступ до об'єкта може отримати будь-який процес; у другому – доступ до цього об'єкта надається тільки власнику.

Спільне використання з обмеженнями доступу полягає в тому, що ОС перевіряє дозволеність доступу кожного окремого користувача до кожного окремого об'єкта. В цьому сенсі ОС виступає в якості охоронця, гарантуючи, що доступ до об'єкта отримають тільки авторизовані користувачі.

Спільне використання за допомогою динамічних можливостей. Цей варіант розширює концепцію контролю доступу, дозволяючи динамічно створювати права спільного використання об'єкта.

Існує також така ситуація, коли йдеться про обмежене використання об'єкта. При цьому обмежується не стільки доступ до об'єкта, скільки його використання. Наприклад, користувачеві дозволено переглядати важливий документ, але не роздруковувати, або користувач має доступ до бази даних, може брати з неї статистичні зведення, але не має можливості визначити значення певних величин.

Необхідно зауважити наступне: в ОС має підтримуватися баланс між можливостями спільного використання компонентів комп'ютерної системи в цілому, що сприяє підвищенню ефективності її використання, і стає ступенем захищеності ресурсів окремих користувачів.

При всій багатогранності проблеми не можна не відзначити негативний вплив на захищеність ОС незаконного використання привілеїв та щкідливе програмне забезпечення. Останнє розглядалося нами раніше у [13], тому це питання нами вже не обговорюється.

Наприкінці розглянемо типові загрози безпеці ОС мобільного пристрою, які суттєво відрізняються від аналогічних загроз для ОС персонального комп'ютера або мережевого сервера. Деякі загрози, незначні для звичайних комп'ютерів, стають дуже небезпечними для мобільних пристроїв, і навпаки. Наприклад, крадіжка мобільного телефону кишеньковим злодієм є набагато серйознішою загрозою, ніж крадіжка сервера злодієм-домушником. Програмна закладка, впроваджена в ОС персонального комп'ютера і що отримала доступ до електронних банківських рахунків користувача, зазвичай має дуже обмежені можливості з несанкціонованого переказу коштів з цих рахунків. Але програмна закладка, внесена в ОС мобільного пристрою, елементарно вирішує це завдання шляхом несанкціонованого замовлення дорогих SMS-послуг з телефонного номера, контрольованого порушником, або (рідше) шляхом імітації голосового дзвінка на платний номер. З іншого боку, загрози, пов'язані з одночасним доступом кількох користувачів до одного екземпляра ОС, для мобільних ОС, як правило, неактуальні.

До найбільш актуальних загроз безпеки мобільних ОС зазвичай відносять такі:

• розкриття конфіденційної інформації внаслідок втрати чи крадіжки мобільного пристрою;

• несанкціоноване замовлення дорогих послуг програмною закладкою, впровадженою в операційну систему мобільного пристрою;

• розкриття конфіденційної інформації в результаті перехоплення бездротового мережевого трафіку, що генерується мобільним пристроєм;

• несанкціонований збір програмною закладкою персональних даних користувача мобільного пристрою;

• втрата даних, що зберігаються на мобільному пристрої.

В даний час для мобільних операційних систем розробляється величезна кількість шкідливого програмного забезпечення. За даними [14], близько чверті всіх додатків, написаних для ОС Android, є шкідливими. Кількість шкідливих програм для Android зростає експонентно, щороку воно збільшується в 2-16 разів. Така велика різниця в цифрах, що даються різними джерелами, пояснюється тим, що межа між множинами шкідливих і неушкоджених мобільних додатків дуже умовна. Хорошим прикладом «прикордонних» додатків є програми, що дозволяють легальному власнику телефону виявити вкрадений у нього телефон шляхом непомітного перехоплення та доставки на задану адресу електронної пошти інформації про зроблені дзвінки, відправлені та отримані SMS, географічне розташування тощо.

Практичні дані свідчать, що компанія Google, на відміну від Apple, декларує максимальну відкритість своєї ОС Android. Кожна програма для Android повинна бути підписана розробником, але завірення цього підпису будь-яким засвідчуючим центром не потрібно. До каталогу Google Play та інтернет-магазину Android Market допускаються всі додатки, крім свідомо шкідливих і свідомо непрацездатних. Іноді це призводить до скандальних ситуацій, що негативно впливають на репутацію торгової марки Android. Наприклад, Android-версія програми eMobiStudio MemoryUp, що добре зарекомендувала себе на платформах Symbian, BlackBerry і Windows Mobile, несанкціоновано видалила через програмну помилку дані адресних книг кількох тисяч користувачів, які встановили цю програму.

На наш погляд, відкритість операційної системи Android не слід перебільшувати. На відміну від більшості розробників універсальних ОС, Google зберігає за собою контроль над усіма додатками, що працюють на всіх примірниках операційної системи Android. При необхідності компанія Google, як і Apple, може одночасно видалити всі екземпляри будь-якої заданої програми, встановленої на всіх мобільних пристроях зі своєю ОС.

Довгий час забезпечення безпеки мобільних ОС розглядалося їх розробниками як другорядне, низькопріоритетне завдання. Але ситуація поступово змінюється на краще, з кожною наступною версією мобільні операційні системи стають все більш захищеними. При цьому розробники мобільних ОС широко використовують рішення, апробовані на універсальних ОС.

Висновки та перспективи подальших досліджень. Таким чином, основною проблемою забезпечення безпеки ОС залишається контроль доступу до ресурсів системи. Для вирішення цієї проблеми ОС повинна мати власний допоміжний захист. Здебільшого зустрічаються такі атаки на ОС як сканування системи та спроби злому пароля. Виділяють дві основні загрози: крадіжка та добірка, а сам процес не повинен мати багато привілеїв. Існують фрагментарний та комплексний підхід створення захищеної ОС. При фрагментарному підході система насамперед захищається від однієї погрози, тільки потім від іншої. При комплексному підході захист системи вкладається в ОС під час проектування архітектури і є частиною ОС. Плюсом такого підходу є те, що захист, створений комплексним підходом, взаємодіє

один з одним і надійніше допомагає організовувати захист інформації. Під час розробки підсистема проходить перевірку на сумісність та конфліктів між частинами підсистеми захисту не відбувається. Головним здобутком такої підсистеми захисту є те, що при збоях, викликаних зловмисником, система не дозволяє вимкнути систему захисту. В мобільних ОС широко використовують рішення, апробовані на універсальних ОС.

Список використаних джерел:

1. Т*анненбаум Э.* Современные операционные системы. СПб. : Питер, 2006, 1040 с.

2. Проскурин В. Г. Защита в операционных системах. Учебное пособие для вузов. М. : Горячая линия – Телеком, 2014. 192 с.

3. Derrick Rountree, Security for Microsoft Windows System Administrators Introduction to Key Information Security Concepts, Syngress, 211 p., 2011.

4. Artes, N.O., and S.M. Elsakov. "Protection System of Applications on 'Windows' Platform on the Basis of Activity Profile". *Journal of Computational and Engineering Mathematics.* 3, no. 3 (2016): 3–9. https://doi.org/10.14529/jcem160301.

5. HU, Hong-yin, Feng YAO, and Cheng-wan HE. "Solution of Windows Files Security Protection Based on File System Filter Driver". *Journal of Computer Applications.* 29, no. 1 (June 25, 2009): 168–171. https://doi.org/10.3724/sp.j.1087.2009.00168.

6. Küenzlen, Jürgen, Eckehard Scheller, and Hermann Hamm. "Fixing of Windows with Fall Protection / Befestigung von Absturzsichernden Fensterelementen". Mauerwerk 20, no. 6 (December 2016): 423–444. https://doi.org/10.1002/dama.201600714.

7. Ільєнко А.В., Ільєнко С.С., Куліш Т.М. Перспективні методи захисту операційної системи WINDOWS. *Кібербезпека. Освіта, наука, техніка.* № 4(8). 2020. С. 124- 132.

8. Зерко А., Оксіюк О. Аналіз питань захисту інформації в операційних системах на прикладах захищених операційних систем. Геометричне моделювання та інформаційні технології. 2017. № 1 (3). С. 53-55.

9. Бойко В.Д., Василенко М.Д. Відкриті системи і протоколи взаємодії в контексті кібербезпеки «розумного міста». *Безпека та виклики сучасності: ризики та кібербезпека в період пандемії безпека в сучасному світі* : матеріали II Всеукраїнської науково-практ. конф. (м. Одеса, 20 листоп. 2020 р.) / за ред. О. В. Дикого. Одеса : Видавничий дім «Гельветика», 2020. С. 99-106.

10. Бойко В. Д., Василенко М. Д., Слатвінська В. М. Версіонування файлової системи для боротьби з програмами-вимагачами. Тези VIII Міжнар. науково-техн. конф. *«Інформатика, управління та штучний інтелект»* (м. Харків, 16–19 листоп. 2021 р.). Харків: НТУ «ХПІ», 2021. С. 9.

11. Бойко В.Д., Василенко М.Д., Новіков В.П., Рачук В.О. «Розумне місто» в контексті розвитку технологій блокчейн. *Комунальне господарство міст. Серія : технічні науки та архітектура.* Харків, 2021. Вип. 3 (163). С. 152-158.

12. Бойко В.Д., Василенко М.Д., Слатвінська В.М. Живучість та стійкість функціонування компонентів інформаційних систем розумного міста. Науково-технічний збірник *«Комунальне господарство міст». Серія: технічні науки та архітектура*. Харків, 2021. Вип. 6 (166). С. 20-27.

13. Василенко М.Д., Рачук В.О. Слатвінська В.М. Шкідливі програми в контексті розуміння комп'ютерної вірусології та техніко-правової змагальності: міждисциплінарне дослідження. *Наукові праці Національного університету «ОЮА».* Т. 28. Одеса : 2021. С. 28-36.

14. Android отметила 5-летие. URL: https://www.cybersecurity.ru/os/ 163693.html

References:

1. Tannenbaum Je. (2006). Sovremennye operacionnye sistemy [Modern operating systems]. SPb. : Piter, 1040 [in Russian].

2. Proskurin V. G. (2014). Zashhita v operacionnyh sistemah [Protection in operating systems]. Uchebnoe posobie dlja vuzov. M. : Gorjachaja linija – Telekom, 192 [in Russian].

3. Derrick Rountree (2011). Security for Microsoft Windows System Administrators Introduction to Key Information Security Concepts, Syngress, 211.

4. Artes, N.O., Elsakov S.M. (2016). "Protection System of Applications on 'Windows' Platform on the Basis of Activity Profile." *Journal of Computational and Engineering Mathematics.* 3, 3: 3–9. https://doi.org/10.14529/jcem160301.

5. HU, Hong-yin, Feng YAO, and Cheng-wan HE. (2009). "Solution of Windows Files Security Protection Based on File System Filter Driver." *Journal of Computer Applications.* 29, 1: 168–171. https://doi.org/10.3724/sp.j.1087.2009.00168.

6. Küenzlen, Jürgen, Eckehard Scheller, and Hermann Hamm. (2016). "Fixing of Windows with Fall Protection / Befestigung von Absturzsichernden Fensterelementen." Mauerwerk 20, 6: 423–444. https://doi.org/10.1002/dama.201600714.

7. Ilienko A.V., Ilienko S.S., Kulish T.M. (2020). Perspektyvni metody zakhystu operatsiinoi systemy WINDOWS [Perspective methods of protecting the WINDOWS operating system]. *Kiberbezpeka. Osvita, nauka, tekhnika – Cybersecurity. Education, science, technology*. 4(8). 124-132. [in Ukrainian].

8. Zerko A., Oksiiuk O. (2017). Analiz pytan zakhystu informatsii v operatsiinykh systemakh na prykladakh zakhyshchenykh operatsiinykh system [Analysis of information security issues in operating systems on the examples of secure operating systems]. *Heometrychne modeliuvannia ta informatsiini tekhnolohii – Geometric modeling and information technology*, 1, (3), 53-55. [in Ukrainian].

9. Boiko V.D., Vasylenko M.D. (2020). Vidkryti systemy i protokoly vzaiemodii v konteksti kiberbezpeky «rozumnoho mista» [Open systems and interaction protocols in the context of cybersecurity of the "smart city"]. Bezpeka ta vyklyky suchasnosti: ryzyky ta kiberbezpeka v period pandemii bezpeka v suchasnomu sviti : materialy II Vseukrainskoi nau-

kovo-prakt. konf. (m. Odesa, 20 lystop. 2020 r.) / za red. O. V. Dykoho. Odesa : Vydav nychyi dim «Helvetyka», 99-106. [in Ukrainian].

10. Boiko V. D., Vasylenko M. D., Slatvinska V. M. (2021). Versionuvannia failovoi systemy dlia borotby z prohramamy-vymahachamy [File system versioning to combat ransomware]. Tezy VIII Mizhnar. naukovo-tekhn. konf. «Informatyka, upravlinnia ta shtuchnyi intelekt». (m. Kharkiv, 16 – 19 lystop. 2021 r.). Kharkiv: NTU «KhPI», 9. [in Ukrainian].

11. Boiko V.D., Vasylenko M.D., Novikov V.P., Rachuk V.O. (2021). «Rozumne misto» v konteksti rozvytku tekhnolohii blokchein ["Smart city" in the context of blockchain technology development]. *Komunalne hospodarstvo mist. Seriia: tekhnichni nauky ta arkhitektura – Municipal economy of cities. Series: technical sciences and architecture*. Kharkiv, 3 (163), 152-158. [in Ukrainian].

12. Boiko V.D., Vasylenko M.D., Slatvinska V.M. (2021). Zhyvuchist ta stiikist funktsionuvannia komponentiv informatsiinykh system rozumnoho mista [Survivability and sustainability of smart city information system components]. *Naukovo-tekhnichnyi zbirnyk «Komunalne hospodarstvo mist». Seriia: tekhnichni nauky ta arkhitektura – Municipal economy of cities. Series: technical sciences and architecture*. Kharkiv, 6 (166), 20-27. [in Ukrainian].

13. Vasylenko M.D., Rachuk V.O. Slatvinska V.M. (2021). Shkidlyvi prohramy v konteksti rozuminnia kompiuternoi virusolohii ta tekhniko-pravovoi zmahalnosti: mizhdystsyplinarne doslidzhennia [Malware in the context of understanding computer virology and technical-legal competition: an interdisciplinary study]. *Naukovi pratsi Natsionalnoho universytetu «OIuA» – Scientific works of the National University «OSA»*. T. 28. Odesa: 28-36. [in Ukrainian].

14. Android otmetyla 5-letye [Android celebrated its 5th anniversary]. Retrieved from https://www.cybersecurity.ru/os/ 163693.html [in Russian].

УДК 004.4
DOI https://doi.org/10.32689/maup.it.2022.3.2

Вячеслав ЛІСКІН
кандидат технічних наук, старший викладач кафедри прикладної математики, факультет прикладної математики, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», пр. Перемоги, 37, Київ, Україна, індекс 03056 (liskinslava@gmail.com)

ORCID: 0000-0002-9418-0633

Сергій СИРОТА
кандидат технічних наук, доцент, доцент кафедри прикладної математики, факультет прикладної математики, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», пр. Перемоги, 37, Київ, Україна, індекс 03056 (sergiy.syrot@gmail.com)

ORCID: 0000-0003-0795-167X

Ольга ЧОЛИШКІНА
кандидат технічних наук, доцент, директор Інституту комп'ютерно-інформаційних технологій, Міжрегіональна Академія управління персоналом, вул. Фрометівська, 2, Київ, Україна, індекс 03039 (greenhelga5@gmail.com)

Viacheslav LISKIN
PhD., Department of Applied Mathematic National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37 Peremohy Ave, Kyiv, Ukraine, postal code 03056 (liskinslava@gmail.com)

Sergiy SYROTA
PhD., Associated professor, Department of Applied Mathematic National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37 Peremohy Ave, Kyiv, Ukraine, postal code 03056 (sergiy.syrot@gmail.com)

Olha CHOLYSHKINA
Candidate of Technical Sciences, Associate Professor, Director of Institute of computer information technologies, Interregional Academy of Personnel Management, Frometivska str., 2, Kyiv, Ukraine, postal code 03039 (greenhelga5@gmail.com)

Бібліографічний опис статті: Ліскін, В., Сирота, С., Чолишкіна, О. (2022). Технологія для автоматизації генерації тестових запитань за допомогою аналізу формальних понять. *Інформаційні технології та суспільство*, 3 (5), 14–19. DOI:

Bibliographic description of the article: Liskin, V., Syrota, S., Cholyshkina, O. (2022). Tekhnolohiia dlia avtomatyzatsii heneratsii testovykh zapytan za dopomohoiu analizu formalnykh poniat [Automated test question generation approach using formal concept analysis]. *Informatsiini tekhnolohii ta suspilstvo – Information technology and society*, 3 (5), 14–19. DOI:

AUTOMATED TEST QUESTION GENERATION APPROACH USING FORMAL CONCEPT ANALYSIS

The article is devoted to the E-learning systems content development. In continuation of the previous works of the authors, the technology for the automatic creation of closed-type test questions with multiple correct answers proposed. The proposed technology is based on the set theory model of test question. Grounded Formal Concept Analysis approach for solving this task. This technology helps to create big sets of tests, with the help of which chunk-knowledge will be tested. It was shown, that this approach can help to solve the problem of the e-learning testing students' knowledge and will help to avoid cheating.

Key words: *E-learning, Content, Chunk, Test, Formal Concept Analysis.*

ТЕХНОЛОГІ ДЛЯ АВТОМАТИЗАЦІЇ ГЕНЕРАЦІЇ ТЕСТОВИХ ЗАПИТАНЬ ЗА ДОПОМОГОЮ АНАЛІЗУ ФОРМАЛЬНИХ ПОНЯТЬ

Стаття присвячена розробці контенту систем електронного навчання. У продовження попередніх робіт авторів запропоновано технологію автоматичного створення тестових запитань закритого типу з кількома правильними відповідями. Запропонована технологія базується на теоретико-множинній моделі тестового питання. Обґрунтований підхід аналізу формальної концепції для вирішення цього завдання. Ця технологія дозволяє створювати великі набори тестів, за допомогою яких будуть перевірятися chunk-знання. Показано, що такий підхід може допомогти вирішити проблему перевірки знань студентів електронного навчання та допоможе уникнути списування.

Ключові слова: *Електронне навчання, вміст, чанк, тест, формальний аналіз понять.*

Formulation of the problem. In the context of the COVID-19 pandemic, distance learning systems have been widely developed and used in educational activities. At the present stage of their use, which is accompanied by the constant growth of educational materials and distance courses directly, requirements are put forward for their quality and the ability to adequately assess the results of students.

Testing is an important part of the educational process. A characteristic feature of distance education is the use of information technology in the control of students' knowledge.

A distance course with automated teaching capabilities makes it possible to implement an individual approach to learning, to promote the development of independence, self-discipline, and creative abilities of students, and will provide access to the necessary sources of educational information.

Analysis of the possibilities of traditional testing with a fixed number of tasks and the time to complete them, used in distance learning as the main method of control, led to the conclusion that it is necessary to turn to adaptive testing. By adaptive testing, we mean tests that are formed by the E-learning environment, considering the student's passing of previous tests.

One of technics of adaptive testing was proposed in [1]. Such technics is based on the assumption that the system has a large number of questions related to a given topic but creating such bank of questions is quite laborious. According to the authors of this work, this process can and should be automated.

Denotation of question types and review of existing approaches.

Conventional types of test questions. Let us denote types of test questions:

"Matching questions" provide a list of subjects which corresponds to other subject answers. The respondent must "match" the correct answers with each question.

For example

$$\begin{vmatrix} What\ time\ of\ year\ is\ January\ ? \\ What\ time\ of\ year\ if\ it\ is\ March? \\ What\ season\ in\ September\ is? \end{vmatrix} \rightleftharpoons \begin{vmatrix} summer \\ autumn \\ winter \end{vmatrix}$$

"Multiple Choice" – in response to a question, the respondent chooses from multiple answers. There are two types of multiple-choice questions – single answer where there is only one correct answer. For example:

January is

- *a summer month*
- *a winter month*
- *an autumn month*

Multiple answer where the student must pick all answers that apply.

The summer month are

- *January*
- *March*
- *September*
- *July*

In "True/False" response to a question, the student selects either True or False.

January is a winter month

- *yes*
- *no*

From simple examples above we understand that all questions where almost about the same. And there is possible generate all possible questions of all types automatically. For this purpose, we should formalize the problem, develop an ontology, and propose an efficient algorithm to do this.

Review of existing models and approaches for automatic test creation.

Most test generators give the opportunity to create question without controlling its content. The question is manually entered, answer options are also entered manually, and scores are assigned for each answer. This approach requires the attention of the course developer as well as the time required [2].

The mostly represented approach, as already mentioned, is based on the development of tools for creating and conducting testing without reference to the meaning of tests and the subject area in which training is performed.

There are also known approaches that try to model the subject area and automatically build and verify test questions based on the obtained models. Also, group questions into tests based on the student's answers.

In previous publications, the authors considered technologies for automating the creation of test questions [3]. The approaches considered were based on the so-called concept of educational chunk.

Chunk is a unit of information which is understanding as one part of the memory, as like an array of letters, words or numbers, which are consider as one knowledge.

Fundamental works describing the subject area of the issues themselves have been known since the 50s of the 20-th century [4].

The logic of questions and answers, also known as erotetic logic, can be approached in different ways. The most general approach treats it as a branch of epistemic logic [5; 6] specially devoted to the problems of the logic of questions and answers.

The general idea of this direction is the construction of sets of queries and responses and finding dependencies on these sets. As well as the study of the fact that set-theory operations with over a set of queries will lead to a match in on sets of responses.

Therefore, it becomes necessary to normalize such issues, that is, to isolate the sets of entities, relations between entities and to clarify the restrictions that are imposed on such sets.

However, according to the authors, there is a need to narrow the mathematical base for the development of applied software which can generate questions.

Formal model of test question. Returning to the example above, we see that in fact the same question can be formulated in different phrases.

What time of year is January? What time of year if it is January? What season is in January?

Therefore, it becomes necessary to normalize such issues, that is, to isolate the sets of entities, relations between entities and to clarify the restrictions that are imposed on such sets.

There are published approaches to formalizing this task by creating an ontology of the question itself [7].

The scheme of one line of a matching question can be represented by a chain: *notation – essence1 – link(relation) – essence2,*

$$Q: < N, E^V, I, E^A >, \quad (1)$$

where, $N = \{n_i\}$ *set of* notations, $E^V=V=\{e_i\}$ – set of essences of subject area wich located on the right side of question $E^A=A=\{e_i\}$ (must be noted that right side can be switched with a left one), $L\ \{l_k\}$– links between essences.

Relation I_k can be considered as a mapping from set E^V to set E^A. Such mapping can be one-to-one, one-to-many, or many-to-many type.

In case one-to-one relation we get so called *strict* maching wich means that one essence from right side coresponds to one and only one on the left side denote ↔. In case one-to-many relation we get so called *strong* maching of inclusion type wich means that one essence from right side coresponds to more than one on the left side denote ⇚. Respectively relation many-to-many we get *soft* matching denote ⇆.

We can see that in the case of strict compliance, the question will be formed quite simply. Like it was proposed [3] we get some essences from *V*, coresponding essences from *A.* And using sets *N, L,* build oll possible strict matching question. Getting one of essence from *V* and one coresponding essence from *A* than few not corresponding from *A* (distructors) we obtain a set of "multiple choice one anwer" quastions.

In simple cases, it is easy enough to construct sets with relations of this type. But in practice, on existing subject areas that contain more than two dozen entities, it is quite difficult not to miss the characteristics of these relations.

Using Formal Concept Analisys for generating.

Let us consider the set of essences *V* and the set of essences *A* with an arbitrary relation $I \subseteq V \times A$, such that *pIa*, where $p \in V$, $a \in A$, if and only if *a* is the essence *p* is in relation *I* Then triplet $K = (V, A, I)$ is called the *formal context.* The the relation can be perfommed by binary matrix with rows from *V* and columns from *A*. Let define the connection [8]:

$$P' := \{y \in A \mid xIy \text{ for all } x \in P\}, \text{ for } P \subseteq V$$
$$G' := \{x \in V \mid xIy \text{ for all } y \in G\}, \text{ for } G \subseteq A. \quad (2)$$

Then the pairs (P, G) satisfying $P \subseteq V$, $G \subseteq A$, $P' = G$, $G' = P$ are called *the formal concepts* of the formal context $K = (V, A, I)$. Every object $p \in P$ have all relation *L* to *G.* So, the formal concept is the set of objects from domain such that every one of them have all attributes from certain subset of attributes of that objects.

The set of formal concepts (P, G), where $P \subseteq V$, $G \subseteq A$, is partially ordered by the relation: $(P_1, G_1) \leq (P_2, G_2)$, if $P_1 \subseteq P_2$ and $G_2 \subseteq G_1$, and form a complete lattice $L(K)$, called *the concept lattice* of the context *K* [9]. The pair (P_1, G_1) is called the subconcept of the concept (P_2, G_2), and the pair (P_2, G_2) is called the superconcept of the concept (P_1, G_1).

For the sets $\wp(V)$ и $\wp(A)$ ($\wp$ denotes that $\wp(X)$ – is set of all subsets *X*) mappings G |→ G' и P |→ P', are Galois connections having next properties [14]:

1) If $G_1 \subseteq G_2$ than $G_1' \supseteq G_2'$ for any $G_1, G_2 \subseteq V$, i.e. if the set G_1 is included in a set of G_2, than a set G_2' is included in a set G_1';

2) If $P_1 \subseteq P_2$ than $P_1' \supseteq P_2'$ for any $P_1, P_2 \subseteq A$, т.е., i.e. if the P_1 is included in a set P_2, than set P_2' is included to set P_1';

3) $G \subseteq G''$ and $G' = G'''$ for $G \subseteq V$;

4) $P \subseteq P''$ and $P' = P'''$ for $P \subseteq A$.

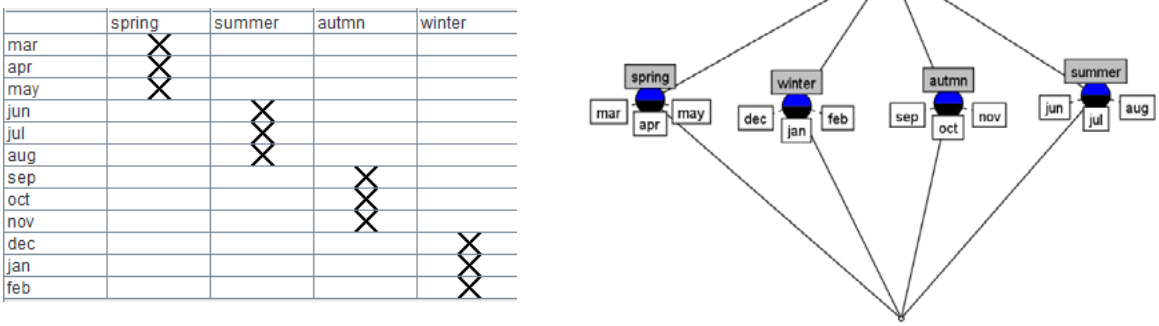
It follows from properties 1–4 [10], that double applying of operator ’’ has properties of closure operator, and connections G |→ G’ и P |→ P’ define the dual isomorphism between complete lattices of closed subsets *V* and *A*. According to properties (1)-(4), concept (*G, P*) also be denoted (P’, G’) = (G’’, G’) = (P’, P’’).

I context *K* = (*V, A, I*) set of concepts is preordered by relation [5, 14]:

$$(G_1, P_1) \leq (G_2, P_2), \text{if } G_1 \subseteq G_2 \text{ и } P_1 \supseteq P_2. \quad (3)$$

The concept lattice can be represented by the line diagram (Hasse diagram) in which every node of the concept lattice is corresponded to the concept from context.

Fo examle simple let us consider a siple context with month simple context with months and their relation to seasons. Using free software ConceptExplorer (special thanks to our colegue Serhiy A. Yevtushenko) [11] we built next Hasse diagrams (fig. 1-2)



(a) **(b)**

Fig. 1. Binary matrix (a) and lattice (b) of Context “Months-seasons”

First step is to fill binary matrix of relation I which means that month relates to season (Fig.1-a). Then we can generate a lattice (Fig.1-b). Conventional top means universal set, bottom empty set, every node means the subset of substances which participate in correct multiple-choice-multiple-answer question while the rest concepts are distractors.

Let us make the context more complex adding, for example, the weather conditions typical for each month (Fig. 2).

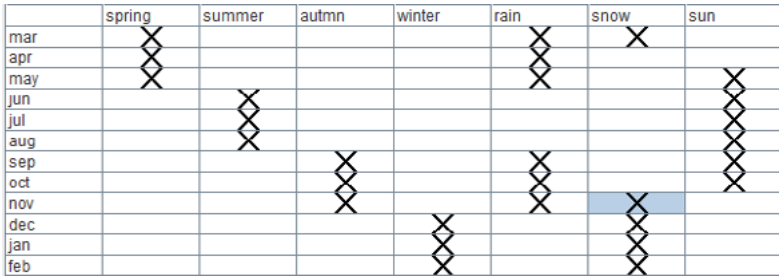


Fig. 2. Binary matrix of complicated context

The lattice is shown in Fig. 3.

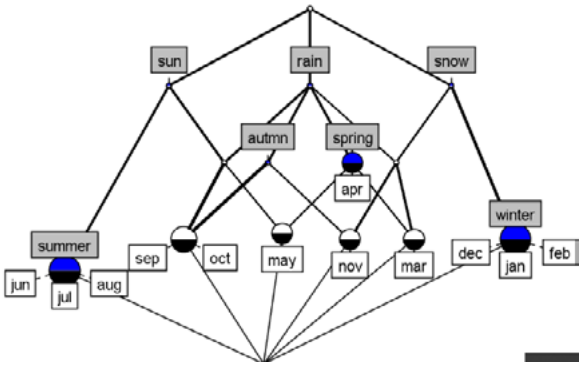


Fig. 3. Lattice of complicated context

How do we read this lattice? Going up from bootom to top by the path of edges we ad every time a substance marked in a bar. When a concept has an addge with other concept it means that consepts participate in one answer. Let us consider the ideal connected with node marred (sep-oct) we found that September and October are autmn months wich can be sunny and rainy (Fig. 4).

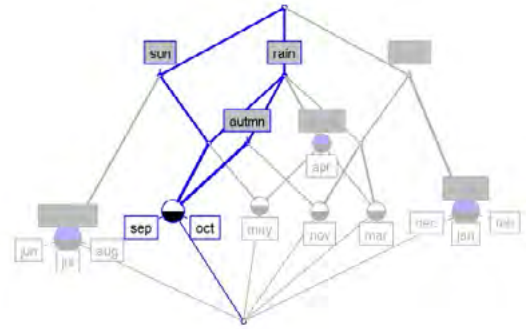


Fig. 4. Ideal generated by concept sep-oct

At same time nolabelled node Fig. 4-b gives us November and March which are connected whith rain and snow.

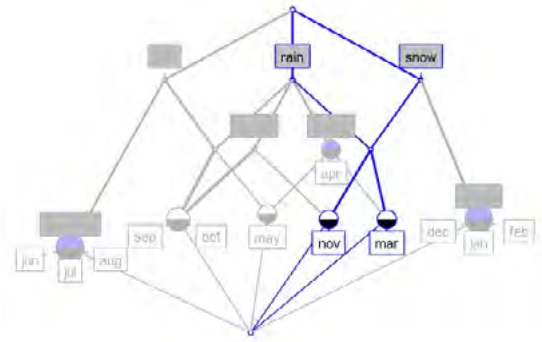


Fig. 5. Ideal generated by nonlabelled concept

Let $J(a)$ – ideal, generated by the concept a, $a \in L$, where L – initial lattice; $D(a)$ – dual ideal, generated by the concept a. It contains all upper elements having path to a. For selection of destructors it is necessary to take sequentially concepts $x \in D(a)/\{a\}$ under criteria of minimal distance from a, and generate new ideals $J'(x)$. The destructors will be objects from all concepts y, where $y \in J'(x)/J(a)$. For attributes testing we generate new dual ideals $D'(x)$ and choose destructors from the set $D(x)/D'(x)$.

Conclusion. As a result of the review of mathematical models and software for creating test questions, it was concluded that it is necessary to continue research to develop technologies for automatic generation of test questions. The existing ontological model of the test question was improved by introducing restrictions on the set of entities that are part of the test question. The types of relations between the entities of the test question were explored. Correspondence between such relations was established for subtypes of generated questions. Proposed clasification for maching in question to *strict, strong, soft* depending of relation type between question and proposed answer. Automatic generation technology of "multiple choice with multiple answer" questions based on Formal Concept Analysis was suggested.

References:

1. Syrota S., Kopychko S., Liskin V. (2020) Ontology-Based Approach for E-learning Course Creation Using Chunks. In: Hu Z., Petoukhov S., Dychka I., He M. (eds) Advances in Computer Science for Engineering and Education II. ICCSEEA 2019. Advances in Intelligent Systems and Computing, vol. 938. Springer, Cham https://doi.org/10.1007/978-3-030-16621-2_60

2. Test constructor. Available at: https://onlinetestpad.com/ru/testmaker

3. S.V. Syrota, V.O. Liskin. Development of the test generator for "Moodle" based on ontology. *Eastern-European Journal of Enterprise Technologies*. 2015, 5/2 (77). P. 44–48. doi: 10.15587/ 1729-4061.2015.51334

4. Prior A., Prior M. Erotetic logic. Philosophical review, 1955, 64, 43–59

5. Logic of questions and answers. Available at: pplied-logic/Logic-of-questions-and-answers
6. Nuel D. BelnapJr., and Thomas B. Steel Jr. The logic of questions and answers. Yale University Press, New Haven and London 1976, vii + 209 pp.
7. Liskin V. E-learning Information Technology Based on an Ontology Driven Learning Engine / V. Liskin, S. Syrota // *International Journal of Computer Science and Information Security*. 2017. Vol. 15(8). P. 258–263.
8. B. Ganter Composition and Decomposition in Formal Concept Analysi H.H. Bock (Ed.), Classification and Related Methods of Data Analysis, North-Holland, Amsterdam (1988), pp. 561-566.
9. B. Ganter and R. Wille, Formal Concept Analysis: Mathematical Foundations, Springer, 1999.
10. Syrota, S., Liskin, V., & Kopychko, S. (2020). Using Galois Connection to Hierarchical Classifying Chunks in Ontology-Based E-learning System. 2020 IEEE 2nd International Conference on System Analysis & Intelligent Computing (SAIC), 1-5. DOI: 10.1109/saic51296.2020.9239131
11. Concept explorer. Avaible at https://sourceforge.net/projects/conexp/files/ /conexp/

УДК 517.977.5
DOI https://doi.org/10.32689/maup.it.2022.3.3

Андрій НЕСТЕРУК
аспірант кафедри інформаційних систем та технологій, асистент кафедри інформаційних систем та технологій, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», проспект Перемоги, 37, Київ, Україна, індекс 03056 (aonesterukr@gmail.com)

ORCID: 0000-0002-1563-7245

Богдан КОРНІЄНКО
доктор технічних наук, професор кафедри інформаційних систем та технологій, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», проспект Перемоги, 37, Київ, Україна, індекс 03056 (bogdanko@gmx.net)

ORCID: 0000-0002-2521-0878

Andrii NESTERUK
graduate student of the Department of Information Systems and Technologies, assistant of the Department of Information Systems and Technologies, National Technical University of Ukraine "Ihor Sikorskyi Kyiv Polytechnic Institute", 37 Peremogy Avenue, Kyiv, Ukraine, postal code 03056 (aonesterukr@gmail.com)

Bogdan KORNIYENKO
Doctor of Technical Sciences, Professor of the Department of Information Systems and Technologies, National Technical University of Ukraine "Ihor Sikorsky Kyiv Polytechnic Institute", 37 Peremogy Avenue, Kyiv, Ukraine, postal code 03056 (bogdanko@gmx.net)

Бібліографічний опис статті: Нестерук, А., Корнієнко, Б. (2022). Математичне моделювання процесу гранулювання у псевдозрідженому шарі. *Інформаційні технології та суспільство*, 3 (5), 20–28. DOI:

Bibliographic description of the article: Nesteruk, A., Kornienko, B. (2022). Matematychne modeliuvannia protsesu hranuliuvannia u psevdozridzhenomu shari [Mathematical modeling of the granulation process in a fluidized bed]. *Informatsiini tekhnolohii ta suspilstvo – Information technology and society*, 3 (5), 20–28. DOI:

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ПРОЦЕСУ ГРАНУЛЮВАННЯ У ПСЕВДОЗРІДЖЕНОМУ ШАРІ

Одним із найпоширеніших методів виготовлення мінеральних добрив є гранулювання. Грануляція в псевдозрідженому шарі є складним процесом взаємодії, на який впливає великий ряд факторів: робочі змінні, властивості матеріалу та параметри обладнання. Будь-яка незначна зміна цих факторів може призвести до значного погіршення кінцевої якості гранул і призвести до втрати енергії та матеріалів, тому створення стратегії контролю для поліпшення якості кінцевих гранул є необхідним. Добрива у вигляді гранул мають ряд переваг перед звичайними добривами у вигляді порошку чи рідини, а саме, легкість у транспортуванні, добре засвоюються та менше піддаються вивітрюванню із ґрунту, зручні у застосування. Щоб отримати тверді частинки з рідкого вихідного матеріалу такого як: розчини, емульсії чи суспензії, застосовують такі процеси як: кристалізація, гранулювання, сушка розпиленням. Залежно від фокусу дослідження, процес гранулювання у псевдозрідженому шарі можна моделювати на різних рівнях абстракції. У мікроскопічному масштабі моделюється динаміка окремих частинок. Розглядається взаємодія частинки з рідиною, обладнанням чи іншими частинками. Наступним більш грубим рівнем абстракції є мезомасштаб. Тут частинки поділяються на класи відповідно до їх характеристик. Передбачається, що частинки класу мають однакові властивості та динаміку. У макроскопічному масштабі, найгрубішому рівні наближення, увага зосереджена на інтегральній поведінці всієї сукупності частинок. Як наслідок, вибрані значення характеристики описують стан шару частинок. Існують різні підходи до моделювання для кожної шкали. Пропонується мікроскопічний масштаб описувати за допомогою моделі гідродинаміки, мезомасштаб – за допомогою моделі балансу, а макроскопічний – методом моментів або за допомогою моделі Лагранжа-Ейлера. Розглянуто також поєднана модель балансу-гідродинаміки та багатокамерна модель балансу, що можуть використовуватись для задач побудови інформаційної технології управління процесом гранулювання у псевдозрідженому шарі.

Ключові слова: *псевдозріджений шар, математична модель, гранулювання, мінеральні добрива.*

MATHEMATICAL MODELLING OF GRANULATION PROCESS IN FLUIDISED BED

One of the most common methods of making mineral fertilizers is granulation. Fluidized bed granulation is a complex interaction process influenced by a large number of factors: operating variables, material properties, and equipment parameters. Any slight change in these factors can lead to a significant deterioration of the final pellet quality and lead to energy and material waste, so creating a control strategy to improve the final pellet quality is essential. Fertilizers in the form of granules have a number of advantages over conventional fertilizers in the form of powder or liquid, namely, ease of transportation, well absorbed and less susceptible to weathering from the soil, convenient to use. To obtain solid particles from liquid starting material such as solutions, emulsions or suspensions, the following processes are used: crystallization, granulation, spray drying. Depending on the focus of the study, the fluidized bed granulation process can be modeled at different levels of abstraction. The dynamics of individual particles is modeled on a microscopic scale. The interaction of a particle with a liquid, equipment or other particles is considered. The next rougher level of abstraction is the mesoscale. Here the particles are divided into classes according to their characteristics. It is assumed that the particles of the class have the same properties and dynamics. On a macroscopic scale, the roughest level of approximation, attention is focused on the integral behavior of the whole set of particles. As a result, the selected characteristic values describe the state of the particle layer. There are different approaches to modeling for each scale. It is proposed to describe the microscopic scale using the hydrodynamics model, the mesoscale using the balance model, and the macroscopic scale using the moments method or the Lagrange-Euler model. A combined balance-hydrodynamics model and a multi-chamber balance model that can be used for the tasks of building information technology for fluidized bed granulation process control technology are also considered.

Key words: *fluidized bed, mathematical model, granulation, mineral fertilizers.*

Постановка проблеми. Розповсюджені в різних галузях сучасних переробних галузей, процеси гранулювання мають велике значення. Більшість вироблених товарів є частиною повсякденного життя: цукор, борошно, розчинна кава та какао, пральний порошок можна знайти майже в кожному будинку, інгалятори сухого порошку значно підвищують якість життя астматиків. Інші тверді продукти, у свою чергу, є проміжним продуктом у складному технологічному ланцюгу. Наприклад, сухе молоко необхідне в харчовій промисловості, гранули добрив зазвичай використовуються в сільському та лісовому господарстві, таблетки для медичних препаратів складаються з гранул, а порошки каталізаторів мають вирішальне значення в хімічній промисловості. Цей короткий, але неповний огляд ілюструє значення твердих частинок у багатьох галузях, таких як харчова, фармацевтична, хімічна та енергетична промисловість. Відповідні процеси приготування частинок настільки ж різноманітні, як і потенційні застосування. Залежно від сировини, частинки можуть бути виготовлені, серед іншого, шляхом розпилювального сушіння рідини, подрібнення твердої речовини або шляхом процесів кристалізації. Ще одним класом широко застосовуваних процесів формування рецептури є гранулювання. Загалом, процес гранулювання перетворює дрібні порошки або рідину, наприклад, розчин, суспензію або розплав, у гранули із заздалегідь визначеними властивостями. Щодо процесу та вимог до продукту, існують різні методи гранулювання. Наприклад – опалювальна грануляція часто використовується для переробки руд і будівельних матеріалів, таких як цемент, грануляція з псевдозрідженим шаром є поширеною для виробництва добрив, неорганічних солей, миючих засобів та фармацевтичних препаратів, компресійна агломерація, наприклад, таблетування та двошнекова агломерація, має безліч застосувань у фармацевтичній, харчовій, хімічній та мінеральній промисловості.

У порівнянні з вихідним матеріалом, вироблені гранули мають багато переваг. Наприклад, вони можуть бути більш довговічними або легшими в обробці. Одним із важливих етапів побудови інформаційної технології управління процесом гранулювання у псевдозрідженому шарі – створення математичної моделі процесу, яка повинна коректно відображати всі етапи процесу гранулювання.

Аналіз попередніх досліджень. Розглянуто поточне розуміння трьох ключових областей процесів вологої грануляції: змочування і зародження, консолідація і зростання, а також розрив і стирання, що надане Івесоном [1]. Боуффард [2] здійснив огляд змінних, пов'язаних з процесом, і фізико-хімічними властивостями у грануляції з псевдозрідженим шаром та описав їх вплив на механізм грануляції. Рейнольдс [3] розглянув поведінку розриву при грануляції від масштабу процесу до масштабу однієї гранули, опрацював велику кількість експериментальних результатів. В основному, механізми грануляції часто розрізняють як зародження, зростання і консолідація, а також розрив і стирання, які можуть відбуватися одночасно в процесі грануляції.

Зародження означає утворення початкових ядер шляхом злипання первинних частинок разом після додавання рідкого розчину в контакті із сухим порошком. На цей процес в основному впливає здатність розчину поширюватися по твердій поверхні. Завдяки експериментальним дослідженням Шефера та Матісена запропоновано два різних механізми зародження: механізм занурення та механізм розподілу залежно від співвідношення розміру частинок до розміру крапель [4-5].

Зростання гранул відбувається, коли частинки контактують і злипаються в процесі грануляції. Можна виділити два різні механізми росту: нашарування та злиття, відповідно до розміру частинок,

що стикаються. Нашаруванням називають єднання між однією великою гранулою та багатьма дрібними частинками або налипання дрібних частинок на поверхню великих вже існуючих гранул. Злиття відбувається, коли дві вологі гранули майже однакового розміру стикаються і обмежуються рідким утвореним між двома частинками, яке стає твердим протягом наступного періоду сушіння.

Консолідація гранул відбувається, коли відбувається зіткнення між гранулами, між гранулами та стінкою в грануляторі. Для аналізу процесу консолідації часто використовуються три сили: капілярна сила, сила в'язкості та сила тертя. Стирання і розриви означають руйнування вологих гранул в процесі гранулювання і стирання в процесі сушіння. Розрив і стирання гранул може вплинути на кінцевий розподіл гранул за розміром. З точки зору якості продукції, вони допомагають покращити однорідність і сипучість гранул.

Вологе дроблення гранул – це процес, за допомогою якого гранули, що містять рідке сполучне і первинні частинки розбиваються на більш дрібні гранули. Гранула розбивається, якщо зовнішнє напруження під час удару перевищує величину власної міцності гранули.

Розглянуто змінні, які впливають на процес грануляції. Відносна вологість повітря на вході, впливає на ріст агломератів і розмір частинок кінцевого продукту. Встановлено, що більші гранули утворюються у міру відносної вологості вхідного повітря, яке збільшується під час гранулювання [6-7], оскільки випаровуваність води на вході повітря зменшується, а утворення рідких і твердих з'єднань посилюється.

Швидкість вхідного нагрітого повітря є важливим робочим параметром, який впливає на гідродинаміку псевдозрідження та росту гранул. Швидкість вхідного повітря при розрідженні впливає на поведінку грануляції та кінцевий розподіл гранул за розміром [7].

Температура повітря на вході є ключовим параметром, що впливає на температуру псевдозрідженого шару та вологість випаровування. Вища температура повітря на вході може висушити вологі гранули швидко і зменшує час гранулювання. Діаметр гранул зменшується при підвищенні температури псевдозріджуючого повітря [6].

Метою статті є системний аналіз гідродинаміки та механізмів, що переважають в процесі гранулювання, для коректного відображення всіх процесів, які відбуваються під час гранулювання, у новій математичній моделі.

Виклад основного матеріалу. Модель балансу використовується для опису зміни розподілу розміру частинок процесу гранулювання у псевдозрідженому шарі [8-9].

Загальне рівняння балансу на основі довжини в пакетному режимі, яке описує швидкість зміни функції щільності числа частинок *n(t,L)* задається таким чином [10]:

$$\frac{\partial n(t,L)}{\partial t} = B(t,L), \tag{1}$$

$$-\frac{\partial}{\partial L}(G(t,L)n(t,L)), \tag{2}$$

$$+\frac{L^2}{2}\int_0^L \frac{\beta(t,(L^3-\lambda^3)^{\frac{1}{3}},\lambda)}{(L^3-\lambda^3)^{\frac{2}{3}}} n(t,(L^3-\lambda^3)^{\frac{1}{3}})n(t,\lambda)d\lambda - n(t,L)\int_0^\infty \beta(t,L,\lambda)n(t,\lambda)d\lambda, \tag{3}$$

$$+\int_L^\infty S(t,\lambda)b(t,L\mid\lambda)n(t,\lambda)d\lambda - S(t,L)n(t,L), \tag{4}$$

де (1) – зародження, (2) – зростання, (3) – агрегація, (4) – розламування, $n(t,L)$ – функція щільності в термінах діаметра частинок L, $\beta(t, L, \lambda)$ – ядро агрегації на основі довжини, яке описує частоту, яку частинки діаметром L та λ стикають, утворюючи частинку об'ємом $L^3 + \lambda^3$, $b(t, L/\lambda)$, є функція розподілу фрагментів, що описує частоту утворення частинок діаметром L від розпаду частинки діаметром λ, $S(t, L)$, засноване на довжині розриву швидкості вибору, яка є частотою руйнування частинки діаметром L, $B(t, L)$ є швидкістю зародження і $G(t, L)$ є швидкістю росту.

При гранулюванні у псевдозрідженому шарі зростання частинок в основному викликано механізмом нашаруванням. Однак під час гранулювання у псевдозрідженому шарі відбувається агрегація частинок, яка розглядається як домінуючий процес, і зростання рідко вивчається дослідниками та не входить до моделювання балансу. Для моделювання балансу використовується лише одне ядро зростання гранулювання у псевдозрідженому шарі [9]:

$$G=\frac{2(1-b)\Phi}{\pi\int_0^\infty nL^2 dL}, \tag{5}$$

де *b* – фракція розпилення, яка сприяє зародженню, *Ф* – розпилена розчинена речовина, *L* – діаметр частинок.

У дискретному методі рівняння моделі балансу розв'язується на кожному окремому інтервалі розмірів. Тому перевага цього методу полягає в тому, що як результат отримаємо розподіл частинок за розміром. З іншого боку, щоб досягти відмінної точності, потрібна велика кількість інтервалів, що означає недолік у вигляді величезних витрат на обчислення, особливо для випадків з широким розподілом частинок за розміром.

Метод моменту

Метод моменту є варіантом розв'зання моделі балансу в умовах моменту, який визначається як:

$$m_k(t) = \int_0^{+\infty} n(L;t)L^k dL \ , \tag{6}$$

де *k* – порядок моменту, *n* – чисельна щільність довжини, *L* – розмір частинок.

Тут популяція описується не властивостями окремих частинок, наприклад, розподілом щільності чисельності *n*, а інтегральними величинами розподілу, так званими моментами.

Моменти тісно пов'язані з важливими інтегральними величинами популяції частинок, наприклад, середньою або загальною поверхнею та об'ємом частинок. У порівнянні з моделлю балансу для чисельного розв'язання потрібно менше обчислювальних потужностей.

Модель гідродинаміки

З поширенням високошвидкісних комп'ютерів, запропоновано модель гідродинаміки – нова модель, що має великий потенціал у наданні детальної інформації про складну динаміку газо-твердої рідини [11]. Загалом, існують дві різні категорії моделей гідродинаміки, які використовуються для моделювання динаміка газових частинок у грануляторі із псевдозрідженим шаром: модель Ейлера та модель Лагранжа.

Модель Ейлера розглядає кожну фазу як взаємопроникний континуум, і обсяг кожної фази не може бути зайнятий іншою фазою, яка вводить поняття частки фазового об'єму. Сума простору частки, яку займає кожна з фаз, дорівнює одиниці. Рівняння частки фазового об'єму:

$$\sum_{q=1}^{n\_p} a_q = 1, \tag{7}$$

де *n _ p* позначає загальну кількість фаз і являє собою об'ємну частку кожної фази *q* (*q = g*, газова фаза і *q = s*, тверда фаза).

Ефективна густина фази *q* визначається:

$$\hat{\rho}_q = a_q \rho_q, \tag{8}$$

де p_q – фізична густина фази *q*.

Модель Ейлера дозволяє включати декілька вторинних твердих тіл фази. Збереження маси та імпульсу виконуються відповідно для кожної фази. Таким чином, модель Ейлера вирішує набір з *n* рівнянь безперервності та імпульсу, що змушує цей підхід бути одним із найбільш складних багатофазних моделей.

Рівняння збереження маси для фази *q* є:

$$\frac{\partial(a_q \rho_q)}{\partial t} + \nabla \cdot (a_q \rho_q \vec{v}_q) = \sum_{p=1}^{n\_p} (\dot{m}_{\mathbf{pq}} - \dot{m}_{\mathbf{qp}}) + S_q \ , \tag{9}$$

де *n _ p* – кількість фаз і $\dot{m}_{pq}$ характеризує фазу масообміну ρ у фазу *q*, і $\dot{m}_{qp}$ характеризує масообмін від фази *q* до фази ρ, s_q означає термін джерела маси для фази *q*, який за замовчуванням дорівнює нулю.

Рівняння збереження імпульсу для газової фази можна записати так:

$$\frac{\partial(a_g \rho_g \vec{v}_g)}{\partial t} + \nabla \cdot (a_g \rho_g \vec{v}_g) = -a_g \nabla p + \nabla \cdot \overline{\overline{\tau}}_g + a_g \rho_g \vec{g} +$$

$$+ \sum_{s=1}^{n\_s} (K_{\mathbf{gs}} (\vec{v}_s - \vec{v}_g) + \dot{m}_{\mathbf{gs}} \vec{v}_{\mathbf{gs}} - \vec{m}_{\mathbf{gs}} \vec{v}_{\mathbf{gs}}) + (\vec{F}_g + \vec{F}_{lift,g} + \vec{F}_{\mathbf{vm}}, g), \tag{10}$$

де *n _ s* позначає кількість твердих фаз. Перший доданок у лівій частині рівняння (10) означає нестаціонарне прискорення, а другий доданок представляє конвективне прискорення потоку.

Рівняння перенесення гранульованої температури для *s*-ої твердої фази пропорційне кінетичній енергії випадкового руху частинок. Отримане рівняння транспорту з кінетичної теорії за моделлю Дінга і Гідаспова [12-13] виглядає так:

$$\frac{3}{2}\left[\frac{\partial(a_S\rho_S\Theta_S)}{\partial t}+\nabla\cdot(a_S\rho_S\vec{v}_S\Theta_S)\right]=(-p_S\overline{\overline{I}}+\overline{\overline{\tau}}_S):\nabla\vec{v}_S-\nabla\cdot(k_{\Theta_S}\nabla\Theta_S)-\gamma_{\Theta_S}+\varphi_p, \quad (11)$$

де $(-p_S\overline{\overline{I}}+\overline{\overline{\tau}}_S):\nabla\vec{v}_S$ – генерація енергії твердим тензором напружень, $k_{\Theta_S}\nabla\Theta_S$ являє собою дифузію енергії з коефіцієнтом дифузії для гранульованої енергії k_{Θ_S}.

Модель багатофазного потоку Ейлера-Ейлера розглядає кожну фазу як окремі взаємопроникні та взаємодіючі континууми в спільній обчислювальній області використання модифікованого рівняння Нав'є-Стокса, при цьому об'єм кожної фази не може бути зайнятий за іншою фазою. У цій схемі, як правило, вважаються тверді частинки ідентичними, що мають репрезентативний діаметр і щільність. Сили взаємодії між фазами моделюються як члени в рівняннях, що описують кожну окрему фазу. Перевагою цього підходу є повномасштабне моделювання процесу. В результаті моделювання методу обчислювальної гідродинаміки на основі каркасу Ейлера-Ейлера є прийнятим для виконання дослідження багатофазного потоку газо-твердих тіл гранулятора з псевдозрідженим шаром [14-18].

Модель Лагранжа розв'язує рівняння руху для кожної частинки з урахуванням зіткнення частинок і сили, що діють на частинку з боку газу, що дозволяє впливати різним властивостям частинок на рух досліджуваної рідини [19-22]. Через обчислювальні обмеження за допомогою моделі Лагранжа все ще неможливо відстежити більше мільйона частинок в межах розумного часу моделювання [23-27]. Тому, коли кількість частинок велика, як у випадку з псевдозрідженим шаром, обчислювальні вимоги можуть змусити віддавати перевагу моделям Ейлера-Ейлера.

Нарешті модель Лагранжа-Ейлера описує газові бульбашки у вигляді дискретних частинок, які можуть зіштовхуватися, зливатися, зупинятися, скорочуватися та зростати. Модель Ейлера не підходить для твердої фази, але підходить для опису емульсії фази газу та частинок.

Необхідно зазначити, що основою багатомасштабного підходу до моделювання є використання моделей меншого розміру, що враховують різного виду взаємодії (частинка-частинка, газ-частинка), для розробки законів взаємодії для моделей великого масштабу. Експериментальні дослідження підтверджують ефективність такого опису газових та твердих потоків [28-31].

Поєднана модель балансу-гідродинаміки

Для того, щоб чітко врахувати явища, пов'язані з частинками (наприклад, агломерація), модель балансу повинна бути розв'язана разом з рівнянням безперервності та рівноваги імпульсу, що дає величезний потенціал комбінованого дослідження моделі балансу-обчислювальної гідродинаміки. Об'ємна частка твердого тіла, швидкість частинок і температура, розрахована з рівнянь переносу Нав'є-Стокса за допомогою моделі гідродинаміки допомагає розв'язати модель балансу, оскільки вони пов'язані з ростом частинок, агрегацією та розривом [32-38]. Після того, як рівняння балансу розв'язані, результати можна використовувати для розрахунку середнього діаметра для подальшої модифікації взаємодії газо-твердих тіл в моделі багатофазного потоку і оновлення інформації про частку, швидкість частинок і температуру для моделі балансу.

Багатокамерна модель балансу

Виходячи з аналізу, моделювання обчислювальної гідродинаміки дає величезні переваги в прогнозуванні гідродинаміки багатофазного потоку гранулювання з псевдозрідженим шаром. При цьому основні проблеми – висока вартість обчислень і відсутність безперервного виходу моделі, що є важливими для управління системою на основі моделі. Таким чином багатокамерна модель балансу може скоротити час обчислень і використовуватись для управління, враховуючи локальну гідродинаміку різних областей гранулятора з псевдозрідженим шаром. У багатокамерній моделі балансу гранулятор з псевдозрідженим шаром поділяється на різні камери згідно з дослідженням газо-твердої рідини. Кожен регіон вважається як однорідний, на основі чого реалізована одновимірна модель балансу, щоб передбачити локальне зростання частинок і еволюцію розподілу частинок за розміром, враховуючи частинки, що спілкуються між сусідніми регіонами. Кількість регіонів визначається експериментальним вимірюванням, моделюванням гідродинаміки або емпіричним досвідом [39-40].

Висновки. Розглянуто механізм гранулювання та властивості розчину та матеріалу, що впливають на процес гранулювання. Гранулювання у псевдозрідженому шарі є складним процесом взаємодії, на який впливає великий ряд факторів: робочі змінні, властивості матеріалу та параметри обладнання.

Залежно від фокусу дослідження, процес гранулювання у псевдозрідженому шарі можна моделювати на різних рівнях абстракції. У мікроскопічному масштабі моделюється динаміка окремих частинок. Розглядається взаємодія частинки з рідиною, обладнанням чи іншими частинками. Наступним більш грубим рівнем абстракції є мезомасштаб. Тут частинки поділяються на класи відповідно до їх характеристик. Передбачається, що частинки класу мають однакові властивості та динаміку. У макроскопічно-

му масштабі, найгрубішому рівні наближення, увага зосереджена на інтегральній поведінці всієї сукупності частинок. Як наслідок, вибрані значення характеристики описують стан шару частинок. Існують різні підходи до моделювання для кожного масштабу. Пропонується мікроскопічний масштаб описувати за допомогою моделі гідродинаміки, мезомасштаб – за допомогою моделі балансу, а макроскопічний – методом моментів або за допомогою моделі Лагранжа-Ейлера.

Розглянуто також поєднана модель балансу-гідродинаміки та багатокамерна модель балансу, що можуть використовуватись для задач побудови інформаційної технології управління процесом гранулювання у псевдозрідженому шарі.

Список використаних джерел:

1. Iveson S.M. Nucleation, growth and breakage phenomena in agitated wet granulation processes: a review. *Powder Technology.* 2001. № 117(1–2). P. 3-39.
2. Bouffard J., Kaster M., Dumont H. Influence of Process Variable and Physicochemical Properties on the Granulation Mechanism of Mannitol in a Fluid Bed Top Spray Granulator. *Drug Development and Industrial Pharmacy.* 2005. № 31(9). P. 923-933.
3. Reynolds G.K. Breakage in granulation: A review. *Chemical Engineering Science.* 2005. № 60(14). P. 3969-3992.
4. Schæfer T., Mathiesen C. Melt pelletization in a high shear mixer. IX. Effects of binder particle size. *International Journal of Pharmaceutics*. 1996. № 139(1-2). P. 139-148.
5. Biggs C.A. Fluidised bed granulation: modelling the growth and breakage kinetics using population balances. *Proceedings of World Congress on Particle Technology*. Sydney. 2002. P. 629-636.
6. Rambali B., Baert L., Massart D. Using experimental design to optimize the process parameters in fluidized bed granulation on a semi-full scale. *International Journal of Pharmaceutics*. 2001. № 220(1–2). P. 149-160.
7. Hemati M. Fluidized bed coating and granulation: influence of processrelated variables and physicochemical properties on the growth kinetics. *Powder Technology.* 2003. № 130(1–3). P. 18-34.
8. Adetayo A.A. Population balance modelling of drum granulation of materials with wide size distribution. *Powder Technology.* 1995. № 82(1). P. 37-49.
9. Vreman A.W., van Lare C.E., Hounslow M.J. A basic population balance model for fluid bed spray granulation. *Chemical Engineering Science.* 2009. № 64(21). P. 4389-4398.
10. Hounslow M.J., Pearson J.M.K., Instone T. Tracer studies of high-shear granulation: II. Population balance modeling. *AIChE Journal*. 2001. № 47(9). P. 1984-1999.
11. van Peborgh Gooch, J.R., Hounslow M.J. Monte Carlo simulation of sizeenlargement mechanisms in crystallization. *AIChE Journal.* 1996. № 42(7). P. 1864-1874.
12. Gidaspow D. Multiphase flow and fluidization: Continuum and kinetic theory descriptions. Academic Press. San Diego. 1994.
13. Ding J., Gidaspow D. A bubbling fluidization model using kinetic theory of granular flow. *AIChE Journal.* 1990. № 36(4). P. 523-538.
14. Syamlal M., Rogers W., Brien O. MFIX Documentation. National Technical Information Service, Springfield. № 1.
15. Gidaspow D., R.B., J.D., Hydrodynamics of Circulating Fluidized Beds. Kinetic Theory Approach. In Fluidization VII, Proceeding of the 7th Engineering Foundation Conference on Fluidization. 1992. P. 75-82.
16. Lun C.K.K., Kinetic Theories for Granular Flow: Inelastic Particles in Couette Flow and Slightly Inelastic Particles in a General Flow Field. *Journal of Fluid Mechanics.* 1984. № 140. P. 223-256.
17. Korniyenko B.Y. The two phase model of formation of mineral fertilizers in the fluidized–bed granulator. *The Advanced Science Journal*. 2013. № 4. P. 41-44.
18. Корнієнко Б.Я. Двохфазна модель процесу зневоднення та гранулювання у псевдозрідженому шарі. *Вісник Національного технічного університету України «Київський політехнічний інститут». Серія «Хімічна інженерія, екологія та ресурсозбереження».* 2012. № 2(10). С. 31-35.
19. Ладієва Л.Р., Борзєнкова С.В. Трьохфазна математична модель процесу зневоднення та гранулювання у псевдозрідженому шарі. *Наукоємні технології.* 2019. 2(42). С. 239-245.
20. Корнієнко Б.Я. Інформаційні технології оптимального управління виробництвом мінеральних добрив. Київ. 2014. 288 с.
21. Ладієва Л.Р.б Колесник М.В. Побудова оптимальної системи керування процесом гранулювання в псевдозрідженому шарі. *Інтелектуальні системи прийняття рішень та проблеми обчислювального інтелекту.* Херсон. 2015. С. 78-80.
22. Ладієва Л.Р.б Мироненко О.М. Стохастичне керування процесом гранулювання мінеральних добрив у псевдозрідженому шарі. *Інтелектуальні системи прийняття рішень та проблеми обчислювального інтелекту.* Херсон. 2014. С. 121-123.
23. Корнієнко Б.Я. Двохфазна модель процесу зневоднення та гранулювання у псевдозрідженому шарі. *Вісник Національного технічного університету України «Київський політехнічний інститут». Серія «Хімічна інженерія, екологія та ресурсозбереження».* 2012. № 2(10). С. 31-35.
24. Корнієнко Б.Я. Математичне моделювання динаміки процесів переносу при зневодненні та гранулюванні у псевдозрідженому шарі. *Науковий журнал «Вісник Національного авіаційного університету».* 2012. № 4(53). С. 84-90.
25. Korniyenko B.Y. Modeling of transport processes in disperse systems. *The Advanced Science Journal.* 2013. № 1. P. 7-10.

26. Корнієнко Б.Я. Мінеральні добрива. Двохфазна модель утворення в грануляторі із псевдозрідженим шаром. *Хімічна промисловість України.* 2013. № 1. С. 39-43.

27. Корнієнко Б.Я., Ладієва Л.Р., Снігур О.В. Гранулювання у псевдозрідженому шарі. Дослідження детермінованого хаосу процесу. *Хімічна промисловість України.* 2013. № 2. С. 20-23.

28. Korniyenko B.Y. Research modes of a fluidized bed granulator. *The Advanced Science Journal.* 2013. № 5. P. 12-15.

29. Корнієнко Б.Я. Ідентифікація процесу гранулювання мінеральних добрив у апараті з псевдозрідженим шаром. *Наукоємні технології.* 2013. № 3(19). С. 280-284.

30. Korniyenko B.Y., Osipa L. Identification of the granulation process in the fluidized bed. *ARPN Journal of Engineering and Applied Sciences.* 2018. № 13(14). P. 4365-4370.

31. Korniyenko B., Ladieva L. Mathematical Modeling Dynamics of the Process Dehydration and Granulation in the Fluidized Bed. *Advances in Intelligent Systems and Computing.* 1247 AISC. 2021. P. 18-30.

32. Korniyenko B., Ladieva L., Galata L. Control system for the production of mineral fertilizers in a granulator with a fluidized bed. 2020 2nd IEEE *International Conference on Advanced Trends in Information Theory.* 2020. № 9349344. P. 307-310.

33. Korniyenko B.Y., Borzenkova S.V., Ladieva L.R. Research of three-phase mathematical model of dehydration and granulation process in the fluidized bed. *ARPN Journal of Engineering and Applied Sciences*. 2019. № 14(12). P. 2329-2332.

34. Korniyenko B.Y., Ladieva L.R. Mathematical modeling dynamics of the process dehydration and granulation in the fluidized bed. *Інтелектуальні системи прийняття рішень і проблеми обчислювального інтелекту*. Херсон. 2019. P. 86-88.

35. Корнієнко Б.Я. Мінеральні добрива. Оптимізація процесу зневоднення та гранулювання у псевдозрідженому шарі. *Хімічна промисловість України.* 2013. № 4. С. 69-73.

36. Korniyenko B.Y. Static and dynamic characteristics of transport processes in disperse systems. *Наукоємні технології.* 2013. № 2(18). С. 166-170.

37. Корнієнко Б.Я. Мінеральні добрива. Статична оптимізація процесу гранулювання у псевдозрідженому шарі. *Хімічна промисловість України*. 2013. № 5. С. 36-40.

38. Корнієнко Б.Я. Ідентифікація процесу гранулювання мінеральних добрив у апараті з псевдозрідженим шаром. *Наукоємні технології.* 2013. № 3(19). С. 280-284.

39. Корнієнко Б.Я. Задачі оптимізації зневоднення та гранулювання мінеральних добрив у псевдозрідженому шарі. *Вісник Національного технічного університету України «Київський політехнічний інститут». Серія «Хімічна інженерія, екологія та ресурсозбереження».* 2014. № 1(12). С. 28-31.

40. Kornienko Y.M., Liubeka A.M., Sachok R.V., Korniyenko B.Y. Modeling of heat exchangement in fluidized bed with mechanical liquid distribution. *ARPN Journal of Engineering and Applied Sciences*. 2019. № 14(12). P. 2203-2210.

References:

1. Iveson, S.M., et al. (2001). Nucleation, growth and breakage phenomena in agitated wet granulation processes: a review. *Powder Technology*, 117(1–2), 3-39.

2. Bouffard, J., M. Kaster, and H. Dumont (2005). Influence of Process Variable and Physicochemical Properties on the Granulation Mechanism of Mannitol in a Fluid Bed Top Spray Granulator. *Drug Development and Industrial Pharmacy*, 31(9), 923-933.

3. Reynolds, G.K., et al. (2005). Breakage in granulation: A review. *Chemical Engineering Science*, 60(14), 3969-3992.

4. Schæfer, T. and C. Mathiesen (1996). Melt pelletization in a high shear mixer. IX. Effects of binder particle size. *International Journal of Pharmaceutics*, 139(1-2), 139-148.

5. Biggs, C.A., et al. (2002). Fluidised bed granulation: modelling the growth and breakage kinetics using population balances. *Proceedings of World Congress on Particle Technology*, 629-636.

6. Rambali, B., L. Baert, and D.L. Massart (2001). Using experimental design to optimize the process parameters in fluidized bed granulation on a semi-full scale. *International Journal of Pharmaceutics*, 220(1–2), 149-160.

7. Hemati, M., et al. (2003). Fluidized bed coating and granulation: influence of processrelated variables and physicochemical properties on the growth kinetics. *Powder Technology*, 130(1–3), 18-34.

8. Adetayo, A.A., et al. (1995). Population balance modelling of drum granulation of materials with wide size distribution. *Powder Technology*, 82(1), 37-49.

9. Vreman, A.W., C.E. van Lare, and M.J. Hounslow (2009). A basic population balance model for fluid bed spray granulation. *Chemical Engineering Science*, 64(21), 4389-4398.

10. Hounslow, M.J., J.M.K. Pearson, and T. Instone (2001). Tracer studies of high-shear granulation: II. Population balance modeling. *AIChE Journal*, 47(9), 1984-1999.

11. van Peborgh Gooch, J.R. and M.J. Hounslow (1996). Monte Carlo simulation of sizeenlargement mechanisms in crystallization. *AIChE Journal*, 42(7), 1864- 1874.

12. Gidaspow, D. (1994). Multiphase flow and fluidization: Continuum and kinetic theory descriptions. *Academic Press*.

13. Ding, J. and D. Gidaspow (1990). A bubbling fluidization model using kinetic theory of granular flow. *AIChE Journal*, 36(4), 523-538.

14. M. Syamlal, W. Rogers, and O.Brien T. J. MFIX Documentation. *National Technical Information Service*, 1

15. Gidaspow, D., R.B., and J.D. (1992). Hydrodynamics of Circulating Fluidized Beds. *Kinetic Theory Approach. In Fluidization VII, Proceeding of the 7th Engineering Foundation Conference on Fluidization*, 75-82.

16. Lun, C.K.K., et al. (1984). Kinetic Theories for Granular Flow: Inelastic Particles in Couette Flow and Slightly Inelastic Particles in a General Flow Field. *Journal of Fluid Mechanics*, 140, 223-256.

17. Korniyenko B.Y. (2013). The two phase model of formation of mineral fertilizers in the fluidized – bed granulator. *The Advanced Science Journal*, 4, 41-44.

18. Korniyenko B.Y. (2012). Dvokhfazna model protsesu znevodnennia ta hranuliuvannia u psevdozridzhenomu shari [Two-phase model of dehydration and granulation process in fluidized bed]. *Visnyk Natsionalnoho tekhnichnoho universytetu Ukrainy «Kyivskyi politekhnichnyi instytut», Seriia «Khimichna inzheneriia, ekolohiia ta resursozberezhennia» – Bulletin of the National Technical University of Ukraine "Kyiv Polytechnic Institute", Series "Chemical Engineering, Ecology and Resource Conservation"*, 2(10), 31–35.

19. Ladieva L.R. (2019). Trokhfazna matematychna model protsesu znevodnennia ta hranuliuvannia u psevdozridzhenomu shari [Three-phase mathematical model of the process of dehydration and granulation in a fluidized bed]. *Naukoiemni tekhnolohii – Science-intensive technologies*, 2(42), 239-245.

20. Korniyenko B.Y. (2014). Informatsiini tekhnolohii optymalnoho upravlinnia vyrobnytstvom mineralnykh dobryv [Information technologies of optimal management of mineral fertilizers production: monograph]. *Vydavnytstvo «Ahrar Media Hrup» – Agrar Media Group Publishing House*, 288.

21. Ladieva L.R. (2015). Pobudova optymalnoi systemy keruvannia protsesom hranuliuvannia v psevdozridzhenomu shari [Construction of an optimal control system for the granulation process in a fluidized bed]. *Intelektualni systemy pryiniattia rishen ta problemy obchysliuvalnoho intelektu: zb. Nauka. Materialy mizhnarodnoi konferentsii (ISDMCI 2015) – Intelligent decision-making systems and problems of computational intelligence: coll. Science. Proceedings of the International Conference (ISDMCI 2015)*, 78-80.

22. Ladieva L.R. (2014). Stokhastychne keruvannia protsesom hranuliuvannia mineralnykh dobryv u psevdozridzhenomu shari [Stochastic control of the process of granulation of mineral fertilizers in a fluidized bed]. *Intelektualni systemy pryiniattia rishen ta problemy obchysliuvalnoho intelektu: zb. Nauka. Materialy mizhnarodnoi konferentsii (ISDMCI 2014) – Intelligent decision-making systems and problems of computational intelligence: coll. Science. Proceedings of the International Conference (ISDMCI 2014)*, 121-123.

23. Korniyenko B.Y. (2012). Dvokhfazna model protsesu znevodnennia ta hranuliuvannia u psevdozridzhenomu shari [Two-phase model of dehydration and granulation process in fluidized bed]. *Visnyk Natsionalnoho tekhnichnoho universytetu Ukrainy «Kyivskyi politekhnichnyi instytut», Seriia «Khimichna inzheneriia, ekolohiia ta resursozberezhennia» – Bulletin of the National Technical University of Ukraine "Kyiv Polytechnic Institute", Series "Chemical Engineering, Ecology and Resource Conservation*, 2(10), 31–35.

24. Korniyenko B.Y. (2012). Matematychne modeliuvannia dynamiky protsesiv perenosu pry znevodnenni ta hranuliuvanni u psevdozridzhenomu shari [Mathematical modeling of the dynamics of transfer processes during dehydration and granulation in a fluidized bed]. *Naukovyi zhurnal «Visnyk Natsionalnoho aviatsiinoho universytetu» – Scientific Journal "Bulletin of the National Aviation University"*, 4(53), 84-90.

25. Korniyenko B.Y. (2013). Modeling of transport processes in disperse systems. *The Advanced Science Journal*, 1, 7-10.

26. Korniyenko B.Y. (2013). Mineralni dobryva. Dvokhfazna model utvorennia v hranuliatori iz psevdozridzhenym sharom [Mineral fertilizers. Two-phase model of formation in a granulator with a fluidized bed]. *Khimichna promyslovist Ukrainy – Chemical industry of Ukraine*, 1, 39-43.

27. Korniyenko B.Y. (2013). Hranuliuvannia u psevdozridzhenomu shari. Doslidzhennia determinovanoho khaosu protsesu [Granulation in a fluidized bed. Research of deterministic process chaos]. *Khimichna promyslovist Ukrainy – Chemical industry of Ukraine*, 2, 20-23.

28. Korniyenko B.Y. (2013). Research modes of a fluidized bed granulator. *The Advanced Science Journal*, 5, 12-15.

29. Korniyenko B.Y. (2013). Identyfikatsiia protsesu hranuliuvannia mineralnykh dobryv u aparati z psevdozridzhenym sharom [Identification of the process of granulation of mineral fertilizers in the apparatus with a fluidized bed]. *Naukoiemni tekhnolohii – Science-intensive technologies*, 3(19), 280-284.

30. Korniyenko B.Y. (2018). Identification of the granulation process in the fluidized bed. *ARPN Journal of Engineering and Applied Sciences*, 13(14), 4365-4370.

31. Korniyenko, B., Ladieva, L. (2021). Mathematical Modeling Dynamics of the Process Dehydration and Granulation in the Fluidized Bed. *Advances in Intelligent Systems and Computing*, 18-30.

32. Korniyenko, B., Ladieva, L., Galata, L. (2020). Control system for the production of mineral fertilizers in a granulator with a fluidized bed. *ATIT 2020 – Proceedings: 2020 2nd IEEE International Conference on Advanced Trends in Information Theory*, 9349344, 307-310.

33. Korniyenko, B.Y., Borzenkova, S.V., Ladieva, L.R. (2019). Research of a three-phase mathematical model of dehydration and granulation process in the fluidized bed. *ARPN Journal of Engineering and Applied Sciences*, 14(12), 2329-2332.

34. Korniyenko B.Y., Ladieva L.R. (2019). Mathematical modeling dynamics of the process of dehydration and granulation in the fluidized bed. *Intelligent decision-making systems and problems of computational intelligence: materials intern. Science*, 86-88.

35. Korniyenko B.Y. (2013). Mineralni dobryva. Optymizatsiia protsesu znevodnennia ta hranuliuvannia u psevdozridzhenomu shari [Mineral fertilizers. Optimization of the dehydration and granulation process in the fluidized bed]. *Khimichna promyslovist Ukrainy – Chemical industry of Ukraine*, 4, 69-73.

36. Korniyenko B.Y. (2013). Static and dynamic characteristics of transport processes in disperse systems. *Scientific technologies*, 2(18), 166-170.

37. Korniyenko B.Y. (2013). Mineralni dobryva. Statychna optymizatsiia protsesu hranuliuvannia u psevdozridzhenomu shari [Mineral fertilizers. Static optimization of the granulation process in a fluidized bed]. *Khimichna promyslovist Ukrainy – Chemical industry of Ukraine*, 5, 36-40.

38. Korniyenko B.Y. (2013). Identyfikatsiia protsesu hranuliuvannia mineralnykh dobryv u aparati z psevdozridzhenym sharom [Identification of the process of granulation of mineral fertilizers in the apparatus with a fluidized bed]. *Naukoiemni tekhnolohii – Science-intensive technologies*, 3(19), 280-284.

39. Korniyenko B.Y. (2014). Zadachi optymizatsii znevodnennia ta hranuliuvannia mineralnykh dobryv u psevdozridzhenomu shari [Problems of dehydration optimization and granulation of mineral fertilizers in the fluidized bed]. *Visnyk Natsionalnoho tekhnichnoho universytetu Ukrainy «Kyivskyi politekhnichnyi instytut», Seriia «Khimichna inzheneriia, ekolohiia ta resursozberezhennia» – Bulletin of the National Technical University of Ukraine "Kyiv Polytechnic Institute", Series "Chemical Engineering, Ecology and Resource Conservation"*, 1(12), 28-31.

40. Kornienko, Y. M., Liubeka, A. M., Sachok, R. V., Korniyenko, B. Y. (2019). Modeling of heat exchangement in fluidized bed with mechanical liquid distribution. *ARPN Journal of Engineering and Applied Sciences*, 14(12), 2203-2210.

УДК 51-7+004.4
DOI https://doi.org/10.32689/maup.it.2022.3.4

Євген ОГІНСЬКИЙ
аспірант кафедри інженерії програмного забезпечення, Державний університет «Житомирська політехніка», вулиця Чуднівська, 103, Житомир, Україна (oginsky2@gmail.com)

ORCID: 0000-0002-7777-8449

Дмитро АНТОНЮК
кандидат педагогічних наук, доцент, доцент кафедри інженерії програмного забезпечення, Державний університет «Житомирська політехніка», вулиця Чуднівська, 103, Житомир, Україна (dmitry_antonyuk@yahoo.com)

ORCID: *0000-0001-7496-3553*

Тетяна ВАКАЛЮК
доктор педагогічних наук, професор, професор кафедри інженерії програмного забезпечення, Державний університет «Житомирська політехніка», вулиця Чуднівська, 103, Житомир, Україна (tetianavakaliuk@gmail.com)

ORCID: *0000-0001-6825-4697*

Дмитро МОСКАЛИК
аспірант кафедри інженерії програмного забезпечення, Державний університет «Житомирська політехніка», вулиця Чуднівська, 103, Житомир, Україна (d.moskalyk@sana-commerce.com)

ORCID: *0000-0002-4421-9325*

Вячеслав ВАСИЛЕНКО
аспірант кафедри інженерії програмного забезпечення, Державний університет «Житомирська політехніка», вулиця Чуднівська, 103, Житомир, Україна (slava.vasylenko@gmail.com)

ORCID: 0000-0001-7595-689X

Yevhen OHINSKYI
PhD student, Department of Software Engineering, Zhytomyr Polytechnic State University, Chudnivska str, 103, Zhytomyr, Ukraine (oginsky2@gmail.com)

Dmytro ANTONIUK
Candidate of Pedagogical Sciences, Associate Professor, Associate Professor of the Software Engineering Department, Zhytomyr Polytechnic State University, Chudnivska str, 103, Zhytomyr, Ukraine (dmitry_antonyuk@yahoo.com)

Tetyana VAKALIUK
Doctor of Pedagogical Sciences, Professor, Professor of Software Engineering Department, Zhytomyr Polytechnic State University, Chudnivska str, 103, Zhytomyr, Ukraine (tetianavakaliuk@gmail.com)

Dmytro MOSKALYK
PhD student, Department of Software Engineering, Zhytomyr Polytechnic State University, Chudnivska str, 103, Zhytomyr, Ukraine (d.moskalyk@sana-commerce.com)

Viacheslav VASYLENKO
PhD student, Department of Software Engineering, Zhytomyr Polytechnic State University, Chudnivska str, 103, Zhytomyr, Ukraine (slava.vasylenko@gmail.com)

Бібліографічний опис статті: Огінський, Є., Антонюк, Д., Вакалюк, Т., Москалик, Д., Василенко, В. (2022). Аналіз алгоритмічного і математичного апарату для систем побудови та аналізу інструментів управління персональними фінансами. *Інформаційні технології та суспільство*, 3 (5), 29–40. DOI:

Bibliographic description of the article: Ohinskyi, Ye., Antoniuk D., Vakaliuk T., Moskalyk D., Vasylenko V. (2022). Analiz algorytmichnoho i matematychnoho aparatu dlia system pobudovy ta analizu instrumentiv upravlinnia personalnymy finansamy [Analysis of algorithmic and mathematical apparatus for system of development and analysis of personal finance management tools]. *Informatsiini tekhnolohii ta suspilstvo – Information technology and society,* 3 (5), 29–40. DOI:

АНАЛІЗ АЛГОРИТМІЧНОГО І МАТЕМАТИЧНОГО АПАРАТУ ДЛЯ СИСТЕМ ПОБУДОВИ ТА АНАЛІЗУ ІНСТРУМЕНТІВ УПРАВЛІННЯ ПЕРСОНАЛЬНИМИ ФІНАНСАМИ

Управління персональними фінансами має велике значення для сучасної людини. Воно визначає ухвалення фінансових рішень для того, щоб людина почувалася захищеною, покращувала персональний фінансовий добробут, завчасно формувала резервний фонд на випадок непередбачуваних ситуацій. Для досягнення фінансових цілей можуть використовуватися різні фінансові інструменти. Це може бути банківський депозит, облігації, акції, нерухомість, тощо. Кожен фінансовий інструмент також має свої характеристики. Кожен інструмент має свою ризикованість, часові обмеження і прибутковість. Валютні ризики, фінансові кризи, пандемії дуже сильно впливають на характеристики фінансових інструментів, роблять їх більш або менш привабливими. Вкладення у державні облігації можуть бути дуже вигідними в один період часу і значно менш вигідними у інші періоди часу. Тобто фінансові інструменти можуть підходити для визначених фінансових цілей в один період часу і не підходити в інший. Виникає питання вибору математичного апарату для вирішення проблеми знаходження оптимальних стратегій для фінансових вкладень людей, для того щоб врахувати різні фактори. Розглянуто детально такі методи і їх особливості як Supervised learning, Unsupervised learning, навчання з підкріпленням. Також розглянуто деякі алгоритми Supervised learning: лінійна регресія, дерева ухвалення рішень, поліноміальна регресія. Алгоритми методу Unsupervised learning: кластерний аналіз, метод К-середніх, ієрархічна кластеризація, кластеризація на основі щільності. Також розглянуто підхід агентного моделювання і спосіб статистичного моделювання випадкових процесів за допомогою ланцюгів Маркова, алгоритми навчання з підкріпленням, а саме Value-based метод і Policy-based метод. Також було розглянуто різні типи задач, які можна вирішувати за допомогою наведених підходів. Дані алгоритми дають можливість для їх широкого застосування у різних сферах, а також для вирішення питань управління персональними фінансами.

Ключові слова: *Персональні фінанси, інструменти управління персональними фінансами, навчання з підкріпленням, лінійна регресія, дерева ухвалення рішень, поліноміальна регресія.*

ANALYSIS OF ALGORITHMIC AND MATHEMATICAL APPARATUS FOR SYSTEM OF DEVELOPMENT AND ANALYSIS OF PERSONAL FINANCE MANAGEMENT TOOLS

Personal finance management is important for people nowadays. It determines financial decision making so that a person feels protected, improves personal financial well-being, and creates a reserve fund in advance to handle unforeseen situations. Various financial instruments can be used to achieve financial goals. It can be a bank deposit, bonds, shares, real estate, etc. Each financial instrument also has its own characteristics. Each instrument has certain riskiness, time limits and profitability. Currency risks, financial crises, and pandemics greatly affect the characteristics of financial instruments, make them more or less attractive. Investing in government bonds can be very profitable in one time period and less profitable in other time period. So, financial instruments may be suitable for certain financial goals in one time period and not suitable in another. The question arises of choosing a mathematical apparatus for solving the problem of finding optimal strategies for people's financial investments, in order to consider various factors.

Such methods as Supervised learning, Unsupervised learning, and Reinforcement learning and their peculiarities are considered in detail. It is also considered certain Supervised learning algorithms: linear regression, decision trees, polynomial regression. Algorithms of the Unsupervised learning method: cluster analysis, K-means method, hierarchical clustering, density-based clustering. The approach of agent modeling and the method of statistical modeling of random processes using Markov chains, reinforcement learning algorithms, namely the Value-based method and the Policy-based method, are also considered. Different types of problems that can be solved using the above approaches were also considered. These algorithms can be widely used in various areas, as well as for solving personal finance management tasks.

Key words: *Personal finance, tools personal finance management, reinforcement learning, linear regression, decision trees, polynomial regression.*

Постановка проблеми. Управління персональними фінансами має велике значення для сучасної людини. Воно визначає ухвалення фінансових рішень для того, щоб людина почувалася захищеною, покращувала персональний фінансовий добробут, завчасно формувала резервний фонд на випадок непередбачуваних ситуацій.

Окрім збалансованого підходу до керування поточних доходів і витрат у людини виникає необхідність планувати накопичення з бажаною сумою для виконання мети. Наприклад, пенсійні накопичення, накопичення на освіту дітей, накопичення на купівлю житла, авто, тощо.

До кожної фінансової цілі ставляться свої вимоги і кожна ціль має свою важливість. Наприклад, людина може легко відкласти купівлю авто, або змінити суму купівлі. Але накопичення коштів на навчання дітей мають набагато більшу важливість, і фіксовані часові рамки. Не досягнення цієї цілі може мати вкрай небажаний наслідок.

Також фінансові цілі мають різні вимоги по доступності коштів. Резервний фонд має бути диверсифікованим і легко доступним, він має конкретну суму і бути достатнім для проживання родини протягом певного періоду часу за відсутності інших доходів. Накопичення на освіту мають конкретну дату і до досягнення цієї дати вони можуть бути недоступними.

Для досягнення фінансових цілей можуть використовуватися різні фінансові інструменти. Це може бути банківський депозит, облігації, акції, нерухомість, тощо. Кожен фінансовий інструмент також має свої параметри. Кожен інструмент має свою ризикованість, часові обмеження і прибутковість. Наприклад депозит має дату закриття і фіксовану процентну ставку, має помірну прибутковість і маленьку ризикованість. У депозита є можливість дострокового розірвання з відповідними штрафами і фінансовими втратами. Облігації також мають дату погашення і фіксовану прибутковість. Державні облігації мають найменшу ризикованість. У свою чергу вкладення в акції не мають конкретної дати повернення вкладень, акції можна купити або продати в будь який час. Акції мають набагато більшу ризикованість порівняно з депозитами або облігаціями. Але також акції мають набагато більший потенціал по прибутковості на великому проміжку часу.

Властивості фінансових інструментів робить їх найбільш підходящими або небажаними для використання в досягненні конкретних фінансових цілей. Наприклад депозити або державні облігації, що мають гарантований прибуток і роблять кошти доступними на конкретну дату можуть бути найбільш відповідними інструментами для накопичень для освіти дітей, частково використання в резервному фонді, використання для пенсійних накопичень, у випадку скорого настання пенсії. В свою чергу акції можуть бути досить перспективним інструментом для пенсійних накопичень для молодих людей.

Також люди мають свої характеристики, що може впливати на вибір того чи іншого інструменту. Наприклад, варто враховувати вік, схильність до ризику, можливість переносити тимчасові зниження поточної вартості інвестицій.

Валютні ризики, фінансові кризи, пандемії дуже сильно впливають на параметри фінансових інструментів. Роблять їх більш або менш привабливими. Вкладення у державні облігації можуть бути дуже привабливими в один період часу і значно зменшувати свою привабливість у інші періоди часу. Тобто фінансові інструменти можуть підходити для визначених фінансових цілей в один період часу і не підходити в інший.

Виникає питання вибору математичного апарату для вирішення проблеми знаходження оптимальних стратегій для фінансових вкладень людей, для того щоб врахувати різні фактори.

Аналіз останніх досліджень і публікацій. Існують різні методи машинного навчання, які можуть широко застосовуються у фінансовій сфері і які можуть бути застосовані для систем побудови та аналізу інструментів управління персональними фінансами.

Зокрема, зазначеним питанням приділяли увагу в своїх дослідженнях такі науковці як: R. S. Sutton [1; 2], A. G. Barto [1, 2], К. Ю. Кононова [5], B. Hambly [7], R. Xu [7], H. Yang [7] та ін. Розглянемо детально такі методи і їх особливості як Supervised learning, Unsupervised learning, навчання з підкріпленням. Також розглянемо деякі алгоритми Supervised learning: лінійна регресія, дерева ухвалення рішень, поліноміальна регресія. Алгоритми методу Unsupervised learning: кластерний аналіз, метод К-середніх, ієрархічна кластеризація, кластеризація на основі щільності. Також розглянемо підхід агентного моделювання і спосіб статистичного моделювання випадкових процесів за допомогою ланцюгів Маркова, алгоритми навчання з підкріпленням, а саме Value-based метод і Policy-based метод.

Мета статті – аналіз алгоритмічного і математичного апарату для систем побудови та аналізу інструментів управління персональними фінансами.

Виклад основного матеріалу.

Машинне навчання. Підхід Supervised learning

Навчання з учителем використовує набір даних для тренування для того щоб навчити модель видавати бажані вихідні дані. Набір даних для тренування включає вхідні дані і правильні вихідні дані, що дає змогу моделі вчитись протягом часу (рис. 1).

Навчання з учителем використовується для двох типів проблем: класифікація і регресія.

Класифікація використовує алгоритм щоб вказати вхідним даним відповідну категорію. Він розпізнає специфічні сутності в наборі даних і намагається зробити висновок як ці сутності повинні бути позначені. Основні алгоритми класифікації: лінійна класифікація, метод опорних векторів, дерева ухвалення рішень, метод k-найближчих сусідів.

Регресія використовується для того щоб визначити зв'язок між залежними і незалежними змінними. Вона може використовуватись наприклад для визначення вартості будинку в залежності від параметрів. Популярні алгоритми регресії – лінійна регресія, логістична регресія і поліноміальна регресія.

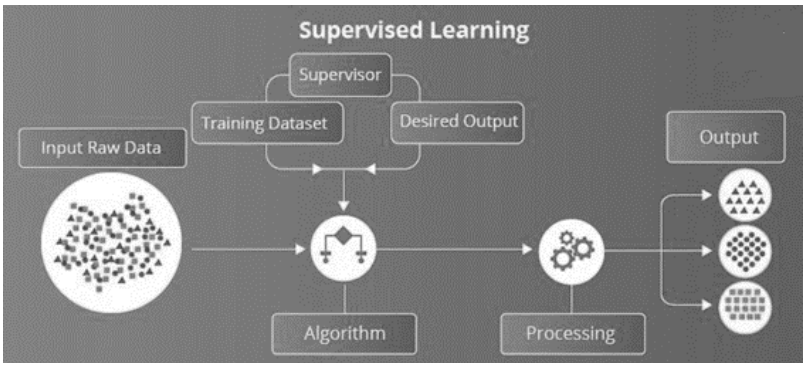


Рис. 1. Принцип роботи машинного навчання Supervised learning

Дерево рішень як алгоритм машинного навчання – це об'єднання логічних правил типу «Якщо ... То ...» в структуру «дерева», створюючи ієрархічну структуру правил. Дерево рішень складається з вузлів, де проводиться перевірка умови і листя (вузлів рішення), що вказують на клас або його середнє значення. Під час побудови дерева рішень обчислюється приріст інформації (на основі оцінки ентропії). Ентропія відповідає ступеню хаосу в системі. Чим вище ентропія, тим менше впорядкована система і навпаки. Інформація протилежна ентропії. Ентропія Шеннона визначається для системи з N можливими станами так:

$$H(Y) = -\sum p_i \log_2 p_i$$

де p_i – ймовірності знаходження системи в *i*-му стані.

Основою алгоритмів побудови дерева рішень є принцип жадібної максимізації приросту інформації – на кожному кроці вибирається та ознака, за якою під час розподілу приріст інформації виявляється найбільшим. Далі процедура повторюється рекурсивно, поки ентропія не буде дорівнювати нулю або якійсь малій величині (якщо дерево не підлаштовується ідеально під навчальну вибірку, щоб уникнути перенавчання). У різних алгоритмах застосовуються різні евристики для «ранньої зупинки» або «відсікання», щоб уникнути побудови перенавченого дерева.

Алгоритм дерева рішень класифікує об'єкти, відповідаючи на «питання» про їх атрибутах, розташовані у вузлових точках. Залежно від відповіді вибирається одна з гілок, і так до тих пір, поки не буде досягнуто «лист» – остаточну відповідь.

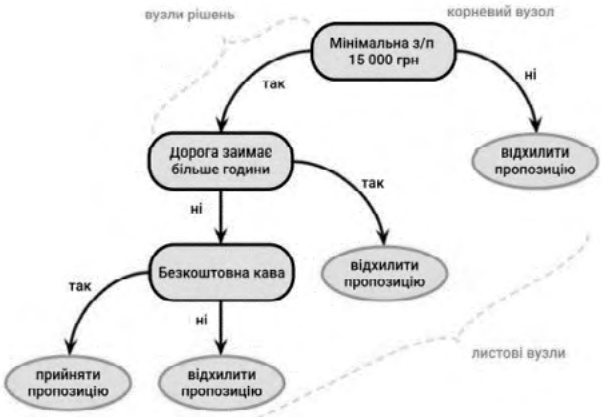


Рис. 2. Алгоритм дерева прийняття рішень щодо пропозиції нової роботи

Серед застосувань дерева рішень – платформи управління знаннями для клієнтського обслуговування, прогнозного призначення цін і планування випуску продукції. У страховій компанії дерево рішень допоможе з'ясувати, які види страхових продуктів і премій краще задіяти з урахуванням можливого ризику. Використовуючи дані про місцезнаходження і відомості про страхові випадки з урахуванням погодних умов, система може визначати категорії ризику на підставі поданих вимог і витрачених сум. Потім, використовуючи моделі, система буде оцінювати нові заяви про страховий захист, класифікуючи їх по категорії ризику і можливого фінансового збитку.

Прикладом використання алгоритму в банківській сфері є кредитування. Якщо визначити ключові фактори – вік, рівень доход, утриманці, сімейний стан, кредити в інших організаціях, наявність рухо-

мого і нерухомого майна, то по кожній з ключових гілок можна скласти приблизний план можливих дій. При послідовному виясненні інформації, наявність рухомого і нерухомого майна може стати додатковою гарантією повернення коштів, тому, якщо потенційний позичальник дійшов до цього етапу і позитивно відповів на останнє запитання, то однозначно рішення про видачу йому грошей буде позитивним. Скоротити шлях до будь-якого з рішень «Видати» або «Не видати» можна на будь-якому етапі.

Модель лінійної регресії – часто використовувана в статистиці модель лінійної залежності однієї змінної у від іншої, незалежної змінної x, або декількох змінних. Загальна лінійна регресійна модель має вигляд:

$$y = b_0 + b_1 x.$$

Уявімо, що маємо просту модель, в якій ми намагаємося використовувати вік, щоб передбачити, скільки покупець витратить на певний товар цього тижня.

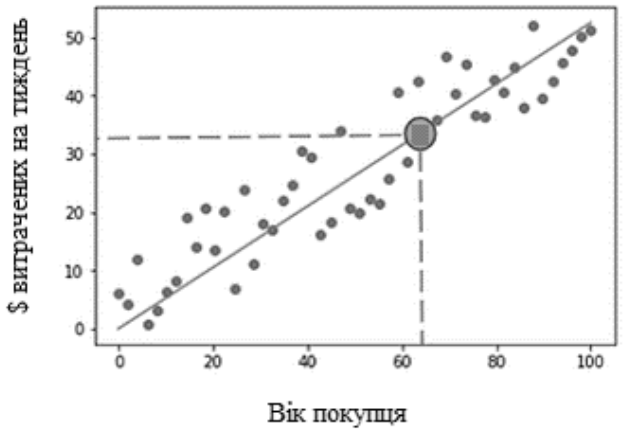


Рис. 3. Модель що описує залежність витрачених грошей від віку покупця

Точки на графіку – це точки, які використані для навчання моделі. Лінія між точками – це лінія найкращого підходу, яку створила модель, вона якнайкраще відображає напрямок цих точок. Згідно графіка, схоже, що чим старше хтось, тим більше грошей він витратить. Припустимо що ми знаємо вік покупця (65 років), тому ми знайдемо 65 на осі x і будемо йти по пунктирній лінії, доки не зустрінемо визначену «лінію найкращого підходу». Тепер ми можемо прослідкувати другу пунктирну лінію до осі y і визначити наше передбачення – ми передбачимо, що покупець витратить 33 долари цього тижня.

Для отримання «лінії найкращого підходу», у формулі лінійної регресії змінимо кілька змінних, додамо термін помилки (e), щоб врахувати випадковість, і заповнити нашу ціль ($ spent this week, витрачені гроші) та характеристики (age, вік).

$$y = mx + b$$
$$y = b + m1*x$$
$$\$ \text{ spent this week} = b + m_1*\text{age} + e$$
$$\$ \text{ spent this week} = b + m_1*\text{age} + m_2*\text{has\_kids} + \ldots + e$$

Рис. 4. Формули для опису моделі залежності витрачених грошей від віку покупця

Можна навчити модель, щоб дізнатися зв'язок між віком і витраченими на тиждень доларами на основі попередніх даних. Модель визначить значення m_1 і b, які найкраще передбачають витрачені долари цього тижня, враховуючи вік. Ми можемо легко додати більше функцій, таких як has_kids, і модель також дізнається значення m_2.

Поліноміальна лінійна регресія є статистичним методом, який використовує кілька пояснювальних змінних для прогнозування результату змінної відповіді. Метою множинної лінійної регресії є моделювання лінійного співвідношення між пояснювальними (незалежними) змінними та змінною реакції (залежною). Модель може бути описана наступною формулою:

$$y = b_0 + b_1x_1 + b_2x_2 + \ldots + b_nx_n.$$

В наведеній формулі y – відповідь на величини, тобто представляє результат, передбачений моделлю; b_0 – перехоплення, тобто значення y, коли x_i всі вони рівні 0; перша характеристика b_1 – коефіцієнт x_1; ще одна особливість b_n – коефіцієнт x_n; $x_1, x_2, \ldots, x_n$ є незалежними змінними моделі.

Наприклад, для оцінки викидів CO_2 автомобіля (залежна змінна y), потрібно врахувати потужність двигуна, кількість циліндрів та витрату палива. Останніми чинниками є незалежні змінні x_1, x_2 та x_3. Константи b_i є дійсними числами і називаються розрахунковими коефіцієнтами регресії моделі. Y – неперервна залежна величина.

Множинний регресійний аналіз – це метод, який використовується для виявлення впливу незалежних змінних на залежну змінну.

Розуміння того, як змінюється залежна змінна у міру зміни незалежних змінних, дозволяє нам передбачити наслідки чи наслідки змін у реальних ситуаціях. За допомогою багаторазової лінійної регресії можна зрозуміти, як змінюється артеріальний тиск, коли змінюється індекс маси тіла, враховуючи такі фактори, як вік, стать тощо, таким чином припускаючи, що може статися. Також можна отримати оцінки щодо тенденцій цін, таких як майбутня тенденція до нафти чи золота.

Машинне навчання. Підхід Unsupervised learning.

Навчання без учителя передбачає тільки наявність вхідних значень без відповідних вихідних значень в наборі даних для навчання.

Під час процесу неконтрольованого нав чання система не має конкретних наборів даних, і результати більшості проблем в основному невідомі.

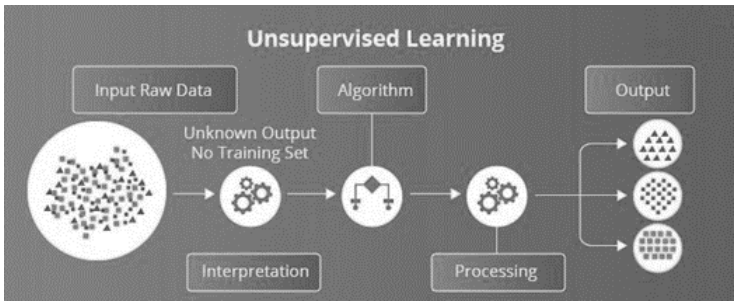


Рис. 5. Принцип роботи машинного навчання Unsupervised learning

Метод навчання без вчителя в основному використовується для кластеризації, або упорядкування об'єктів в порівняно однорідні групи.

У реальності добре розмічені дані – це велика рідкість, тому для їх розмітки зазвичай використовують або спеціальні сервіси, у яких реальні люди з країн з дешевою робочою силою за мінімальну плату вручну класифікують дані, або спеціальні алгоритми для розмітки (які, в свою чергу, можуть також використовувати машинне навчання). Великі корпорації, які мають величезний потік користувачів, можуть використовувати для розмітки своїх клієнтів. Наприклад, Google і його anticaptcha: знаходячи автобуси на фото, користувач підтверджує, що «він не робот», а також навчає нейронну мережу тому, який вигляд має автобус. До таких алгоритмів належать задачі кластеризації, зменшення розмірності і пошуку правил.

Алгоритми навчання без вчителя мають широкий спектр застосувань і корисні для вирішення реальних проблем, таких як виявлення аномалій, побудови рекомендаційних систем, групування документів або пошук клієнтів із спільними інтересами на основі їхніх покупок.

Кластерний аналіз використовується для групування або сегментування наборів даних із загальними атрибутами для визначення алгоритмічних зв'язків. Це розділ машинного навчання, який групує дані, які не були марковані чи класифіковані. Даний аналіз визначає спільність даних та реагує на основі наявності або відсутності таких спільних рис у кожному новому фрагменті даних (рис. 6).

Рис. 6. Схема роботи кластерного аналізу

Модель машинного навчання зможе зробити висновок про наявність двох різних класів, не знаючи нічого іншого з даних.

Алгоритми кластеризації використовують для сегментації ринку (типів покупців, лояльності), об'єднання близьких точок на карті, стиснення зображень, аналіз і розмітки нових даних, визначення аномальної поведінки.

Деякі з найпоширеніших алгоритмів кластеризації: метод К-середніх, ієрархічна кластеризація, кластеризація на основі щільності (DBSCAN).

Алгоритми К-середніх прості у реалізації та дуже ефективні з точки зору обчислень. Але вони не дуже добре ідентифікують класи, коли мають справу з групами, які не мають сферичної форми розподілу. Алгоритми К-середніх спрямовані на пошук і групування в класах точок даних, які мають високу схожість між ними. З точки зору алгоритму, ця подібність розуміється як протилежність відстані між точками даних. Чим ближче розташовані точки даних, тим більше вони будуть схожими та з більшою ймовірністю належати до одного кластеру.

Алгоритми К-середніх спрямовані на пошук і групування в класах точок даних, які мають високу схожість між ними. З точки зору алгоритму, ця подібність розуміється як протилежність відстані між точками даних. Чим ближче розташовані точки даних, тим більше вони будуть схожими та з більшою ймовірністю належати до одного кластеру.

Метод К-середніх є найбільш корисний, коли заздалегідь відома точна кількість кластерів і коли маємо справу зі сферичними розподілами.

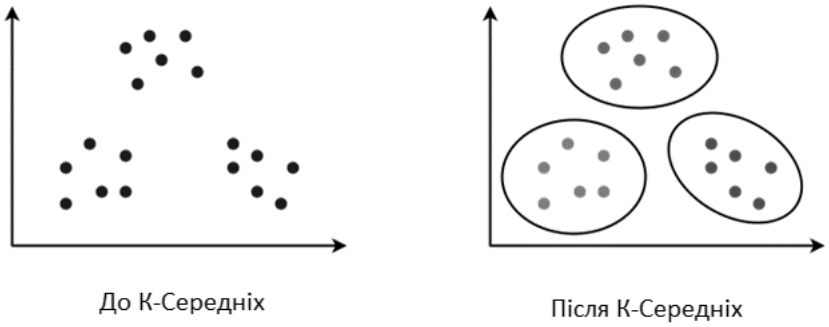


Рис. 7. Візуалізація результату до і після застосування методу К-середніх

Ієрархічна кластеризація є альтернативою алгоритмам кластеризації на основі прототипів. Основна перевага ієрархічної кластеризації полягає в тому, що не потрібно вказувати кількість кластерів, алгоритм визначить їх сам. Крім того, він дає змогу побудувати дендрограми, тобто візуалізації бінарної ієрархічної кластеризації.

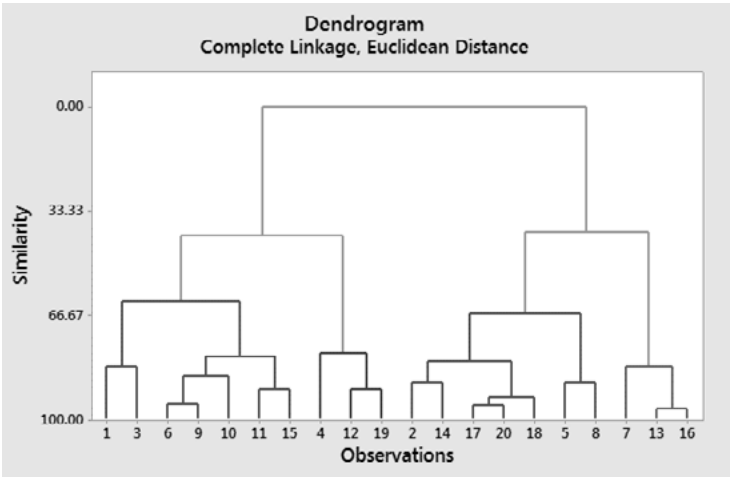


Рис. 8. Дендрограма при ієрархічній кластеризації

Перевагами ієрархічної кластеризації є те, що отримані ієрархічні уявлення можуть бути дуже інформативними, дендрограми забезпечують інформативний спосіб візуалізації. Вони особливо корисні, коли набір даних містить реальні ієрархічні зв'язки. Але методи ієрархічної кластеризації дуже чутливі

до викидів, і за їх наявності продуктивність моделі значно знижується. Крім того, вони дуже дорогі з точки зору обчислень.

У кластеризації на основі щільності, кластери визначаються як області з більшою щільністю, ніж решта набору даних. Об'єктами в розріджених районах, які необхідні для відокремлення скупчень, зазвичай вважаються точки шуму та кордону.

Найпопулярніший метод кластеризації на основі щільності є DBSCAN. Він має чітко визначену кластерну модель, яка називається «щільність-доступність». Подібно до кластеризації на основі зв'язків, вона базується на точках з'єднання в межах певних порогів відстані. Однак він з'єднує лише точки, які задовольняють критерію щільності, в оригінальному варіанті визначений як мінімальна кількість інших об'єктів у цьому радіусі. Кластер складається з усіх об'єктів, пов'язаних із щільністю (які можуть утворювати кластер довільної форми, на відміну від багатьох інших методів), а також усіх об'єктів, що знаходяться в межах діапазону цих об'єктів.

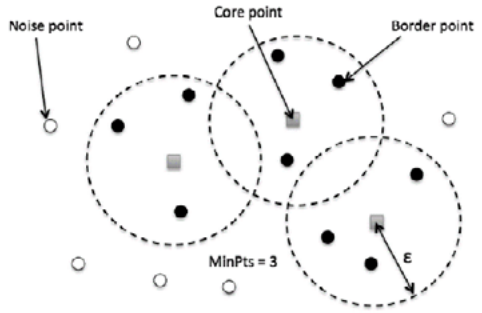


Рис. 9. Процес кластеризації за методом DBSCAN

Перевагами методу DBSCAN є те, що не потрібно вказувати кількість кластерів; існує велика гнучкість у формах і розмірах, які можуть приймати кластери; дуже корисно виявляти й працювати з даними шуму та викидами.

Недоліки методу в складнощах роботи з прикордонними точками, які доступні двом кластерами і в тому, що він не знаходить добре скупчення різної щільності.

Агентне моделювання

Агентне моделювання (agent-based modeling) це система яка побудована з автономних елементів, що приймають рішення. Кожен агент аналізує середовище і приймає рішення на основі набору правил. Агенти можуть виконувати різноманітні дії, що підходять для системи яку вони представляють, наприклад виробляти, споживати, продавати. Процес моделювання полягає в повторювані взаємодії агентів. Навіть у простому вигляді агентне моделювання може відтворювати складні шаблони поведінки і надавати інформацію про процеси у системах реального світу, що вони емулюють. Також агенти можуть еволюціонувати і показувати непередбачувану поведінку. Складні системи агентного моделювання може містити нейронні мережі, еволюційні алгоритми, або інші техніки навчання, щоб відтворити реалістичне навчання і адаптацію під різноманітні умови.

В багатьох випадках агентне моделювання найбільш природньо підходить для опису і симуляції систем, які складаються з сутностей, що відтворюють певну поведінку. Це може бути опис автомобільного руху, ринку акцій, або роботу організацій.

Ланцюги Маркова

Ланцюги Маркова є досить поширеним і відносно простим способом статистичного моделювання випадкових процесів. Вони використовувалися в багатьох різних областях, починаючи від генерації тексту і закінчуючи фінансовим моделюванням. Ланцюги Маркова концептуально досить інтуїтивно зрозумілі й доступні, оскільки їх можна реалізувати без використання складних статистичних чи математичних концепцій.

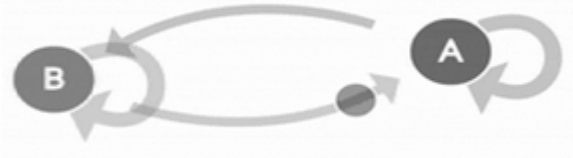


Рис. 10. Простий ланцюг Маркова з двома станами і чотирма переходами

Уявімо, що для погоди можливі два стани: сонячно або хмарно. Якщо ми ставимо задачу передбачити, якою буде погода завтра, інтуїтивно ми припускаємо, що в цьому процесі є невід'ємний перехід, оскільки поточна погода має певний вплив на те, якою буде погода наступного дня. Припустимо, за результатом багаторічних досліджень за погодою, розрахунок ймовірності настання сонячного дня після похмурого дня становить 0,25. Відповідно, оскільки є лише два можливих стани, ймовірність настання похмурого дня після похмурого дня має становити 0,75.

Можна використовувати цей розподіл, щоб передбачати погоду на найближчі дні на основі поточного стану погоди на даний момент.

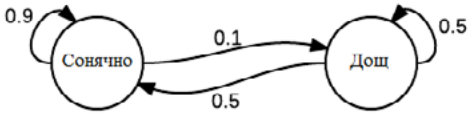


Рис. 11. Візуалізація прикладу погоди

Цей приклад ілюструє багато ключових понять ланцюга Маркова. Ланцюг Маркова по суті складається з набору переходів, які визначаються деяким розподілом ймовірностей, що задовольняють властивість Маркова. Розподіл ймовірностей отримується виключно шляхом спостереження переходів від поточного дня до наступного. Це ілюструє властивість Маркова, характеристику Марківських процесів, яка робить їх безпам'ятними, тобто такими що наступний стан залежить лише від поточного стану, а не від послідовності подій, що передували йому. Це, як правило, залишає їх не в змозі успішно створювати послідовності, в яких очікується певна основна тенденція. Наприклад, у той час як ланцюг Маркова може імітувати стиль письма автора на основі частоти слів, він не зможе створити текст, який містить глибоке значення або тематичне значення, оскільки не може враховувати повний ланцюжок попередніх станів.

Навчання з підкріпленням

Зазвичай для навчання моделей потрібна величезна кількість даних. Чим складніша модель, тим більше даних для неї може знадобитися. Але ці дані можуть бути недоступні або їх може не існувати. Крім того, зібрані дані можуть бути ненадійними, мати помилкові значення, чи бути застарілими. Ці проблеми можна подолати за допомогою навчання з підкріпленням.

В схемі навчання з підкріпленням приймають участь агент і середовище. На кожній ітерації середовище генерує стан та надає агенту спостереження. Агент, в свою чергу, вибирає одну з можливих дій, на що отримує винагороду та наступне спостереження. Рисунок 12 ілюструє дану схему взаємодії.

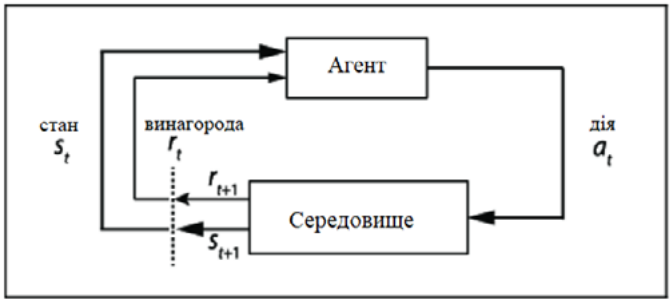


Рис. 12. Схема взаємодії середовища з агентом

Навчання з підкріпленням це галузь машинного навчання яка заснована на спробах і помилках. Агент отримує дані з середовища і вибирає наступну дію для даного стану. При виборі кожної дії агент може отримати певну винагороду. Рішення приймається для максимізації винагороди. Навчання з підкріпленням використовується в системах для знаходження найкращої поведінки чи шляху, що вибирається в специфічній ситуації (рис. 13).

Навчання з підкріпленням широко застосовується у різноманітних фінансових застосунках і у торгівлі акціями. У даних сферах вибрані дії можуть мати віддалений ефект, який не завжди можна миттєво виміряти. Деякі фінансові проблеми можуть бути вирішені прийняттям послідовних рішень. Комбінація спроб нових дій і повторного вибору дій на основі вже отриманих агентом даних може бути використана для покращення оптимальності навчання алгоритмами з підкріпленням.

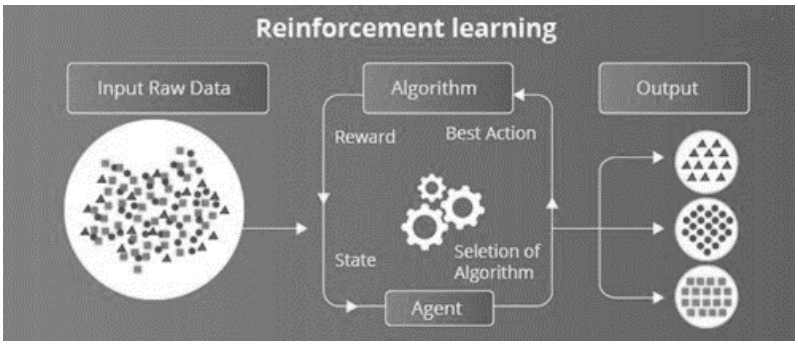


Рис. 13. Схема навчання з підкріпленням

Value-based підхід визначає стан або значення вартості для кожного стану. Мета агенту знайти таку стратегію яка оптимізує очікуване винагороду яка визначається функцією. Один з алгоритмів даного підходу Q-навчання.

Алгоритм має такі складові:

– r_t винагорода, яку отримає агень при переході від стану s_t до $s_{(t+1)}$;

– α – швидкість навчання, що визначає, якою мірою нова інформація перевизначатиме стару;

– γ – коефіцієнт знецінювання, що визначає більший ефект від винагород, отриманих раніше, ніж отриманих пізніше.

Алгоритм базується на рівняння Белмана що визначає ітеративне оновлення значень, використовуючи середньозважене старе значення та нову інформацію.

$$Q^{new}(s_t, a_t) \leftarrow \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \overbrace{\left(\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}} - \underbrace{Q(s_t, a_t)}_{\text{old value}} \right)}^{\text{temporal difference}}$$

$$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}_{\text{new value (temporal difference target)}}$$

Рис. 14. Алгоритм Q-навчання

Для неглибокого навчання значення Q-функції зберігається в таблиці:

	a_1	a_2	a_3	a_4
s_1	10	52	15	-2
s_2	14	30	8	7
s_3	42	0	-5	-10
s_4	-3	-1	-7	-20

↑ стани ↓

← дії →

Рис. 15. Таблиця Q-навчання

Перед початком навчання значення Q ініціалізують довільним значенням, наприклад 0. Потім на кожному кроці агент обирає дію α_t і переходить до нового стану S_{t+1} й таблиця Q оновлюється. В цій таблиці кожен елемент – це значення винагороди, яка змінюється при навчанні.

При навчанні на основі стратегій (англ. Policy Based Reinforcement Learning), агент використовує певні стратегії для вибору дії. Для Value-based методів агент підраховує значення функції вартості, але можливі також і інші правила для вибору дії, без підрахування функції вартості.

Стратегія визначається наступним чином $\pi_Q(s, a) = P[a \mid s, Q]$, де s – значення стану, а – дія, Q-параметри моделі. Тобто стратегія – це поведінка агента, або функція яка визначає дію для поточного стану.

Виділяють наступні види:

– детерміністичні стратегії – де дії чітко визначені;

– стохастичні стратегії – де є ймовірність вибору різних дій в даному стані.

Навчання на основі стратегій є більш ефективним в середовищах з великою кількістю станів. Велика кількість станів може призвести до значного використання пам'яті при інших алгоритмів.

Але навчання на основі стратегій дозволяє уникнути цього, так як алгоритм може використовувати лише деякі параметри системи, а не підраховувати значення для всіх можливих станів. Також перевагою даного методу є те, що він дозволяє використовувати стохастичні стратегії.

Висновки. В даній роботі був розглянутий можливий алгоритмічний і математичний апарат для систем побудови та аналізу інструментів управління персональними фінансами. Були наведені різні підходи машинного навчання, а саме Supervised learning, Unsupervised learning і навчання з підкріпленнями. Також було розглянуто різні типи задач, які можна вирішувати за допомогою наведених підходів. Дані алгоритми дають можливість для їх широкого застосування у різних сферах, а також для вирішення питань управління персональними фінансами.

Список використаних джерел:

1. R. S. Sutton, and A. G. Barto, Reinforcement learning: an introduction. Cambridge: The MIT Press, 2015, p. 143-160.
2. Sutton R. S., & Barto A. G. Introduction to reinforcement learning (Vol. 135). Cambridge: MIT press; 1998.
3. Ronald van Loon. Machine learning explained: Understanding supervised, unsupervised, and reinforcement learning. URL: https://bigdata-madesimple.com/machine-learning-explained-understanding-supervised-unsupervised-and-reinforcement-learning.
4. Антонюк Д. С., Вакалюк Т. А., Дідківський В. В., Візгалов О. Ю., Необхідність розробки симулятора управління персональними фінансами. *Інноваційна педагогіка: науковий журнал.* Вип. 24. Том 2. Видавничий дім «Гельветика», 2020. С. 208-212.
5. Кононова К. Ю. Машинне навчання: методи та моделі. МОН України, Харківський національний університет імені В. Н. Каразіна, 2020. С. 47-48
6. What are the types of Reinforcement learning algorithms? URL: https://www.finsliqblog.com/ai-and-machine-learning/what-are-the-types-of-reinforcement-learning-algorithms/
7. B. Hambly, R. Xu, H. Yang, Recent Advances in Reinforcement Learning in Finance, 2021 URL: https://arxiv.org/abs/2112.04553
8. Unsupervised learning URL: https://en.wikipedia.org/wiki/Unsupervised_learning.
9. Reinforcement learning URL: https://en.wikipedia.org/wiki/Reinforcement_learning.
10. Markov decision process URL: https://en.wikipedia.org/wiki/Markov_decision_process.
11. Q-learning URL: https://en.wikipedia.org/wiki/Q-learning.
12. Value-based Methods in Deep Reinforcement Learning. URL: https://towardsdatascience.com/value-based-methods-in-deep-reinforcement-learning-d40ca1086e1
13. Machine Learning and Linear Models: How They Work. URL: https://blog.dataiku.com/top-machine-learning-algorithms-how-they-work-in-plain-english-1
14. Introduction to Markov Chains. URL: https://towardsdatascience.com/introduction-to-markov-chains-50da3645a50d
15. Introducing the AI Economist: Why Salesforce Researchers are Applying Machine Learning to Economics. URL: https://www.salesforce.com/news/stories/introducing-the-ai-economist-why-salesforce-researchers-are-applying-machine-learning-to-economics.

References:

1. R. S. Sutton, and A. G. Barto, Reinforcement learning: an introduction. Cambridge: The MIT Press, 2015. P. 143-160. [in English].
2. Sutton R. S., & Barto A. G. Introduction to reinforcement learning (Vol. 135). Cambridge: MIT press; 1998. [in English].
3. Ronald van Loon. Machine learning explained: Understanding supervised, unsupervised, and reinforcement learning [Online] – Retrieved from: https://bigdata-madesimple.com/machine-learning-explained-understanding-supervised-unsupervised-and-reinforcement-learning. [in English].
4. Antoniuk D. S., Vakaliuk T. A., Didkivskyi V. V., Vizghalov O. Y. (2020). Neobhidnist rozrobky symuliatora upravlinnia personalnymy finansamy [The need of developing a personal finance management simulator]. Innovaciyna pedahohika: naukovyi zhurnal – Innovative Pedagogy: scientific journal, 24, Vol. 2. Vydavnychiy dim "Helvetyka" Derzhavnoho universytetu "Zhytomyrska politehnika" – Publishing House Helvetica. P. 208-212 [in Ukrainian].
5. K. Y. Kononova (2020). Mashynne navchannia: metody ta modeli [Machine learning: methods and models] MON Ukrainy, Kharkivskyi nacionalnyi universytet imeni V. N. Karazina – The Ministry of Education and Science of Ukraine, Kharkiv National University named after V.N. Karazin. P. 47-48 [in Ukrainian].
6. What are the types of Reinforcement learning algorithms? [Online] – Retrieved from: https://www.finsliqblog.com/ai-and-machine-learning/what-are-the-types-of-reinforcement-learning-algorithms/ [in English].
7. B. Hambly, R. Xu, H. Yang, Recent Advances in Reinforcement Learning in Finance, 2021 [Online] – Retrieved from: https://arxiv.org/abs/2112.04553 [in English].
8. Unsupervised learning [Online] – Retrieved from: https://en.wikipedia.org/wiki/Unsupervised_learning [in English].
9. Reinforcement learning [Online] – Retrieved from: https://en.wikipedia.org/wiki/Reinforcement_learning [in English].

10. Markov decision process [Online] – Retrieved from: https://en.wikipedia.org/wiki/Markov_decision_process [in English].
11. Q-learning [Online] – Retrieved from: https://en.wikipedia.org/wiki/Q-learning [in English].
12. Value-based Methods in Deep Reinforcement Learning. [Online] – Retrieved from: https://towardsdatascience.com/value-based-methods-in-deep-reinforcement-learning-d40ca1086e1 [in English].
13. Machine Learning and Linear Models: How They Work. [Online] – Retrieved from: https://blog.dataiku.com/top-machine-learning-algorithms-how-they-work-in-plain-english-1 [in English].
14. Introduction to Markov Chains. [Online] – Retrieved from: https://towardsdatascience.com/introduction-to-markov-chains-50da3645a50d [in English].
15. Introducing the AI Economist: Why Salesforce Researchers are Applying Machine Learning to Economics [Online] – Retrieved from: https://www.salesforce.com/news/stories/introducing-the-ai-economist-why-salesforce-researchers-are-applying-machine-learning-to-economics [in English].

УДК 519.8
DOI https://doi.org/10.32689/maup.it.2022.3.5

Дмитро ОЛЬХОВСЬКИЙ
кандидат фізико-математичних наук, доцент кафедри комп'ютерних наук та інформаційних технологій, Полтавський університет економіки і торгівлі, вул. Коваля, 3, Полтава, Україна, 36000 (dmitriy@olhovsky.name)

ORCID: 0000-0003-0313-6977

Олена ОЛЬХОВСЬКА
кандидат фізико-математичних наук, завідувач кафедри комп'ютерних наук та інформаційних технологій, Полтавський університет економіки і торгівлі, вул. Коваля, 3, Полтава, Україна, 36000 (lena@olhovsky.name)

ORCID: 0000-0001-5366-5995

Оксана ЧЕРНЕНКО
кандидат фізико-математичних наук, доцент кафедри комп'ютерних наук та інформаційних технологій, Полтавський університет економіки і торгівлі, вул. Коваля, 3, Полтава, Україна, 36000 (oksanachernenko7@gmail.com)

ORCID: 0000-0002-9084-0999

Тетяна ПАРФЬОНОВА
кандидат фізико-математичних наук, доцент кафедри комп'ютерних наук та інформаційних технологій, Полтавський університет економіки і торгівлі, вул. Коваля, 3, Полтава, Україна, 36000 (tapa.poltava@gmail.com)

ORCID: 0000-0001-9343-2061

Юрій ОЛЕКСІЙЧУК
кандидат фізико-математичних наук, доцент кафедри комп'ютерних наук та інформаційних технологій, Полтавський університет економіки і торгівлі, вул. Коваля, 3, Полтава, Україна, 36014 (olexijchuk@gmail.com)

ORCID: 0000-0002-0585-3307

Оксана ОРІХІВСЬКА
старший викладач кафедри комп'ютерних наук та інформаційних технологій, Полтавський університет економіки і торгівлі, вул. Коваля, 3, Полтава, Україна, 36014 (aka.jeita@gmail.com)

ORCID: 0000-0003-2775-0832

Артем ЗАДОРОЖНІЙ
аспірант кафедри комп'ютерних наук та інформаційних технологій, Полтавський університет економіки і торгівлі, вул. Коваля, 3, Полтава, Україна, 36014

Dmytro OLHOVSKIY
Candidate of Physical and Mathematical Sciences, Associate Professor, Associate Professor of Department of Computer Science and Information Technology, Poltava University of Economics and Trade, Koval str., 3, Poltava, Ukraine, 36000 (dmitriy@olhovsky.name)

Olena OLKHOVSKA
Candidate of Physical and Mathematical Sciences, Associate Professor, Associate Professor of Department of Computer Science and Information Technology, Poltava University of Economics and Trade, Koval str., 3, Poltava, Ukraine, 36000 (lena@olhovsky.name)

Oksana CHERNENKO
Candidate of Physical and Mathematical Sciences, Associate Professor, Associate Professor of Department of Computer Science and Information Technology, Poltava University of Economics and Trade, Koval str., 3, Poltava, Ukraine, 36000 (oksanachernenko7@gmail.com)

Tatyana PARFONOVA
Candidate of Physical and Mathematical Sciences, Associate Professor, Associate Professor of Department of Computer Science and Information Technology, Poltava University of Economics and Trade, Koval str., 3, Poltava, Ukraine, 36000 (tapa.poltava@gmail.com)

Yuriy OLEKSIYCHUK
Candidate of Physical and Mathematical Sciences, Associate Professor of Department of Computer Science and Information Technology, Poltava University of Economics and Trade, Koval str., 3, Poltava, Ukraine, 36000 (olexijchuk@gmail.com)

Oksana ORIKHIVSKA
Senior lecturer of Department of Computer Science and Information Technology, Poltava University of Economics and Trade, Koval str., 3, Poltava, Ukraine, 36000 (aka.jeita@gmail.com)

Artem ZADOROZNYI
Graduate student of Department of Computer Science and Information Technology, Poltava University of Economics and Trade, Koval str., 3, Poltava, Ukraine, 36000 (lena@olhovsky.name)

Бібліографічний опис статті: Ольховський, Д., Ольховська, О., Черненко, О., Парфьонова, Т., Олексійчук, Ю., Оріхівська, О., Задорожний, А. (2022). Розв'язування задач комбінаторної оптимізації ігрового типу на перестановках з обмеженнями на стратегії одного гравця. *Інформаційні технології та суспільство,* 3 (5), 41–48. DOI:

Bibliographic description of the article: Olkhovsky, D., Olkhovska, O., Chernenko, O., Parfyonova, T., Oleksiichuk, Yu., Orikhovska, O., Zadorozhny, A. (2022). Rozviazuvannia zadach kombinatornoi optymizatsii ihrovoho typu na perestanovkakh z obmezhenniamy na stratehii odnoho hravtsia [Solving game-type combinatorial optimization problems on permutations with constraints on the strategies of one player]. *Informatsiini tekhnolohii ta suspilstvo – Information technology and society,* 3(5), 41–48. DOI:

РОЗВ'ЯЗУВАННЯ ЗАДАЧ КОМБІНАТОРНОЇ ОПТИМІЗАЦІЇ ІГРОВОГО ТИПУ НА ПЕРЕСТАНОВКАХ З ОБМЕЖЕННЯМИ НА СТРАТЕГІЇ ОДНОГО ГРАВЦЯ

Задачі комбінаторної оптимізації на множині перестановок все частіше зустрічаються на практиці та потребують дослідження і розв'язання, тому постає необхідність розробки нових та модифікації вже існуючих методів для їх розв'язування.

***Мета роботи** – запропонувати нові методи розв'язування задач комбінаторної оптимізації ігрового типу на множинах перестановок, побудувати алгоритм розв'язування таких задач. Провести аналіз його складності, зокрема, дати теоретичну оцінку методу.*

***Методологія.** Для створення алгоритму розв'язування задач комбінаторної оптимізації ігрового типу на перестановках з обмеженнями на стратегії одного гравця використовувалися методи комбінаторної оптимізації та математичного програмування.*

***Наукова новизна**. У рамках дослідження задач комбінаторної оптимізації ігрового типу було вивчено можливість використання монотонного ітераційного алгоритму для розв'язування даного класу задач на множинах перестановок. У роботі проведено опис алгоритму розробленого монотонного ітераційного методу для пошуку ціни гри для розв'язування задач комбінаторної оптимізації ігрового типу на множині перестановок з обмеженнями на стратегії одного гравця. Розглянутий монотонний ітераційний алгоритм включає одинадцять кроків і дозволяє знайти ціну гри, заданої матрицею довільної вимірності та множиною пронумерованих перестановок – стратегіями першого гравця.*

Проведено оцінку складності запропонованого алгоритму. Для зручності викладення матеріалу введено необхідні позначення та пояснення. При розрахунку складності алгоритму визначено асимптотичну верхню границю з точністю до постійного множника. Знайдено теоретичну оцінку часу роботи монотонного ітераційного методу. Результати дослідження сформульовано у вигляді теореми з послідовно викладеним обґрунтованим доведенням.

Представлено ілюстративний приклад з метою застосування розробленого алгоритму. Детально розписано розв'язок завдання відповідно до кроків алгоритму. Проведено порівняння отриманого результату з розв'язками за іншими методами, зокрема, шляхом переходу від ігрової задачі комбінаторної оптимізації ігрового типу на множині перестановок до пари двоїстих задач лінійного програмування для матричної гри з платіжною матрицею

та ітераційним методом. Підтверджено коректність отриманих результатів на основі співпадіння відповідей, отриманих трьома різними способами.

Висновки. *Монотонний ітераційний метод дає змогу швидко отримати значення ціни гри із заданою точністю та оптимальну стратегію першого гравця, причому, як було встановлено, кількість кроків методу слабко залежить від розмірності задачі.*

Розроблений алгоритм монотонного ітераційного методу дозволив провести порівняння результатів з раніше відомими методами для підтвердження їх коректності.

Ключові слова: *комбінаторна оптимізація, монотонний ітераційний метод, теоретична оцінка складності методу.*

SOLVING GAME-TYPE COMBINATORIAL OPTIMIZATION PROBLEMS ON PERMUTATIONS WITH CONSTRAINTS ON THE STRATEGIES OF ONE PLAYER

Problems of combinatorial optimization for multiple permutations are increasingly encountered in practice and require research and solution, therefore there is a need to develop new and modify existing methods for their solution.

The purpose of the work *is to propose new methods of solving combinatorial optimization problems of the game type on multiple permutations, to build an algorithm for solving such problems. Conduct an analysis of its complexity, in particular, give a theoretical assessment of the method.*

Methodology. *Combinatorial optimization and mathematical programming methods were used to create an algorithm for solving game-type combinatorial optimization problems on permutations with restrictions on the strategies of one player.*

Scientific novelty. *As part of the study of game-type combinatorial optimization problems, the possibility of using a monotone iterative algorithm for solving this class of problems with multiple permutations was studied. The paper describes the algorithm of the developed monotonic iterative method for finding the price of the game for solving problems of combinatorial optimization of the game type on a set of permutations with restrictions on the strategy of one player. The considered monotonic iterative algorithm includes eleven steps and allows finding the price of the game given by a matrix of arbitrary dimension and a set of numbered permutations – the strategies of the first player.*

The complexity of the proposed algorithm was evaluated. For the convenience of presenting the material, the necessary designations and explanations have been introduced. When calculating the complexity of the algorithm, an asymptotic upper bound is determined with accuracy up to a constant factor. A theoretical estimation of the working time of the monotone iterative method was found. The results of the research are formulated in the form of a theorem with a consistently presented substantiated proof.

An illustrative example is presented for the purpose of applying the developed algorithm. The solution of the task is described in detail according to the steps of the algorithm. The obtained result is compared with solutions by other methods, in particular, by moving from the game problem of combinatorial optimization of the game type with multiple permutations to a pair of dual problems of linear programming for a matrix game with a payment matrix and an iterative method. The correctness of the obtained results was confirmed based on the coincidence of the answers obtained in three different ways.

Conclusions. *The monotone iterative method allows us to quickly obtain the value of the price of the game with a given accuracy and the optimal strategy of the first player, and, as it was found, the number of steps of the method depends weakly on the dimension of the problem.*

The developed algorithm of the monotonic iterative method made it possible to compare the results with previously known methods to confirm their correctness.

Key words: *combinatorial optimization, monotonic iterative method, theoretical evaluation of the complexity of the method.*

Постановка проблеми в загальному вигляді та її зв'язок з важливими науковими чи практичними завданнями. Задачі комбінаторної оптимізації на комбінаторних множинах все частіше зустрічаються на практиці та потребують дослідження і розв’язування [1-9]. Окремим класом можна виділити задачі комбінаторної оптимізації ігрового типу з обмеженнями, що визначені різними типами комбінаторних конфігурацій. Даний клас задач є доволі новим, а отже потребує розробки нових та дослідження вже існуючих методів.

Аналіз останніх досліджень і публікацій. У роботі [2] розглянута постановка задачі комбінаторної оптимізації ігрового типу з обмеженнями, що визначені множиною перестановок на стратегії одного гравця. У [3] запропоновано, а у [4] обґрунтовано ітераційний метод розв’язування задач комбінаторної оптимізації. У [5] розглянута оцінка швидкості збіжності даного методу. У роботі [6] проведено порівняння відомих методів розв’язування ігрових задач та наведені числові експерименти.

У даній публікації розглядається алгоритм модифікованого ітераційного методу для розв’язування задач комбінаторної оптимізації ігрового типу та проведена оцінка складності роботи алгоритму.

Постановка завдання. В даній публікації пропонується розглянути алгоритм модифікованого ітераційного методу для розв’язування задач комбінаторної оптимізації ігрового типу та провести оцінку складності його роботи.

Виклад основного матеріалу дослідження. Для математичної моделі задачі з роботи [2] розглянемо алгоритм її розв’язування, а саме, задачі комбінаторної оптимізації ігрового типу з обмеженнями, що визначені множиною перестановок на стратегії одного гравця. Монотонний ітераційний метод, як

ітераційний процес, дозволяє знайти υ – ціну гри Γ_A, що задана матрицею $A' = \left(a'_{ij}\right)$ вимірності $m \times n$ та множиною пронумерованих перестановок $E_{mv}\left(P^x\right)$ – стратегіями першого гравця.

На нульовому кроці перший гравець обирає довільну перестановку

$$x^0 = x_\tau = \left(x^0_{\tau_1}, x^0_{\tau_2}, ..., x^0_{\tau_m}\right) \in E_{mv}\left(P^x\right),\ \gamma^0 = \left(0,0,...,0,1,...,0\right),$$

де в γ^0 одиниця стоїть на місці $\tau \in J_L$ номера перестановки x^0 в $E_{mv}\left(P^x\right)$, $L = \left|E_{mv}\left(P^x\right)\right|$. Визначається допоміжний вектор c^0, як вектор скалярних добутків стовпців матриці A' та вектору-перестановки x^0, координати якого зважені координатами γ^0, тобто

$$c^0_j = \sum_{t=1}^{m} \gamma^0_\tau a'_{ij} x^0_{\tau t},\ \forall j \in J_n,\ c^0 = \left(c^0_1, ..., \mathrm{c}^0_n\right),$$

де $\gamma_i = 1$ – ймовірність використання перестановки $x \quad x$.

Крок 1. Встановлюється початковий номер ітерації N рівний 1, $N = 1$.

Крок 2. Визначається

$$\underline{\upsilon}^{N-1} = \min_{j \in J_n} c^{N-1}_j \tag{1}$$

та позначається множина індексів

$$J^N = \left\{j^{N-1}_1, ..., j^{N-1}_\gamma\right\}, \tag{2}$$

на яких досягається $\underline{\upsilon}^{N-1}$, тобто $J^N = \arg\min_{j \in J_n} c^{N-1}_j$.

Крок 3. Формується матрична гра $\Gamma^N \subset \Gamma_A$ як підгра гри Γ_A з матрицею $A^N = \left(a^N_{ij}\right)$, де $j \in J^N$, $j \in J^N$.

Крок 4. На основі Γ^N формується матрична $l \times \gamma$ $(l \geq \gamma)$ гра з матрицею B^N. Для цього визначається множина E^N перестановок x^N, кожна з яких доставляє максимальне на множині перестановок $E_{mv}(P^x)$ значення скалярного добутку стовпців матриці A^N та перестановок з $E_{mv}\left(P^x\right)$. Якщо перестановок, які для певного стовпця матриці A^N дають максимальне значення, декілька, то обираються в E^N всі такі перестановки, $E^N \subset E_{mv}\left(P^x\right)$. Кількість перестановок в E^N при цьому позначається l. Величина l може бути в межах від одного до m! (в гіршому випадку, коли $a'_{ij} = const\ \forall i \in J_m$). Зі значень скалярних добутків стовпців матриці A^N та перестановок E^N з E^N утворюється матрична гра з матрицею $B^N = \left(b_{ij}\right)^{j=\overline{1,\gamma}}_{i=\overline{1,l}}$, де $b_{ij} = \sum_{t=1}^{m} a^N_{ij} x^N_{it}$, $i \in J_l$, $j \in J^N$, $x^N_i = \left(x^N_{i1}, ..., x^N_{im}\right) \in E^N$, де $i \in J_l$. Слід зауважити, що $l = \gamma$, якщо кожне максимальне значення скалярних добутків стовпців з A^N та $x^N_i \in E_{mv}\left(P^x\right)$ досягається тільки на одній перестановці.

Крок 5. Розв'язується гра з матрицею B^N. Визначається $\tilde{\gamma}^N = \left(\tilde{\gamma}^N_1, ..., \tilde{\gamma}^N_l\right)$ – ймовірність перестановок $x^N_i = \left(x^N_{i_1}, ..., x^N_{i_m}\right)$, $i \in J_l$ що відповідають рядкам матриці B^N.

Крок 6. Визначається вектор $\tilde{c}^N = \left(\tilde{c}^N_1, ..., \tilde{c}^N_n\right)$ за формулою:

$$\tilde{c}^N_j = \sum_{i=1}^{l} \tilde{\gamma}^N_i \left(\sum_{t=1}^{m} x^N_{it} a'_{tj}\right),\ \forall j \in J_n. \tag{3}$$

Крок 7. Розглядається гра з матрицею вимірності 2 x n: $\begin{pmatrix} c^{N-1}_1 \ldots\ c^{N-1}_n \\ \tilde{c}^N_1 \quad \ldots\ \tilde{c}^N_n \end{pmatrix}$.

Визначається оптимальна стратегія $\left(1-\alpha_N, \alpha_N\right)$, $0 \leq \alpha_N \leq 1$ першого гравця в цій грі. При цьому

$$\alpha_N = \arg\max_{\alpha} \min_{j} \left((1-\alpha) c^{N-1}_j + \alpha \tilde{c}^N_j\right). \tag{4}$$

Крок 8. Якщо $\alpha_N = 0$, то зупинка алгоритму з ціною гри $\underline{\upsilon}^{N-1}$ та ймовірностями застосування стратегій-перестановок, що складають вектор γ^N, інакше – перехід на наступний крок алгоритму.

Крок 9. Обчислюється значення вектору $\gamma^N = \left(\gamma^N_1, ..., \gamma^N_L\right)$ за наступною формулою:

$$\gamma^N = \left(1-\alpha_N\right)\gamma^{N-1} + \alpha_N \tilde{\tilde{\gamma}}^N, \tag{5}$$

де x^N_i – це вектор довжини L, утворений з координат вектора x^N_i, розташованих згідно нумерації перестановок x^N_i в множині $E_{mv}(P^x)$, інші елементи – нулі.

Крок 10. Визначаються компоненти вектору $c^N = \left(c^N_1, ..., c^N_n\right)$ за формулою:

$$c^N = \left(1-\alpha_N\right)c^{N-1} + \alpha_N \tilde{c}^N. \tag{6}$$

Крок 11. Збільшується номер N ітерації на 1, виконується перехід на крок 2 алгоритму.

Оцінка складності алгоритму монотонного ітераційного методу

Для алгоритму монотонного ітераційного методу було проведено оцінку складності. Введемо позначення:

– P_i – кількість перестановок на поточній ітерації i (при виявленні однакових елементів);
– S_i – кількість стратегій, які використовуються на поточній ітерації;
– $O(m\log m)$ – відома складність алгоритму сортування;
– n^3 та S_i^3 – необхідний машинний час для двоїстого симплекс-методу [10; 11].

При розрахунку складності алгоритму потрібно визначити асимптотичну верхню границю з точністю до постійного множника [11]. Для функції $g(n)$ позначка $O(g(n)) = f(n)$ [3] означає множину функцій таких, що існує додатна константа c і n_0 такі, що $0 \le f(n) \le cg(n)$ для всіх $n \ge n_0$.

Підрахуємо: $T = \sum_{i=1}^{61} c_j \tau_j$:

$$T = 1(c_{13} + c_{14} + c_{20} + c_{21} + c_{26} + c_{31} + c_{40} + c_{45}) +$$
$$+m(c_1 + c_2 + c_3 + c_{15} + c_{16} + c_{22} + c_{23} + c_{32} + c_{33} + c_{41} + c_{42}) +$$
$$+n(c_4 + c_5 + c_8 + c_9 + c_{10} + c_{11} + c_{12} + c_{34} + c_{35} + c_{43} + c_{44} + c_{46} + c_{47} + c_{48} +$$
$$+ c_{49} + c_{50} + c_{51} + c_{52}) + n \cdot m(c_6 + c_7) + S_i \cdot c_{17} + S_i \cdot m \cdot (c_{18} + c_{19}) +$$
$$+P_i \cdot (c_{24} + c_{25} + c_{27} + c_{28} + c_{29} + c_{30}) + P_i \cdot n(c_{36} + c_{37}) + P_i \cdot n \cdot m(c_{38} + c_{39}) +$$
$$+T_0(m) + T_2(n^2) + T_3(S_i^3) + T_1(P_i!) + T_4(n^2) + T_5(n^3) +$$
$$+O(m\log m) + O(m\log m) + O(P_i \log P_i) + T_6(1)$$

або

$$T = O(1) + O(m) + O(n) + O(n \cdot m) + O(S_i) + O(S_i \cdot m) + O(P_i) +$$
$$+O(P_i \cdot n) + O(P_i \cdot n \cdot m) + O(m\log m) + O(m\log m) + O(P_i \log P_i) +$$
$$T = O(1) + O(m) + O(n) + O(n \cdot m) + O(S_i) + O(S_i \cdot m) + O(P_i) + .$$

Враховуючи, що $T_i(f) = O(f)$ приймаємо:

$$T = O(1) + O(m) + O(n) + O(n \cdot m) + O(S_i) + O(S_i \cdot m) + O(P_i) +$$
$$+O(P_i \cdot n) + O(P_i \cdot n \cdot m) + O(m\log m) + O(m\log m) + O(P_i \log P_i) + O(m) +$$
$$+O(P_i!) + O(n^2) + O(S_i^3) + O(n^2) + O(n^3) + O(1).$$

Враховуючи, що $\forall c > 0 : cO(f(n)) = O(f(n))$, маємо

$$T = O(m) + O(n) + O(n \cdot m) + O(S_i) + O(S_i \cdot m) + O(P_i) + O(P_i \cdot n) +$$
$$+O(P_i \cdot n \cdot m) + O(m\log m) + O(P_i \log P_i) + O(P_i!) + O(n^2) + O(S_i^3) + O(n^3).$$

Відомо, що якщо $\lim_{n\to\infty} \frac{f(n)}{g(n)} = c < \infty$, то $O(f(n)) + O(g(n)) = O(f(n) + g(n))$ та якщо $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$, то $O(f(n)) + O(g(n)) = O(f(n))$. З огляду на це складність алгоритму набуде вигляду:

$$T = O(m+n) + O(n \cdot m) + O(S_i + P_i) + O(S_i \cdot m) + O(P_i \cdot n) + O(P_i \cdot n \cdot m) +$$
$$+O(m\log m) + O(P_i \log P_i) + O(P_i!) + O(n^2) + O(S_i^3) + O(n^3) =$$
$$= O(P_i \cdot n \cdot m) + O(m\log m) + O(P_i!) + O(n^3) =$$
$$T = O(P_i \cdot n \cdot m + m\log m + P_i! + n^3) \cdot$$

Тобто маємо:

$$T = O(P_i \cdot n \cdot m + m\log m + P_i! + n^3).$$

Таким чином, доведено теорему:

Теорема. Час роботи монотонного ітераційного методу має оцінку: $T = O(P_i \cdot n \cdot m + m\log m + P_i! + n^3)$.

Ілюстративний приклад

Нехай задана матрична гра 3×3 із платіжною матрицею $A = \begin{pmatrix} 1 & 2 & 3 \\ 5 & 3 & 1 \\ 3 & 2 & 4 \end{pmatrix}$ за умовою, що на стратегії першого гравця накладаються комбінаторні обмеження, що визначені перестановками $E_{mv}\left(P^x\right)$ з множини $P^x = \{0.1; 0.3; 0.6\}$. Знайти ціну гри ЗКОІТП та ймовірність застосування стратегій-перестановок першим гравцем. Нехай перестановки пронумеровані так: $x_1 = (0.1; 0.3; 0.6)$; $x_2 = (0.1; 0.6; 0.3)$; $x_3 = (0.3; 0.1; 0.6)$; $x_4 = (0.3; 0.6; 0.1)$; $x_5 = (0.6; 0.1; 0.3)$; $x_6 = (0.6; 0.3; 0.1)$, а $L = 3! = 6$.

Крок 0. Перший гравець обирає довільну перестановку $x^0 = (x_1, x_2, x_3) \in E_{mv}\left(P^x\right)$. Нехай $x^0 = (0.1; 0.3; 0.6)$, $\gamma^0 = (1; 0; 0; 0; 0; 0)$.

Визначаємо допоміжний вектор c^0, кожен елемент якого c_j^0 це скалярний добуток стовпця j матриці A та перестановки x^0, тобто $c_j^0 = \sum_{i=1}^{3} a_{ij} x_i$, $\forall j \in J_n$, $c = \left(c_1^0, ..., c_n^0\right)$. Застосувавши до умови задачі, отримаємо: $c^0 = (3.4; 2.3; 3.0)$.

Крок 1. Встановлюємо номер ітерації N =1.

Крок 2. Визначаємо $\underline{\upsilon}^0$ за (4.1), $\underline{\upsilon}^0 = \min\limits_{j \in J_n} c_j^0 = \min\{3.4; 2.3; 3.0\} = 2.3$ та за (4.2) визначаємо множину індексів J^0, $J^N = \arg\min\limits_{j \in J_n} c_j^{N-1}$, на яких досягається $\underline{\upsilon}^0$, $J^0 = \{2\}$.

Крок 3. Формуємо матричну гру $\Gamma^1 \subset \Gamma_A$ як підгру гри Γ_A з матрицею $A^1 = \left(a_{ij}\right)$, де $i = J_3$, $j \in J^0$:

$$A^1 = \begin{pmatrix} 2 \\ 3 \\ 2 \end{pmatrix}.$$

Крок 4. На основі Γ^N сформуємо матричну $l \times 1$ гру. Для цього визначимо (за теоремою 3.1 з [1]) перестановки E^N, які доставляють максимальне на множині перестановок $E_{mv}(P^x)$ значення скалярного добутку елементів стовпців матриці $A^N = A^1$ та перестановки. Згідно умови задачі $E^1 = \{(0.1; 0.6; 0.3); (0.3; 0.6; 0.1)\}$. Отримаємо матрицю B^1, яка на поточній ітерації складається з двох рядків: $B^1 = \begin{pmatrix} 2.6 \\ 2.6 \end{pmatrix}$.

Крок 5. Розв'язавши матричну гру з отриманою матрицею B^1, визначимо оптимальну стратегію. У нашому випадку отримано розв'язок $\tilde{\gamma}^1 = (0,1)$, відповідно до якого оптимальною стратегією є $x_1^1 = (0.3; 0.6; 0.1)$.

Крок 6. За (3), визначаємо вектор $\tilde{c}^1 = \left(\tilde{c}_1^1, ..., \tilde{c}_n^1\right)$, тобто:

$\tilde{c}_1^1 = 1 \cdot (0.3 \cdot 1 + 0.6 \cdot 5 + 0.1 \cdot 3) = 3.6$, $\tilde{c}_3^1 = 1 \cdot (0.3 \cdot 3 + 0.6 \cdot 1 + 0.1 \cdot 4) = 1.9$, $\tilde{c}_3^1 = 1 \cdot (0.3 \cdot 3 + 0.6 \cdot 1 + 0.1 \cdot 4) = 1.9$,

таким чином $\tilde{c}^1 = (3.6; 2.6; 1.9)$.

Крок 7. Розглянемо гру з матрицею $\begin{bmatrix} c_1^{N-1} & \dots & c_n^{N-1} \\ \tilde{c}_1^N & \dots & \tilde{c}_n^N \end{bmatrix}$ вимірності $2 \times n$:

$$\begin{bmatrix} c^0 \\ \tilde{c}^1 \end{bmatrix} = \begin{bmatrix} 3.4 & 2.3 & 3 \\ 3.6 & 2.6 & 1.9 \end{bmatrix}.$$

Розв'язавши цю матричну гру, отримаємо оптимальну стратегію першого гравця $(0.5; 0.5)$. При цьому в ролі α_1 обираємо значення, яке задовольняє (4). Тобто $\alpha_1 = 0.5$.

Крок 8. $\alpha_1 \neq 0$, тому переходимо на наступний крок алгоритму.

Крок 9. Обчислюємо значення вектору $\gamma^N = \left(\gamma_1^N, ..., \gamma_L^N\right)$ за (5), маємо:

$$\gamma^1 = (1 - 0.5)(1; 0; 0; 0; 0; 0) + 0.5(0; 0; 0; 1; 0; 0) = (0.5; 0; 0; 0.5; 0; 0).$$

Крок 10. Визначаємо компоненти вектору $c^1 = \left(c_1^1, ..., c_n^1\right)$ за (6), отримаємо:

$$c^1 = (1 - 0.5)(3.4; 2.3; 3) + 0.5(3.6; 2.6; 1.9) = (3.5; 2.45; 2.45).$$

Крок 11. Збільшуємо номер N ітерації на 1, переходимо на крок 2 алгоритму.

Крок 2. Визначаємо $\underline{\upsilon}^1$ як $\underline{\upsilon}^1 = \min\limits_{j \in J_n}(3.5; 2.45; 2.45) = 2.45$. Визначаємо множину індексів J^1, на яких досягається $\underline{\upsilon}^1$, $J^1 = \{2; 3\}$.

Крок 3. Формуємо матричну гру $\Gamma^2 \subset \Gamma_A$ як підгру гри Γ_A з матрицею $A^2 = \left(a_{ij^2}\right)$, де $i = J_m$, $j^1 \in J^1$:

$$A^2 = \begin{pmatrix} 2 & 3 \\ 3 & 1 \\ 2 & 4 \end{pmatrix}.$$

Крок 4. Для знаходження вектору $\tilde{x}^2$ визначаємо перестановки, що складають множину $E^2 = \{(0.1; 0.6; 0.3); (0.3; 0.6; 0.1); (0.3; 0.1; 0.6)\}$. Сформуємо матрицю B^2 матричної гри:

Крок 5. Розв'язавши отриману на попередньому кроці гру з матрицею B^2, визначимо перестановки $\tilde{x}_1^2 = (0.1; 0.6; 0.3)$ та $\tilde{x}_2^2 = (0.3; 0.1; 0.6)$, що відповідають визначеній стратегії $\tilde{\gamma}^2 = (0.72, 0, 0.28)$.

Крок 6. Визначаємо вектор $\tilde{c}^2$ за (3). Тобто $\tilde{c}^2 = (3.61; 2.46; 2.46)$.

Крок 7. Складемо гру з матрицею вимірності $2 \times n$:

$$\begin{bmatrix} c^1 \\ \tilde{c}^2 \end{bmatrix} = \begin{bmatrix} 3.5 & 2.45 & 2.45 \\ 3.61 & 2.46 & 2.46 \end{bmatrix}.$$

Розв'язавши дану гру знайдено оптимальну стратегію $(0; 1)$ першого гравця в цій грі. При цьому α_2 обираємо за (4). Отримали $\alpha_2 = 1$.

Крок 8. $\alpha_2 \neq 0$, тому переходимо на наступний крок алгоритму.

Крок 9. Обчислюємо значення вектору $\gamma^N = \left(\gamma_1^N, ..., \gamma_L^N\right)$ за (5), маємо:

$$\gamma^2 = 1(0;0.72;0.28;0;0;0) = (0;0.72;0.28;0;0;0).$$

Крок 10. Визначаємо компоненти вектору $c^2 = \left(c_1^2, ..., c_n^2\right)$ за (4.6), отримаємо:

$$c^2 = 1 \cdot (3.61;2.46;2.46) = (3.61;2.46;2.46).$$

Крок 11. Збільшуємо номер $\underline{\upsilon}^2$ ітерації на 1, переходимо на крок 2 алгоритму.

Крок 2. Визначаємо $\underline{\upsilon}^2$ як $\underline{\upsilon}^2 = \min\limits_{j \in J_n}(3.61;2.46;2.46) = 2.46$. Визначаємо множину індексів J^2, на яких досягається $\underline{\upsilon}^2$, $J^2 = \{2;3\}$.

Крок 3. Формуємо матричну гру $\Gamma^3 \subset \Gamma_A$ як підгру гри Γ_A з матрицею $A^3 = \left(a_{ij^3}\right)$, де $i = J_m$, $j^2 \in J^2$:

$$A^3 = \begin{pmatrix} 2 & 3 \\ 3 & 1 \\ 2 & 4 \end{pmatrix}.$$

Крок 4. Для знаходження вектору $\tilde{x}^3$ визначаємо перестановки, що складають множину $E^3 = \{(0.1;0.6;0.3);(0.3;0.6;0.1);(0.3;0.1;0.6)\}$. Сформуємо матрицю B^3 матричної гри:

$$B^3 = \begin{pmatrix} 2.6 & 2.1 \\ 2.6 & 1.9 \\ 2.1 & 3.4 \end{pmatrix}.$$

Крок 5. Розв'язавши отриману на попередньому кроці гру з матрицею B^3, визначимо перестановки $\tilde{x}_1^3 = (0.1;0.6;0.3)$ та $\tilde{x}_2^3 = (0.3;0.1;0.6)$, що відповідають визначеній стратегії $\tilde{\gamma}^3 = (0.72, 0, 0.28)$.

Крок 6. Визначаємо вектор $\tilde{c}^3$ за (3). Тобто $\tilde{c}^3 = (3.61;2.46;2.46)$.

Крок 7. Складемо гру з матрицею вимірності $2 \times n$:

$$\begin{bmatrix} c^2 \\ \tilde{c}^3 \end{bmatrix} = \begin{bmatrix} 3.61 & 2.46 & 2.46 \\ 3.61 & 2.46 & 2.46 \end{bmatrix}.$$

Розв'язавши дану гру знайдено оптимальну стратегію $(1;0)$ першого гравця в цій грі. При цьому α_3 обираємо за (4.4). Отримали $\alpha_3 = 0$. Оскільки виконався критерій зупинки алгоритму ($\alpha_3 = 0$), то вихід з алгоритму з ціною гри $\underline{\upsilon}^3 = \upsilon = 2.46$ та ймовірністю використання стратегії $(0.3;0.1;0.6)$, рівної 0,28 та стратегії $(0.1;0.6;0.3)$, рівної 0,72.

Цю задачу було розв'язано також двома відомими методами: шляхом переходу від ігрової задачі комбінаторної оптимізації ігрового типу на множині перестановок до пари двоїстих задач лінійного програмування для матричної гри з платіжною матрицею та ітераційним методом з [6]. Знайдені значення ціни гри в обох методах збіглися із отриманим значенням при розв'язку задач комбінаторної оптимізації ігрового типу на множині перестановок запропонованим монотонний ітераційний метод, що підтверджує коректність отриманих результатів.

Висновки з даного дослідження та перспективи подальших розвідок у даному напрямі. У публікації на основі монотонного алгоритму розроблено монотонний ітераційний метод пошуку ціни гри для розв'язування задач комбінаторної оптимізації ігрового типу на множині перестановок з обмеженнями на стратегії одного гравця. Запропонований монотонний ітераційний метод дає змогу швидко отримати значення ціни гри із заданою точністю та оптимальну мішану стратегію першого гравця, причому кількість кроків методу слабко залежить від вимірності задачі. Також одержано теоретичну оцінку складності запропонованого монотонного ітераційного методу.

У подальшому доцільно застосувати даний алгоритм для розв'язування практичних задач економічного характеру, що зводяться до задач комбінаторної оптимізації та розв'язуються зазначеним методом.

Список використаних джерел:

1. Стоян Ю. Г., Ємець О.О. Теорія і методи евклідової комбінаторної оптимізації. К. : ІСДО, 1993. 188 с.

2. Емец О. А., Устьян Н. Ю. Исследование математических моделей и методов решения задач на перестановках игрового типа. *Кибернетика и сист. анализ.* 2007. № 6. С. 103-114.

3. Ємець О. О., Устьян Н.Ю. Один ітераційний метод розв'язування ігрових задач на перестановках. *Наукові вісті НТУУ «КПІ».* 2008. № 3. С. 5-10.

4. Емец О. А., Ольховская Е.В. Доказательство сходимости итерационного метода решения задачи комбинаторной оптимизации игрового типа на размещениях. *Кибернетика и сист. анализ.* 2013. № 1. С. 102-114.

5. Ольховська О. В. Оцінка швидкості збіжності ітераційного методу розв'язування комбінаторних оптимізаційних задач ігрового типу. *Таврійський вісник інформатики і математики*. 2014. № 1. С. 31-42.

6. Емец О. А., Ольховський Д. М., Ольховская Е. В. Сравнение методов решения игровых задач: числовые эксперименты. *Искусственный интеллект*. 2014. №1. С. 47-56.

7. Ємець О.О., Черненко О.О., Чілікіна Т. В., Ольховська О. В. Огляд задач комбінаторної оптимізації визначення рентабельності сільськогосподарського виробництва та методи їх розв'язування. *Математичне та комп'ютерне моделювання. Серія: фізико-математичні науки*. Випуск 22. 2021. С. 63-74.

8. Ольховський Д., Ольховська О., Черненко О., Парфьонова Т., Чілікіна Т. Програмний комплекс для розв'язування евклідових комбінаторних оптимізаційних задач точними та наближеними методами. *Інформаційні технології та суспільство,* (2 (4). 2022. С. 78-87.

9. Юрій Олексійчук, Дмитро Ольховський, Олена Ольховська, Тетяна Чілікіна, Оксана Черненко, Оксана Оріхівська. Комбінаторна задача про побудову мостів та методи її розв'язання. *Вісник Кременчуцького національного університету імені Михайла Остроградського*. Кременчук : КРНУ, 2022. Випуск 1(132). С. 115-122.

10. Brown G.W. Iterative solution of games by fictitious play. *Activity analysis of production and allocation: Proceedings of a Conference*. 1951. New York. P. 374-376.

11. Алгоритмы: построение и анализ, 2-е изд. / Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. М.: Вильямс, 2005. 1296 с.

References:

1. Stoyan, Yu. G., Yemets, O.O. (1993). *Teoriia i metody evklidovoi kombinatornoi optymizatsii [Theory and methods of Euclidean combinatorial optimization].* K. : Institute for Systems Research in Education [in Ukrainian].

2. Emets O. A., Ustyan N. Yu. (2007). Yssledovanye matematycheskykh modelei y metodov reshenyia zadach na perestanovkakh yhrovoho typa [Study of mathematical models and methods for solving problems on permutations of the game type]. *Kibernetika i sistemnyj analiz – Cybernetics and systems analysis*. No. 6. P. 103-114 [in Russian].

3. Yemets O.O., Ustyan N. Yu. (2008) Odyn iteratsiinyi metod rozviazuvannia ihrovykh zadach na perestanovkakh [One iterative method of solving game problems on permutations]. *Naukovi visti NTUU "KPI". Scientific news of NTUU "KPI"*. No. 3. P. 5-10 [in Ukrainian].

4. Emets O. A., Olkhovskaya E. V. (2013). Dokazatelstvo skhodymosty yteratsyonnoho metoda reshenyia zadachy kombynatornoi optymyzatsyy yhrovoho typa na razmeshchenyiakh [Proof of the convergence of the iterative method of solving the problem of combinatorial optimization of the game type on placements]. *Kibernetika i sistemnyj analiz – Cybernetics and systems analysis.* No. 1. P. 102-114 [in Russian].

5. O. V. Olkhovska (2014). Otsinka shvydkosti zbizhnosti iteratsiinoho metodu rozviazuvannia kombinatornykh optymizatsiinykh zadach ihrovoho typu [Evaluation of the speed of convergence of the iterative method of solving combinatorial optimization problems of the game type]. *Tavriiskyi visnyk informatyky i matematyky – Tavriyskyi Bulletin of Informatics and Mathematics*. No. 1. P. 31-42 [in Ukrainian].

6. Emets O. A., Olkhovsky D. M., Olkhovskaya E. V. (2014). Sravnenye metodov reshenyia yhrovыkh zadach: chyslovыe эksperymentы [Comparison of game problem solving methods: numerical experiments]. *Yskusstvennыi yntellekt - Artificial intelligence*. No. 1. P. 47-56 [in Russian].

7. Iemets, O.O., Chernenko, O.O., Chilikina, T.V., Olkhovska, O.V. (2021). Ohliad zadach kombinatornoi optymizatsii vyznachennia rentabelnosti silskohospodarskoho vyrobnytstva ta metody yikh rozviazuvannia. [Review of combinatorial optimization problems for determining the profitability of agricultural production and methods for solving them]. *Matematychne ta kompiuterne modeliuvannia. Seriia: Fizyko-matematychni nauky – Mathematical and computer modeling. Series: Physical and Mathematical Sciences*. 22, 63-74 [in Ukrainian].

8. Olkhovskyi D., Olkhovska O., Chernenko O., Parfyonova T., Chilikina T. (2022). Prohramnyi kompleks dlia rozviazuvannia evklidovykh kombinatornykh optymizatsiinykh zadach tochnymy ta nablyzhenymy metodamy [Software complex for solving Euclidean combinatorial optimization problems by exact and approximate methods]. *Informatsiini tekhnolohii ta suspilstvo – Information technologies and society*, 2 (4), P. 78-87 [in Ukrainian].

9. Yurii Oleksiichuk, Dmytro Olkhovskyi, Olena Olkhovska, Tetiana Chilikina, Oksana Chernenko, Oksana Orihivska (2022). Kombinatorna zadacha pro pobudovu mostiv ta metody yii rozviazannia [The combinatorial problem of building bridges and methods of its solution]. *Visnyk Kremenchutskoho natsionalnoho universytetu imeni Mykhaila Ostrohradskoho – Bulletin of Mykhailo Ostrogradsky National University of Kremenchug.* – Kremenchuk: KRNU. Issue 1(132). P. 115-122 [in Ukrainian].

10. Brown G.W. (1951). Iterative solution of games by fictitious play. *Activity analysis of production and allocation: Proceedings of a Conference*. New York. P. 374-376 [in English].

11. Kormen T., Leiserson C., Rivest R., Shtein K. (2005). Alhorytmы: postroenye y analyz [Algorithms: construction and analysis], 2nd ed. M. : Williams. 1296 p. [in Russian].

УДК 004.855.5
DOI https://doi.org/10.32689/maup.it.2022.3.6

Роман ШАПТАЛА
аспірант кафедри системного проектування, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», корпус 14, вул. Політехнічна 14-б, 03056, Київ, Україна (r.shaptala@gmail.com)

ORCID: 0000-0002-4367-5775

Геннадій КИСЕЛЬОВ
кандидат технічних наук, доцент, заступник завідувача кафедри системного проектування, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», корпус 14, вул. Політехнічна 14-б, 03056, Київ, Україна (g.kyselov@gmail.com)

ORCID: 0000-0003-2682-3593

Roman SHAPTALA
PhD student at the Department of System Design, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", build. 14, Politekhnichna 14-b str., Kyiv, Ukraine, postal code 03056 (r.shaptala@gmail.com)

Gennadiy KYSELOV
Candidate of technical sciences, Associate professor, Deputy head of the Department of System Design, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", build. 14, Politekhnichna 14-b str., Kyiv, Ukraine, postal code 03056 (g.kyselov@gmail.com)

Бібліографічний опис статті: Шаптала, Р., Кисельов, Г. (2022). Класифікація текстових документів з використанням доповнення векторних представлень документів графовими представленнями елементів словника синонімів. *Інформаційні технології та суспільство*, 3 (5), 49–55. DOI:

Bibliographic description of the article: Shaptala, R., Kyselov, G. (2022). Klasyfikaciya tekstovyh dokumentiv z vykorystannyam dopovnennya vektornyh predstavlen dokumentiv grafovymy predstavlennyamy elementiv slovnyka synonimiv [Document classification via augmentation of document embeddings with graph embeddings of synonyms dictionary]. *Informatsiini tekhnolohii ta suspilstvo – Information technology and society*, 3 (5), 49–55. DOI:

КЛАСИФІКАЦІЯ ТЕКСТОВИХ ДОКУМЕНТІВ З ВИКОРИСТАННЯМ ДОПОВНЕННЯ ВЕКТОРНИХ ПРЕДСТАВЛЕНЬ ДОКУМЕНТІВ ГРАФОВИМИ ПРЕДСТАВЛЕННЯМИ ЕЛЕМЕНТІВ СЛОВНИКА СИНОНІМІВ

*Стаття присвячена оцінці впливу методів доповнення графовими представленнями елементів словника синонімів векторних представлень документів на якість класифікації даних документів у малоресурсному середовищі. Дослідження таких середовищ є актуальним завданням, адже більшість мов світу, а також вузькоспеціалізовані прикладні області підпадають під даний критерій – даних для побудови та тренування сучасних потужних моделей машинного навчання не достатньо. **Метою роботи** є покращення якості класифікації документів у малоресурсному середовищі за допомогою доповнення їх інформацією зі словника синонімів через його кодування. Дослідження виконано через аналіз та використання сучасних напрацювань у області математичного моделювання, машинного навчання, обробки природних мов та науки про дані.*

***Наукова новизна роботи** полягає у тому, що пропонується векторна модель слів зі словника синонімів, яка на відміну від інших працює на основі представлень окремих вузлів графу словника, а отже може бути використана і в інших задачах обробки текстових даних. У цьому може допомогти трансферне навчання – підхід, що дозволяє комбінувати щільні векторні представлення у нейромережевих методах. При цьому вибір методу побудови векторних представлень словника синонімів напряму впливає на якість результатів, а також швидкодію та вимоги до апаратного забезпечення при їх використанні. Також у роботі представлено набір кроків передобробки та спосіб перетворення словника у граф для моделювання. Як **висновок**, у статті показано, що запропонований метод здатен збільшити F1-міру точності класифікації документів у малоресурсному середовищі на 2-3% на прикладі класифікації петицій до Київської міської ради за темами. Найвищий приріст якості було отримано за допомогою методу побудови векторних представлень графу Node2Vec, що працює на основі випадкових блукань, та не вимагає великої кількості ресурсів для навчання.*

***Ключові слова:** обробка природної мови, векторні представлення, класифікація документів, математична модель, машинне навчання, нейронні мережі, малоресурсне середовище.*

DOCUMENT CLASSIFICATION VIA AUGMENTATION OF DOCUMENT EMBEDDINGS WITH GRAPH EMBEDDINGS OF SYNONYMS DICTIONARY

The article is devoted to the assessment of the influence of the methods of augmentation of vector representations of documents with graph representations of the elements of the synonym dictionary on the quality of the classification of these documents in a low-resource environment. The study of such environments is an important task, because most of the world's languages, as well as highly specialized application areas, fall under this criterion – there is not enough data for building and training modern powerful machine learning models. The ***main goal*** *of this article is to improve the quality of document classification in a low-resource environment by augmenting them with information from the dictionary of synonyms through the encoding of the latter. The research was carried out through the analysis and use of modern developments in the field of mathematical modeling, machine learning, natural language processing and data science.*

The ***scientific novelty*** *of the work lies in the fact that a vector model of words from the dictionary of synonyms is proposed, which, unlike others, works on the basis of representations of individual nodes of the dictionary graph, and therefore can be used in other text data processing tasks. This can be helped by transfer learning, an approach that allows combining dense vector representations in neural network methods. At the same time, the choice of the method of building vector representations of the dictionary of synonyms directly affects the quality of the results, as well as the speed and requirements for hardware when using them. Also, the work presents a set of preprocessing steps and a method of converting a dictionary into a graph for modeling. As a* ***conclusion****, the article shows that the proposed model is able to increase the F1-score of document classification in a low-resource environment by 2-3% using the example of the classification of petitions to the Kyiv City Council by topic. The highest quality gain was obtained using the Node2Vec method of constructing graph vector representations, which works on the basis of random walks and does not require a large amount of training resources.*

Key words: *natural language processing, vector embeddings, document classification, mathematical model, machine learning, neural networks, low-resource.*

Актуальність. Попри стрімкий розвиток технологій та методів штучного інтелекту у контексті обробки природних мов, більшість уваги зосереджена на обмеженій кількості високоресурсних мов та прикладних областей, таких як англійська мова чи сентимент аналіз текстів у соціальних мережах, де наявна велика кількість джерел даних для побудови статистичних моделей. У випадку, коли ж даних – обмаль, багато припущень, на яких базуються популярні підходи не виконуються, а отже у малоресурсних середовищах потрібно шукати нові методи для успішного вирішення завдань. Для цього зазвичай вдаються до розробки підходів, які б могли використати сторонню інформацію аби отримати з неї певне додаткове «розуміння» контексту та середовища для вирішення оригінального завдання. У статті пропонується використати словник синонімів для того, щоб збільшити якість класифікації документів при обробці природних мов у малоресурсному середовищі. Тож **метою дослідження** є перевірка гіпотези, що кодування словника синонімів за допомогою векторних представлень здатне покращити результати роботи інших моделей машинного навчання.

Аналіз останніх досліджень та публікацій. Методи обробки природних мов у малоресурсному середовищі можна поділити на кілька груп [1]: створення додаткових розмічених даних та трансферне навчання. Перша група включає в себе підгрупи: доповнення даних [2], міжмовні проекції [3] та віддалений нагляд [4]. Друга група ділиться на: методи векторних представлень [5], багатомовні моделі мов [6] та адаптацію домену моделі мови [7]. Попри доведену ефективність даних підходів у певних ситуаціях, кожна з підгруп спирається на припущення існування даних або моделей, які мають певну інформацію про середовище, наприклад при використанні методу міжмовних проекцій очікується наявність паралельних корпусів між малоресурсною та багаторесурсними мовами. Наше дослідження пропонує використати словник синонімів, який наявний у багатьох мовах як джерело додаткової інформації для моделювання текстових даних.

Виклад основного матеріалу дослідження. Яким чином використати інформацію зі словника синонімів? Дослідники [8] пропонують робити доповнення даних за допомогою генерації нових екземплярів у тренувальній вибірці, у яких замінені слова-синоніми між собою. Такий підхід має ряд недоліків: розподіли слів у вибірці перестають бути репрезентативними, додається шум у набір для тренування, вибір кількості додатково згенерованих даних напряму впливає на результат, а також даний підхід важко перевикористати для інших завдань – правила заміни слів пишуться залежно від прикладної області. Для вирішення даних проблем пропонується моделювати словник синонімів як граф, де окремі вузли – це слова, а ребра між ними позначають синонімію. Після цього за допомогою методу побудови векторних представлень вузлів графу кожному слову знайти відповідний вектор. При цьому, слова, що знаходяться у графі близько повинні мати близькі за косинусною відстанню представлення. Запропоновану модель на високому рівні схематично зображено на рис. 1.

Як обрати метод побудови векторних представлень вузлів? За принципом дії їх поділяють на декілька груп [9]: на основі факторизації, на основі випадкових блукань та на основі глибокого навчання. Так, методи на основі факторизації розкладають матрицю, що репрезентує граф, на компоненти з бажани-

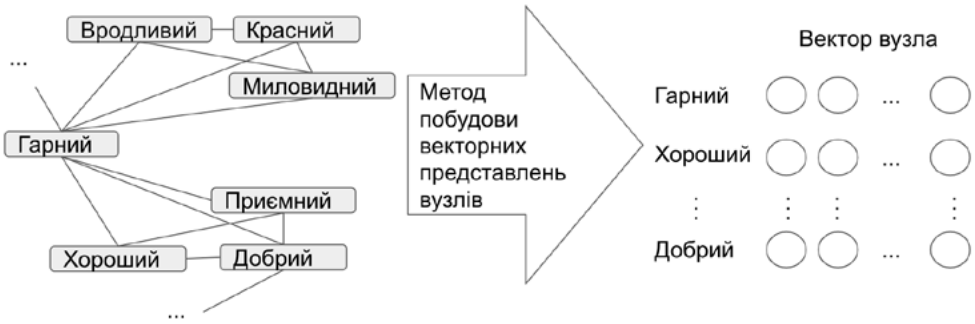


Рис. 1. Схема моделі словника синонімів

ми властивостями. Такою матрицею зазвичай виступає матриця суміжності вузлів, матриця Лапласа чи матриця ймовірностей переходу з одного вузла графу в інший. Прикладами даних методів є факторизація графу [10], HOPE [11], LLE [12], GraRep [13], Лапласівські проекції [14]. Перевагою цієї групи методів є простота та маленька кількість додаткових гіперпараметрів, але є і суттєві недоліки – для великих графів відповідні матриці вимагають забагато пам'яті – як мінімум квадратичну функцію від кількості вузлів, а отже обмежені у можливих застосуваннях. Метод факторизації графу працює має часову складність пропорційну кількості ребер і розкладає матрицю суміжності графу на множники. На відміну від нього, HOPE намагається явно закодувати подібності вищих порядків через мінімізацію схожості між елементами графу. У свою чергу, метод LLE робить припущення, що кожне представлення вузла є лінійною комбінацією представлень його сусідів. Унікальність методу GraRep полягає у використанні матриці ймовірностей переходів між вузлами та тому, що його часова складність не залежить від кількості ребер у графі, але кубічна пропорційно кількості вузлів. Метод Лапласівських проекцій розкладає Лапласіаном графа на власні вектори, отримуючи таким чином векторні представлення вузлів графа.

Методи на основі випадкових блукань вирішують проблему з пам'яттю наявну у попередньої групи методів, адже є ітеративними та працюють з графом напряму, а не з його матричним представленням. Такі методи працюють у два етапи – генерація випадкових блукань, коли створюються послідовності вузлів при випадковому обході графу, та навчання представлень, коли на основі послідовностей налаштовуються вектори для моделювання зв'язків різних порядків. Недоліком є велика кількість гіперпараметрів, що потрібно перебирати, таких як параметри випадкового блукання, кількість згенерованих послідовностей, кількість ітерацій при тренуванні. Популярні реалізації таких методів: Node2Vec [15], Walklets [16] та DeepWalk [17]. DeepWalk є першим онлайн алгоритмом навчання векторних представлень на графах, а тому легко масштабується. Саме його автори вперше почали генерувати випадкові блукання, а потім кодувати векторні представлення так, щоб з їх допомогою можна було прогнозувати наступний вузол у послідовності. Node2Vec узагальнив цей процес додавши можливість керувати генерацією випадкових блукань, а саме фокусом на обхід графу в ширину або в глибину. За це відповідають гіперпараметри p та q відповідно. Таким чином можна сказати, що DeepWalk є частковим випадком застосування Node2Vec при p=q=1. Метод Walklets додає своєрідну особливість до даного ітеративного процесу – він дозволяє при генерації пропускати деяких сусідів, що може спонукати модель краще відтворювати зв'язки вищих порядків.

Останнім часом популярності набула третя група методів, яка працює на основі глибокого навчання. Основна ідея таких методів – за допомогою глибоких автокодерів зменшити розмірність матричних представлень графів. При цьому конкретні реалізації даних методів відрізняються типами автокодерів та функціями втрат, що оптимізуються під час навчання. Такі методи вимагають менше пам'яті ніж перша група і при цьому здатні працювати ітеративно як друга група, але є повільнішими, тому що кількість параметрів при оптимізації значно вища. До даної групи відносяться підходи SDNE [18], DNGR [19] та GCN [20]. Підхід SDNE намагається одночасно оптимізувати збереження близькостей між предствленнями вузлів першого та другого порядку за допомогою глибоких автокодерів. У свою чергу DNGR використовує шумопоглинаючі автокодери для розкладу матриці ймовірностей переходів побудованої на основі тих же випадкових блукань. На жаль, обидва алгоритма можуть бути обчислювально складними, адже на кожному кроці приймають на вхід увесь граф. Цю проблему вирішують GCN – графові згорткові мережі, що визначають операцію згортки на графі, таким чином агрегуючи локальну структуру графу на нижніх шарах, та глобальну на верхніх.

Для перевірки впливу доповнення даних інформацією зі словника синонімів закодованого методами побудови графових представлень було розроблено наступний експеримент. Набір даних – петиції до

Київської міської ради (представлений у [21]), завдання – класифікація документів за темами, набір поділений на вибірки для тренування (75%) та тестування (25%). Малоресурсність середовища забезпечена двома аспектами набору даних: документи у ньому написані українською мовою, вузькоспеціальною прикладною областю урбаністики. Базова модель без доповнення даних – усереднені закодовані Word2Vec підходом вектори слів петицій як ознаки, що подаються на вхід багатошаровому перцептрону як класифікатору. В усіх експериментах кожна частина загального процесу включає підбір гіперпараметрів за допомогою сіткового пошуку. Таким чином випадковість ініціалізації не впливатиме на загальний результат. Метрика якості класифікації – зважена F1-міра, обрана через незбалансованість набору даних за класами, що є частим ускладненням при роботі у малоресурсному середовищі. Точність у такому випадку показувала б надто оптимістичні результати, які не виражали б істинну якість моделі.

Для побудови графу було викачано україномовний словник синонімів з Офіційного сайту української мови [22]. Але перед його кодуванням важливо провести передобробку цих даних. Вона включає такі кроки: видалення зайвої для експерименту інформації такої як приклади вжитку та скорочень; дедублікація пар синонімів, адже пропонується будувати ненапрямлений граф, а у словнику є зв'язки між словами у обидві сторони; так як власні назви у словнику не зустрічаються, усі слова приведені до нижнього регістру для легшого пошуку відповідників у текстах петицій. Таким чином на вхід процесу побудови графу надається список чистих пар «синонім-синонім». При побудові, усі ці пари з'єднуються між собою ненапрямленими ребрами, утворюючи кластери синонімічних зв'язків. Результуючий граф має 44664 вузли та 391047 ребер, а отже є достатньо великим для врахування обмежень за пам'яттю та швидкодією при виборі методів побудови векторних представлень вузлів. Середня степінь вершини у графі складає 8.755, що свідчить про високу зв'язність графу, а також, що його локальна структура відіграватиме важливу роль при моделюванні.

Яким чином поєднати вектори петицій та вектори слів зі словника синонімів? Для цього можна скористатись простими методами з області мультимодального навчання, а саме методами конкатенації та зваженої суми векторних представлень. У першому випадку розмірність результуючого вектора, який піде на вхід моделі класифікації, являється сумою розмірностей вектора петиції та вектора вузла графу словника синонімів, тоді ж як у другому випадку накладається обмеження, що вищевказані вектори повинні бути однієї розмірності, відповідно при виконанні операції зваженої суми, результуючий вектор теж матиме такий розмір. Більш складні методи злиття векторних представлень такі як методи на основі механізму уваги вимагають підбір додаткового набору параметрів, що ускладнює оптимізацію, а також вимагає більше даних для навчання. У малоресурсних середовищах застосування таких методів не практичне, тому у дослідженні перевіряються лише методи злиття на основі простих операцій.

Подібно процедурі усереднення векторів слів Word2Vec, яка є базовою моделлю у дослідженні, варто визначити процедуру агрегації окремих векторів слів зі словника синонімів. Ми пропонуємо також усереднювати дані вектори, ігноруючи ті слова з документу, що не існують у словнику. Саме на цьому етапі важливо максимізувати кількість слів, що перетинаються між множиною слів з документів та множиною слів зі словника. Тому передобробка обох джерел даних має суттєвий вплив на результат. Також, враховуючи, що слова у словниках зазвичай подаються у називному відмінку або інфінітиві, а у документах залежно від контексту можуть зустрічатись у інших формах, процедура їх співставлення не очевидна. Ідеальною процедурою у даному випадку було б використання двох додаткових моделей – моделі визначення частини мови та моделі побудови словоформ. Малоресурсність середовища гарантує, що таких моделей високої якості не існує, а даних для їх розробки обмаль. Тому для процедури співставлення пропонується використати алгоритми пошуку найближчого слова за міжрядковою відстанню. Це додає шум в фінальні представлення, які йдуть на вхід класифікатору, але забезпечує вище покриття слів при співставленні документів та словника синонімів.

Експериментальні результати. Для підтвердження гіпотези, що злиття векторних репрезентацій документів та елементів словника синонімів було проведено набір експериментів у різних комбінаціях способів реалізації запропонованого методу. Таким чином порівнювались базова модель з методами її покращення за допомогою інформації зі словника синонімів: доповнення даних через заміну синонімів та злиття з векторним представленням графа синонімів. При цьому останній залежить від вибору методу векторного представлення словника синонімів (побудови графового представлення), а також методу злиття векторних представлень (конкатенація чи зважена сума). Серед методів векторного представлення словника синонімів було порівняно методи факторизації графу, HOPE, LLE, Лапласівські проекції, GraRep, Node2Vec, Walklets та GCN. Через розмір графу, а також пропорційну кількості ребер часову складність алгоритмів HOPE, LLE та Лапласівські проекції виявились не практичними у застосуванні в даному середовищі. Кубічна складність у кількості вузлів алгоритму GraRep теж завадила йому завершитись успішно, тому ці алгоритми не рекомендуються у застосуванні при роботі зі

словником синонімів. Так як застосування методу Node2Vec з гіперпраметрами q та p рівними одиниці рівнозначне застосуванню методу DeepWalk, а саме дані значення виявились оптимальними, у результатах зазначається лише перший. Серед методів на основі глибокого навчання для експериментів було обрано GCN, адже він є узагальненням SDNE та DNGR, а отже покриває їх результати при правильному підборі гіперпараметрів. Значення результуючих метрик для порівняння наводиться у таблиці 1.

Таблиця 1

Порівняння F1-міри запропонованих підходів

Метод обробки природної мови у малоресурсному середовищі	**Метод векторного представлення словника синонімів**	**Метод злиття векторних представлень**	**Зважена F1-міра**
Базова модель	-	-	0.629
Доповнення даних через заміну синонімів	-	-	0.631
Злиття з векторним представленням графа синонімів	Факторизація графу	Конкатенація	0.639
Злиття з векторним представленням графа синонімів	Факторизація графу	Зважена сума	0.650
Злиття з векторним представленням графа синонімів	Node2Vec	Конкатенація	0.640
Злиття з векторним представленням графа синонімів	Node2Vec	Зважена сума	**0.659**
Злиття з векторним представленням графа синонімів	Walklets	Конкатенація	0.638
Злиття з векторним представленням графа синонімів	Walklets	Зважена сума	0.657
Злиття з векторним представленням графа синонімів	GCN	Конкатенація	0.645
Злиття з векторним представленням графа синонімів	GCN	Зважена сума	0.657

Як бачимо, найвищий приріст метрики – 3% порівняно з базовою моделлю надає злиття методом зваженої суми з векторним представленням графа синонімів побудованим за алгоритмом Node2Vec. Для перевірки статистичної значущості результату було проведено Хі-квадрат тест за методом МакНемара [23], а саме було сформульовано гіпотезу, що результати класифікації базовим методом та запропонованим методом – рівні. Вхідні дані для проведення тесту відображені у таблиці 2. Рівень статистичної значущості приймаємо як 0.05. Фінальне значення статистики вищезазначеного тесту рівне 4.056, що відповідає p-значенню у 0.044, що менше прийнятого рівня статистичної значущості 0.05, а отже початкова гіпотеза про рівність результатів класифікації базовим та запропонованим методами відхиляється.

Таблиця 2

Таблиця невідповідностей для Хі-квадрат тесту за методом МакНемара

		Класифікація запропонованим методом	
		правильна	**помилкова**
Класифікація базовим методом	правильна	757	59
	помилкова	83	350

За допомогою трансферного навчання отримані векторні представлення словника синонімів можна перевикористати для покращення якості вирішення інших завдань. Для цього варто лише пересвідчитись, що кроки передобробки, описані раніше, не прибирають корисної інформації для нового завдання, а отже можна через обраний метод злиття додавати існуючі вектори до іншої класифікаційної архітектури. Якщо ж описана передобробка для нового завдання не оптимальна, варто змінити дані кроки, застосувати нову процедуру на словнику синонімів та перетренувати метод побудови векторних представлень графу на оновлених даних.

Висновки. У статті запропоновано модель словника синонімів української мови, а також метод, який поєднує класичний підхід до класифікації документів з векторними представленнями словника синонімів, що дозволяє збільшити зважену F1-міру класифікації документів на 3% при обробці природної мови у малоресурсних середовищах. Значущість отриманих результатів була підтверджена за допомогою Хі-квадрат тесту за методом МакНемара з рівнем статистичної значущості 0.05. Серед методів побудови векторних представлень графу в даному середовищі найкращі результати показав Node2Vec, а серед методів злиття векторних представлень рекомендується обрати метод зваженої суми. Роботу можна розвивати для дослідження та моделювання інших типів словників за допомогою графових векторних представлень, створення та модифікації методів побудови графових векторних представлень, які були б оптимізовані під структуру графів на основі словників, а також перевірку ефективності дії методу в інших малоресурсних середовищах.

Список використаних джерел:

1. Hedderich, M. A., Lange, L., Adel, H., Strötgen, J., & Klakow, D. (2021). A Survey on Recent Approaches for Natural Language Processing in Low-Resource Scenarios. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 2545–2568).

2. Perez, L., & Wang, J. (2017). The Effectiveness of Data Augmentation in Image Classification using Deep Learning. Retrieved from https://arxiv.org/abs/1712.04621.

3. Eskander, R., Muresan, S., & Collins, M. (2020). Unsupervised Cross-Lingual Part-of-Speech Tagging for Truly Low-Resource Scenarios. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 4820–4831). Association for Computational Linguistics (ACL). https://doi.org/10.18653/V1/2020.EMNLP-MAIN.391.

4. Mintz, M., Bills, S., Snow, R., & Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP* (pp. 1003–1011). Association for Computational Linguistics. Retrieved from https://aclanthology.org/P09-1113.

5. Collobert, R., Weston, J., Com, J., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research*, *12*, 2493–2537. https://doi.org/10.5555/1953048.2078186.

6. Hu, J., Ruder, S., Siddhant, A., Neubig, G., Firat, O., & Johnson, M. (2020). XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalisation. In *International Conference on Machine Learning. PMLR*.

7. Gururangan, S., Marasovi´cmarasovi´c, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D. (2020). Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 8342–8360). Association for Computational Linguistics. Retrieved from https://github.com/allenai.

8. Wei, J., & Zou, K. (2019). EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (pp. 6382–6388). Retrieved from http://github.

9. Goyal, P., & Ferrara, E. (2018). Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, *151*, 78–94. https://doi.org/10.1016/J.KNOSYS.2018.03.022

10. Ahmed, A., Shervashidze, N., Narayanamurthy, S., Josifovski, V., & Smola, A. J. (2013). Distributed Large-Scale Natural Graph Factorization. In *Proceedings of the 22nd International Conference on World Wide Web* (pp. 37–48). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/2488388.2488393.

11. Ou, M., Cui, P., Pei, J., Zhang, Z., & Zhu, W. (2016). Asymmetric Transitivity Preserving Graph Embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1105–1114). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/2939672.2939751.

12. T., R. S., & K., S. L. (2000). Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, *290*(5500), 2323–2326. https://doi.org/10.1126/science.290.5500.2323.

13. Cao, S., Lu, W., & Xu, Q. (2015). GraRep: Learning Graph Representations with Global Structural Information. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management* (pp. 891–900). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/2806416.2806512.

14. Belkin, M., & Niyogi, P. (2003). Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, *15*(6), 1373–1396. https://doi.org/10.1162/089976603321780317.

15. Grover, A., & Leskovec, J. (2016). Node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 855–864). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/2939672.2939754.

16. Perozzi, B., Kulkarni, V., Chen, H., & Skiena, S. (2017). Don't Walk, Skip! Online Learning of Multi-Scale Network Embeddings. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017* (pp. 258–265). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3110025.3110086.

17. Perozzi, B., Al-Rfou, R., & Skiena, S. (2014). DeepWalk: Online learning of social representations. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 701–710). https://doi.org/10.1145/2623330.2623732.

18. Wang, D., Cui, P., & Zhu, W. (2016). Structural Deep Network Embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1225–1234). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/2939672.2939753.

19. Cao, S., Lu, W., & Xu, Q. (2016). Deep Neural Networks for Learning Graph Representations. *Proceedings of the AAAI Conference on Artificial Intelligence*, *30*(1 SE-Technical Papers: Machine Learning Applications). Retrieved from https://ojs.aaai.org/index.php/AAAI/article/view/10179.

20. Kipf, T. N., & Welling, M. (2016). Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017 – Conference Track Proceedings*. International Conference on Learning Representations, ICLR. Retrieved from https://arxiv.org/abs/1609.02907v4.

21. Samvelyan, A., Shaptala, R., & Kyselov, G. (2020). Exploratory data analysis of Kyiv city petitions. In *2020 IEEE 2nd International Conference on System Analysis Intelligent Computing (SAIC)* (pp. 1–4). https://doi.org/10.1109/SAIC51296.2020.9239185.

22. Офіційний сайт Української мови. URL: https://ukrainskamova.com.

23. McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, *12*(2), 153–157. https://doi.org/10.1007/BF02295996.

References:

1. Hedderich, M. A., Lange, L., Adel, H., Strötgen, J., & Klakow, D. (2021). A Survey on Recent Approaches for Natural Language Processing in Low-Resource Scenarios. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 2545–2568).

2. Perez, L., & Wang, J. (2017). The Effectiveness of Data Augmentation in Image Classification using Deep Learning. Retrieved from https://arxiv.org/abs/1712.04621.

3. Eskander, R., Muresan, S., & Collins, M. (2020). Unsupervised Cross-Lingual Part-of-Speech Tagging for Truly Low-Resource Scenarios. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 4820–4831). Association for Computational Linguistics (ACL). https://doi.org/10.18653/V1/2020.EMNLP-MAIN.391.

4. Mintz, M., Bills, S., Snow, R., & Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP* (pp. 1003–1011). Association for Computational Linguistics. Retrieved from https://aclanthology.org/P09-1113.

5. Collobert, R., Weston, J., Com, J., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research*, 12, 2493–2537. https://doi.org/10.5555/1953048.2078186.

6. Hu, J., Ruder, S., Siddhant, A., Neubig, G., Firat, O., & Johnson, M. (2020). XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalisation. In *International Conference on Machine Learning. PMLR*.

7. Gururangan, S., Marasovi´cmarasovi´c, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D. (2020). Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 8342–8360). Association for Computational Linguistics. Retrieved from https://github.com/allenai.

8. Wei, J., & Zou, K. (2019). EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (pp. 6382–6388). Retrieved from http://github.

9. Goyal, P., & Ferrara, E. (2018). Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151, 78–94. https://doi.org/10.1016/J.KNOSYS.2018.03.022.

10. Ahmed, A., Shervashidze, N., Narayanamurthy, S., Josifovski, V., & Smola, A. J. (2013). Distributed Large-Scale Natural Graph Factorization. In *Proceedings of the 22nd International Conference on World Wide Web* (pp. 37–48). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/2488388.2488393.

11. Ou, M., Cui, P., Pei, J., Zhang, Z., & Zhu, W. (2016). Asymmetric Transitivity Preserving Graph Embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1105–1114). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/2939672.2939751.

12. T., R. S., & K., S. L. (2000). Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500), 2323–2326. https://doi.org/10.1126/science.290.5500.2323.

13. Cao, S., Lu, W., & Xu, Q. (2015). GraRep: Learning Graph Representations with Global Structural Information. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management* (pp. 891–900). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/2806416.2806512.

14. Belkin, M., & Niyogi, P. (2003). Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, *15*(6), 1373–1396. https://doi.org/10.1162/089976603321780317.

15. Grover, A., & Leskovec, J. (2016). Node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 855–864). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/2939672.2939754.

16. Perozzi, B., Kulkarni, V., Chen, H., & Skiena, S. (2017). Don't Walk, Skip! Online Learning of Multi-Scale Network Embeddings. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017* (pp. 258–265). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3110025.3110086.

17. Perozzi, B., Al-Rfou, R., & Skiena, S. (2014). DeepWalk: Online learning of social representations. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 701–710). https://doi.org/10.1145/2623330.2623732.

18. Wang, D., Cui, P., & Zhu, W. (2016). Structural Deep Network Embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1225–1234). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/2939672.2939753.

19. Cao, S., Lu, W., & Xu, Q. (2016). Deep Neural Networks for Learning Graph Representations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30 (1 SE-Technical Papers: Machine Learning Applications). Retrieved from https://ojs.aaai.org/index.php/AAAI/article/view/10179.

20. Kipf, T. N., & Welling, M. (2016). Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017 – Conference Track Proceedings*. International Conference on Learning Representations, ICLR. Retrieved from https://arxiv.org/abs/1609.02907v4.

21. Samvelyan, A., Shaptala, R., & Kyselov, G. (2020). Exploratory data analysis of Kyiv city petitions. In *2020 IEEE 2nd International Conference on System Analysis Intelligent Computing (SAIC)* (pp. 1–4). https://doi.org/10.1109/SAIC51296.2020.9239185.

22. Oficiinyi sait Ukrainskoi movy [Official website of Ukrainian language]. Retrieved from: https://ukrainskamova.com [in Ukrainian].

23. McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, *12*(2), 153–157. https://doi.org/10.1007/BF02295996.

НАУКОВЕ ВИДАННЯ

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СУСПІЛЬСТВО

INFORMATION TECHNOLOGY AND SOCIETY

ВИПУСК 3 (5)
ISSUE 3 (5)

2022

Коректура
Ірина Чудеснова

Комп'ютерна верстка
Світлана Калабухова

Формат 60x84/8. Гарнітура Cambria.
Папір офсет. Цифровий друк. Ум. друк. арк. 13,72. Замов. № 0123/027. Наклад 300 прим.

Видавництво і друкарня – Видавничий дім «Гельветика»
65101, Україна, м. Одеса, вул. Інглезі, 6/1
Телефон +38 (095) 934 48 28, +38 (097) 723 06 08
E-mail: mailbox@helvetica.ua
Свідоцтво суб'єкта видавничої справи
ДК No 7623 від 22.06.2022 р.