НАЦІОНАЛЬНА
АКАДЕМІЯ НАУК УКРАЇНИ
ІНСТИТУТ ПРОГРАМНИХ СИСТЕМ

ПРОБЛЕМИ ПРОГРАМУВАННЯ

науковий журнал

№ 1 ***січень - березень*** ***2025***

Заснований у березні 1999 р.

ЗМІСТ

Свідоцтво про державну реєстрацію КВ № 7490 від 01.07.2003

Науковий журнал "Проблеми програмування" занесений до переліку наукових фахових видань України, в яких можуть публікуватися основні результати дисертаційних робіт.

ISSN 1727-4907

NATIONAL ACADEMY
OF SCIENCES OF UKRAINE
INSTITUTE OF SOFTWARE SYSTEMS

PROBLEMS OF PROGRAMMING

scientific journal

№ 1 ***January – March*** ***2025***

Founded in March, 1999

CONTENT

УДК 004.942:[623.465+656.052.4] https://doi.org/10.15407/pp2025.01.003

І.О. Романенко, А.Л. Яловець

АНАЛІТИЧНИЙ ОГЛЯД СТАНУ ДОСЛІДЖЕНЬ З МОДЕЛЮВАННЯ ПРОЦЕСІВ ПЕРЕСЛІДУВАННЯ/УТІКАННЯ У ТРИВИМІРНОМУ ПРОСТОРІ

В статті наведено аналітичний огляд основних тенденцій, домінуючих у світі в рамках вирішення проблем моделювання процесів переслідування/утікання у тривимірному просторі. З метою отримання більш структурованого розгляду існуючого стану виявлено основні аспекти виконуваних досліджень та здійснено подальший аналіз таких аспектів. Проаналізовано основні підходи до моделювання процесу переслідування/утікання у тривимірному просторі та об’єкти, які при цьому розглядаються. Досліджено методи планування шляхів учасників процесу переслідування/утікання у тривимірному просторі. Розглянуто підходи, використані для формування стратегій переслідування/утікання у тривимірному просторі. За результатами аналізу обґрунтовано, що виконаний огляд дозволяє отримати загальне уявлення про основні світові тенденції, що склалися на сьогодні в дослідженнях процесів переслідування/утікання у тривимірному просторі.

Ключові слова: агент, гравець, тривимірний простір, безпілотні літальні апарати, методи планування шляхів, стратегії переслідування/утікання.

I.O. Romanenko, A.L. Yalovets

ANALITICAL REVIEW OF THE STATE OF RESEARCH ON MODELING THE PROCESSES OF PURSUIT/ESCAPE IN THREE-DIMENSIONAL SPACE

The article provides an analytical review of the main trends dominant in the world in solving the problems of modeling the processes of pursuit/escape in three-dimensional space. In order to obtain a more structured consideration of the current state, the main aspects of the research performed are identified and further analysis of such aspects is carried out. The main approaches to modeling the process of pursuit/escape in three-dimensional space and the objects that are considered are analyzed. The methods of planning the paths of participants in the process of pursuit/escape in three-dimensional space are studied. The approaches used to form pursue/escape strategies in three-dimensional space are considered. Based on the results of the analysis, it is substantiated that the review allows to get a general idea of the main global trends that have developed today in the study of the processes of pursuit/escape in three-dimensional space.

Keywords: agent, player, three-dimensional space, drones, path-planning techniques, pursuit/escape strategies.

Вступ

Задачі переслідування/утікання є відомими задачами, які мають велике теоретичне та прикладне значення, є достатньо вивченими та дослідженими, причому традиційно їх дослідження виконуються в рамках теорії диференціальних ігор (див., наприклад, [1-6]). Огляд сучасного стану таких досліджень для випадку переслідування/утікання на площині викладено, наприклад, в [7]. Водночас в останні десятиріччя задачі переслідування/утікання також активно досліджуються і за допомогою мультиагентного підходу (див., наприклад, [8-13]). Однак, як зауважується в [14-16], на сьогоднішній день відомі дослідження в основному стосуються задач переслідування/утікання *на площині*, і є лише незначна кількість відкритих публікацій, присвячених дослідженням задач переслідування/утікання *у тривимірному просторі*.

Метою даної статті є огляд стану досліджень з моделювання процесів переслідування/утікання у тривимірному просторі на основі аналізу відкритих джерел. З метою отримання більш структурованого розгляду існуючого стану, в статті здійсню-

ється виявлення основних аспектів виконуваних досліджень процесів переслідування/утікання у тривимірному просторі та провадиться подальший аналіз таких аспектів.

1. Основні аспекти досліджень процесів переслідування/утікання у тривимірному просторі

За результатами узагальнення аналізу відкритих джерел [14-24] виявлено низку аспектів досліджень процесів переслідування/утікання у тривимірному просторі, яким відповідають:

- Підходи, використані для моделювання процесу переслідування/утікання:
  - підхід, заснований на теорії диференціальних ігор [15, 18, 19, 21, 22, 24];
  - мультиагентний підхід [14, 16, 17, 20, 23].
- Об'єкти досліджень:
  - безпілотні літальні апарати [15, 17 – 22];
  - автономні підводні апарати [16, 22];
  - безпілотні наземні апарати [14, 15, 20];
  - персонажі комп'ютерних ігор [23];
  - гіперзвукові транспортні засоби [24].
- Склад учасників досліджуваного процесу переслідування/утікання:
  - один переслідувач та один утікач [15, 18, 21];
  - два переслідувача та один утікач [20, 24];
  - декілька переслідувачів та один утікач [14, 19, 23];
  - декілька переслідувачів та декілька утікачів [16, 22];
  - до 100 переслідувачів та 100 утікачів [17].
- Структура складу учасників досліджуваного процесу переслідування/утікання:
  - гомогенна [14, 16 – 19, 21, 22, 24];
  - гетерогенна [15, 20, 23].
- Задачі, розв'язувані в рамках моделювання процесу переслідування/утікання:
  - планування дій утікача [19] або переслідувачів [19, 23];
  - планування траєкторій руху переслідувачів [14, 22, 23];
  - прогнозування траєкторій руху утікача [19].
- Особливості врахування впливу навколишнього середовища на об'єкт досліджень:
  - врахування вітрових навантажень [19, 22];
  - врахування підводних потоків [22];
  - врахування рельєфу місцевості [14, 16, 20].
- Використані методи планування шляхів:
  - метод потенціальних полів [19, 20];
  - метод, заснований на побудові діаграми Вороного [14, 22];
  - метод, заснований на побудові графа видимості [14, 23].
- Підходи, використані для формування стратегій переслідування/утікання:
  - геометричний підхід, заснований на побудові кіл (сфер) Аполлонія [15, 19];
  - підхід, заснований на Марківських процесах прийняття рішень [17, 24];
  - підхід, заснований на використанні методів пошуку на графах [14, 23].

2. Змістовний аналіз виявлених аспектів досліджень

Аналіз виявлених аспектів дозволяє зробити два очевидних висновка. По-перше, розглянуті дослідження практично в однаковій мірі засновані на використанні обох підходів до моделювання процесів переслідування/утікання: як підходу, що ґрунтується на теорії диференціальних ігор, так і мультиагентного підходу. По-друге, у якості об'єктів досліджень в переважній більшості випадків виступають безпілотні літальні апарати (БПЛА) та автономні підводні апарати (АПА). Проаналізуємо ці висновки й те, що з них випливає.

2.1. Щодо підходів до моделювання процесу переслідування/утікання. Паритет у використанні обох підходів до моделювання свідчить про наявність однакової зацікавленості в отриманні як теоретичних, так і прикладних результатів досліджень проблеми переслідування/уті-

кання у тривимірному просторі. Однак, як можна побачити вище, вибір конкретного підходу впливає на аналізований склад учасників досліджуваного процесу переслідування/утікання: теоретико-ігровий підхід до моделювання дозволяє досліджувати лише обмежений склад учасників (максимально це декілька переслідувачів та утікачів), тоді як мультиагентний підхід – досить великий склад (до 100 переслідувачів та 100 утікачів). Така різниця пояснюється тим [8], що, якщо теоретико-ігровий підхід ґрунтується на наявності закону еволюції досліджуваної динамічної системи, заданої відповідними диференціальними рівняннями, які передбачають спрощений розгляд процесу переслідування/утікання та дозволяють описати лише досить прості ситуації, то мультиагентний підхід не потребує завдання закону еволюції динамічної системи (за неї в обох випадках виступає процес переслідування/утікання), а еволюція системи відбувається внаслідок дій та взаємодій агентів, які задаються певними правилами або алгоритмами. На їх основі формуються (з урахуванням явища емерджентності) достатньо складні ситуації переслідування/утікання.

Слід зауважити, що іноді в публікаціях спостерігаються випадки (див., наприклад, [22]), коли в дослідженнях використовується теоретико-ігровий підхід, але гравців називають агентами. Наголосимо, що це концептуальна помилка, яка вводить в оману читача й спотворює зміст публікації через підміну понять та свідчить, принаймні, про нездатність авторів розрізнити зазначені підходи.

Далі, не порушуючи загальності викладення, з метою запобігання таких непорозумінь, розглядаючи умовного учасника процесу переслідування/утікання будемо іменувати його як «агент (гравець)», розуміючи під цим щоразу лише одну з цих назв, яка відповідатиме сутності використовуваного підходу для моделювання. Тобто, якщо в дослідженнях використано підхід, заснований на теорії диференціальних ігор, то змістовно під терміном «агент (гравець)» будемо розуміти гравця; якщо ж мультиагентний підхід – то агента.

2.2. Щодо об'єктів досліджень. Розгляд таких об'єктів досліджень як БПЛА та АПА потребують врахування впливу навколишнього середовища на них. Так, в [22] зазначається, що «причина, через яку ми розглядаємо динамічні умови навколишнього середовища, полягає в тому, що наявність змінних в часі або просторово мінливих потоків може істотно вплинути на рух транспортних засобів і відповідну стратегію. Це стосується, зокрема, випадків, коли переслідувачами або утікачами (або обома) є невеликі автономні підводні апарати (АПА) або невеликі безпілотні літальні апарати (БПЛА)». В [19] підкреслюється, що для БПЛА проблема моделювання процесів переслідування/утікання ускладнюється наявністю значних вітрових навантажень, які впливають на траєкторію та стратегії руху агентів (гравців). Крім того, як зауважується в [20], одночасно необхідно враховувати рельєф місцевості.

В аналізованих публікаціях врахування навантажень на об'єкт досліджень з боку навколишнього середовища, зокрема, забезпечується як за допомогою диференціальних рівнянь [22], так і з використанням інтелектуальних методів управління [19]. Наприклад, в [19] інтелектуальні методи управління базуються на стратегіях, що реалізуються за допомогою бази знань, в якій подано множину правил управління, що враховують задані обмеження управління та існуючі збурення навколишнього середовища.

Крім того, вид об'єкта досліджень впливає на вибір методів планування шляхів агентів (гравців), в межах яких також вирішуються проблеми обходу перешкод навколишнього середовища. Наприклад, якщо розглядати як об'єкт досліджень БПЛА або АПА, то в аналізованих дослідженнях [19, 20, 22] такими методами виступають метод потенціальних полів та метод, заснований на побудові діаграми Вороного. В інших випадках [14, 23] використано метод, заснований на побудові графа видимості. Ці методи розглянуто в п. 2.4.

Окремо слід згадати про безпілотні наземні апарати (БНА), які зазвичай не розглядаються як об'єкти досліджень процесів переслідування/утікання у тривимір-

ному просторі. Однак в аналізованих дослідженнях [14, 15, 20] БНА виступають такими об'єктами досліджень або в складі гетерогенної команди агентів (гравців) [15, 20], в якій БНА діє на площині та виступає як утікач [15, 20], так і переслідувач [20], а БПЛА діють у тривимірному просторі та виступають переслідувачами, або в складі гомогенної команди агентів (гравців) [14], що діють у навколишньому середовищі 2.5D (тобто в 2D-середовищі, доповненому картою висот).

2.3. Щодо складу учасників досліджуваного процесу переслідування/утікання. Спостережувані тенденції в аналізованих дослідженнях такі, що понад дві третини розглянутих складів учасників містять лише одного утікача. Водночас, переважна більшість таких випадків досліджувалася з точки зору теорії диференціальних ігор. Це свідчить про те, що сучасні дослідження переслідування/утікання у тривимірному просторі здебільшого орієнтовані на розгляд найпростіших ситуацій.

З іншого боку, у цих дослідженнях розглядались як гомогенні, так і гетерогенні структури складу учасників процесу переслідування/утікання. Наприклад, в [18, 19, 21] гомогенні структури включали тільки БПЛА. В свою чергу, в [15] гетерогенна структура учасників переслідування/утікання включала БПЛА як переслідувача та БНА як утікача, в [20] – БПЛА та БНА як переслідувачів та БНА як утікача, а в [23] – персонажів комп'ютерної гри як переслідувачів та персонажа, керованого людиною-гравцем, як утікача.

2.4. Щодо методів планування шляхів агентів (гравців). Зауважимо, що для вирішення проблем планування шляхів навколишнє середовище необхідно перетворити на певне дискретне подання, придатне для виконання планування шляхів за допомогою відомих методів. Для цього використовуються такі підходи [25]:

– *Планування на основі методу потенціальних полів*. Цей підхід заснований на створенні градієнту в навколишньому середовищі агента (гравця), що направляє його у вільному просторі навколишнього середовища до цільової позиції з множини його попередніх позицій.

– *Побудова графу пошуку*. Цей підхід передбачає побудову графу зв'язності у вільному просторі навколишнього середовища агента (гравця), на якому і здійснюється пошук шляхів.

Планування на основі методу потенціальних полів засновано на ідеї створення штучного потенціального поля в навколишньому середовищі агента (гравця), де він розглядається як точка, на яку це поле впливає. Ціль (глобальний мінімум у цьому полі) діє на агента (гравця) як сила тяжіння, а перешкоди – як сили відштовхування. Суперпозиція всіх сил плавно направляє агента (гравця) до цілі, одночасно уникаючи його зіткнень з відомими перешкодами. Треба зауважити, що використання підходу потенціальних полів дозволяє забезпечити більше, ніж планування шляху. Таке поле утворює закон керування рухом агента (гравця): якщо агент (гравець) може локалізувати своє місцезнаходження відносно навколишнього середовища та потенціального поля, то він завжди може визначити свій наступний потрібний крок, заснований на полі. Основним недоліком підходу на основі потенціальних полів є те, що у полі можуть існувати локальні мінімуми, які можуть дезорієнтувати агента (гравця) у пошуку цілі. Зауважимо, що в аналізованих дослідженнях метод потенціальних полів використовувався з урахуванням певних обмежень навколишнього середовища або поєднувався з іншими методами. Наприклад, в [20] цей метод використовувався для планування дій переслідувачів лише у разі, коли утікачі знаходяться в полі зору (тобто в межах прямої видимості) переслідувачів. У свою чергу, в [19] метод потенціальних полів, використаний для вирішення проблеми уникнення перешкод, поєднано з геометричним підходом, заснованим на побудові сфер Аполлонія, що загалом забезпечило формування раціональних стратегій руху агентів (гравців).

Будуючи граф пошуку, агент (гравець) з'єднує початкову та цільову позиції у вільному просторі навколишнього середовища множиною ліній, які утворюють граф зв'язності як множину шляхів, що існують між початковою та цільовою позиціями. Для створення графу пошуку найбільше поши-

рення набули методи, засновані на побудові *діаграми Вороного* та *графу видимості*.

Метод, заснований на побудові діаграми Вороного. Під час побудови діаграми Вороного знаходиться множина точок, кожна з яких є центром кругу, вписаного у вільний простір навколишнього середовища, який торкається як мінімум двох точок контурів перешкод. У процесі з'єднання послідовностей таких точок утворюються ребра, які складаються з прямих та параболічних сегментів, а вершини графа відповідають точкам зустрічі різних ребер та початковій і цільовій вершинам. Водночас забезпечується максимізація відстані між агентом (гравцем) та перешкодами в навколишньому середовищі. Створюючи діаграму Вороного, агент (гравець) з'єднує початкову та цільову позиції у вільному просторі навколишнього середовища множиною ліній, що утворюють граф зв'язності як множину шляхів, які існують між початковою та цільовою позиціями. Однак найкоротший шлях, знайдений на діаграмі Вороного, не є оптимальним за критерієм довжини. В аналізованих дослідженнях діаграма Вороного використовується для розв'язання різних задач. Наприклад, в [14] вона використовується для виявлення вузьких ділянок навколишнього середовища агента (гравця) з метою виключення таких ділянок з процесу пошуку шляху на графах. А в [22] за допомогою діаграми Вороного здійснюється динамічне призначення доцільних переслідувачів для захоплення відповідних утікачів.

Метод, заснований на побудові графа видимості, де граф видимості складається з множини ребер, що зв'язують усі пари вершин, які є видимими (тобто ребра проходять через вільний простір навколишнього середовища) між собою (включаючи початкову та цільові позиції як вершини). Прямі лінії (шляхи), що з'єднують ці вершини, є найкоротшими відстанями між ними, і задача пошуку на утворюваному графі видимості полягає в знаходженні за допомогою методів пошуку на графах найкоротшого шляху з множини шляхів, існуючих між початковою та цільовою вершинами цього графа. Завдяки досить простій процедурі його побудови, граф видимості отримав досить широке використання на практиці. Водночас, таке подання має свої недоліки. По-перше, розмір графа видимості зростає зі збільшенням полігонів перешкод, що негативно впливає на ефективність використовуваних процедур пошуку на цьому графі. По-друге, ребра, що утворюються під час побудови графа видимості, проходять на мінімальній відстані від перешкод, що негативно впливає на безпечність навігації агента (гравця) обраним найкоротшим шляхом. В аналізованих дослідженнях [14, 23] цей метод використовується для створення графа, на якому агенти (гравці) виконують пошук шляхів за допомогою методів пошуку на графах.

Зауважимо, що за допомогою зазначених методів в рамках виконання пошуку шляхів також забезпечується вирішення проблем обходу перешкод навколишнього середовища.

2.5. Щодо підходів, використаних для формування стратегій переслідування/утікання. Як зазначено в п.1 статті, в аналізованих дослідженнях для формування стратегій переслідування/утікання використано такі підходи:

- геометричний підхід, заснований на побудові кіл (сфер) Аполлонія [15, 19];
- підхід, заснований на Марківських процесах ухвалення рішень [17, 24];
- підхід, заснований на використанні методів пошуку на графах [14, 23].

Коротко проаналізуємо названі підходи.

Геометричний підхід, заснований на побудові кіл (сфер) Аполлонія використано в [15, 19]. Зокрема, в [15] побудова кола (сфери) Аполлонія ґрунтується на використанні ізохронів. Ізохрон – це множина точок, які агент (гравець) може досягти одночасно, а перетин ізохронів переслідувачів та утікачів – це множина точок, до яких вони можуть прибути одночасно. В загальному випадку перетин ізохронів таких агентів (гравців), що рухаються з різною швидкістю, утворює коло (сферу) Аполлонія. Формування стратегії переслідування/утікання в [15] ґрунтується на побудові перетину ізохронів переслідувача, який рухається у тривимірному просторі, та ізохронів утікача, що рухається на площині, що від-

повідає перетину сфери Аполлонія, побудованої в 3D-просторі, та кола Аполлонія, побудованого на 2D-площині. В свою чергу, в [19] процес побудови сфер Аполлонія поєднано з модифікованим методом потенціальних полів, що в результаті дозволило сформувати раціональну стратегію руху агентів (гравців), в межах якої також вирішується проблема уникнення перешкод навколишнього середовища.

Підхід, заснований на Марківських процесах ухвалення рішень використано в [17, 24]. В загальному випадку Марківські процеси ухвалення рішень (МППР) – це основа для моделювання послідовного ухвалення рішень в ситуаціях, коли результати частково випадкові та частково залежать від дій агента (гравця). В МППР агент (гравець) діє в навколишньому середовищі та взаємодіє з ним. Агент (гравець) вибирає дії, а навколишнє середовище реагує на ці дії та формує нові ситуації для агента (гравця). Водночас, навколишнє середовище генерує винагороди – числові значення, які агент (гравець) прагне максимізувати з часом шляхом вибору дій.

Формально МППР описується кортежем $(s_t, a_t, p_{a_t}(s_t, s_{t+1}), r_{a_t}(s_t, s_{t+1}))$, де s_t – стан навколишнього середовища в момент часу t; a_t – дія, вчинена агентом (гравцем) в результаті процесу ухвалення рішень в момент часу t; $p_{a_t}(s_t, s_{t+1})$ – ймовірність переходу навколишнього середовища у стан s_{t+1} за умови виконання дії a_t в стані s_t; $r_{a_t}(s_t, s_{t+1})$ – винагорода, отримувана агентом (гравцем) в результаті виконання дії a_t, внаслідок якої відбувся перехід зі стану s_t у стан s_{t+1} (зазначимо, що $s_t, s_{t+1} \in S$, де S – скінченна множина станів навколишнього середовища; $a_t \in A$, де A – скінченна множина дій, які агент (гравець) може виконати).

Основним механізмом управління поведінкою агента (гравця) в МППР є функція винагороди. Надаючи позитивні та негативні винагороди агенту (гравцю), такий механізм здатний визначити, які дії приводять до позитивної винагороди, а рішення МППР дозволяють максимізувати очікування майбутньої винагороди.

Так, для вирішення проблем переслідування/утікання в [17] використовуються позитивні та негативні винагороди, які поєднуються разом, утворюючи напругу між потенційними діями. Наприклад, розміщення позитивної винагороди біля місця розташування утікача привертає увагу переслідувачів, але розміщення негативної винагороди на утікачі дозволяє запобігти зіткненню з ним. Між цими позитивними і негативними винагородами створюється природна рівновага, а це породжує бажану поведінку наближення переслідувача до утікача без зіткнення з ним.

У свою чергу, в [24] процес переслідування/утікання також моделюється як МППР, однак функція винагороди використовується дещо інакше. Зауважимо, що в [24] досліджується інтелектуальна стратегія маневру для гіперзвукових транспортних засобів. Ефективність цієї стратегії оцінюється двома показниками: величиною відхилення агента (гравця) від цілі та споживанням енергії під час маневру. При цьому, значення цих показників суперечать один одному протягом всього процесу переслідування/утікання: очікування збільшення величини відхилення від цілі вимагає більшого перевантаження під час маневрування, що потребує більше енергії, а економія енергії неминуче призведе до зменшення величини можливого відхилення. Тому ці показники мають бути кількісно кореговані, що здійснюється за рахунок коефіцієнта енергозбереження (який виступає як функція винагороди), а зміна величини коефіцієнта енергозбереження дозволяє кількісно регулювати два вищезгаданих показники.

Підхід, заснований на використанні методів пошуку на графах, розглянуто в [14, 23]. У цих дослідженнях пошук виконується на графі видимості (див. п.2.4). Зазначимо, що для пошуку шляхів на графі видимості можуть використовуватись різні методи (алгоритми), в тому числі: пошуку

вшир, пошуку вглиб, Дейкстри, A*, D*. В аналізованих дослідженнях [14, 23] для формування стратегій переслідування/утікання агентів (гравців) використано алгоритм A*.

Алгоритм A* є удосконаленням алгоритму Дейкстри, яке здійснено шляхом введення евристичної функції, що враховує властивості графу пошуку і дозволяє зменшити кількість досліджуваних вершин графу. В алгоритмі A* порядок обходу вершин графу визначається евристичною функцією $f(n)$ як сумою функції $g(n)$ вартості шляху від початкової до даної вершини n та функції $h(n)$ як евристичної оцінки вартості шляху від цієї вершини n до цільової: $f(n) = g(n) + h(n)$. Алгоритм A* досліджує граф ітеративно, розглядаючи шляхи, що виходять з початкової вершини. На кожному кроці ітерації алгоритм розглядає вершини, суміжні поточній вершині, та обирає ту з них, для якої $f(n)$ є мінімальною, після чого ця вершина «розкривається». В загальному випадку на кожному кроці ітерації алгоритм оперує з множиною шляхів від початкової вершини до ще не «розкритих» вершин графу. Такі шляхи зберігаються в черзі з пріоритетом, де пріоритет визначається за значенням $f(n)$. Алгоритм продовжує свою роботу доти, поки не досягне цільової вершини і не визначить шлях до неї з найменшим значенням $f(n)$. Недоліком алгоритму A* є те, що евристичні оцінки вартості шляхів задані остаточно і не можуть бути змінені в процесі роботи алгоритму (без його перезапуску), що не дозволяє коректно планувати шляхи в динамічно змінюваному середовищі.

Зазначимо, що в [14] граф видимості створюється агентом (гравцем) з використанням карти висот. Алгоритм A* виконує пошук мінімального шляху на такому графі, де для кожного ребра графа призначається функція вартості, яка визначається з урахуванням обчислення мінімальної відстані до інцидентної вершини графа. В свою чергу, в [23] графом видимості виступає навігаційна сітка, яка визначає можливі місця, яким може рухатись агент (гравець). Ребрам такого графа поставлені у відповідність вартісні оцінки, які враховують відстані між вершинами графа та використовуються алгоритмом A* в процесі пошуку мінімального шляху.

Висновки

Виконаний аналітичний огляд відкритих джерел дозволяє, на нашу думку, отримати загальне уявлення про основні світові тенденції, що склалися на сьогодні в дослідженнях процесів переслідування/утікання у тривимірному просторі.

Зокрема, з виконаного огляду стає очевидною однакова зацікавленість в отриманні як теоретичних, так і прикладних результатів досліджень процесів переслідування/утікання у тривимірному просторі, що пояснюється великою актуальністю цієї сфери досліджень для різних галузей застосування (насамперед військової галузі). Водночас, у переважній більшості випадків об'єктом досліджень стають безпілотні літальні й автономні підводні апарати, що наразі відповідає широкому попиту на використання цих апаратів для виконання спеціальних операцій (в тому числі військових). У цьому контексті підвищується актуальність задачі врахування впливу фактору навколишнього середовища на досліджувані об'єкти (вітрових навантажень, підводних потоків, особливостей рельєфу місцевості). Необхідність розв'язання таких задач потребувала вирішення низки проблем, в тому числі проблем планування шляхів в навколишньому середовищі з урахуванням уникнення перешкод. А вирішення проблем планування шляхів, у свою чергу, актуалізувало необхідність вирішення проблем формування стратегій переслідування/утікання у тривимірному просторі.

Наголосимо, що окреслені вище аспекти характеризують лише основні напрями досліджень проблем переслідування/утікання у тривимірному просторі, але не обмежують їх.

ЛІТЕРАТУРА

1. Айзекс Р. Дифференциальные игры. М.: Мир, 1967. 479 с.
2. Красовский Н.Н., Субботин А.И. Позиционные дифференциальные игры. М.: Наука, 1974. 456 с.

3. Петросян Л.А., Рисхиев Б.Б. Преследование на плоскости. М.: Наука, 1991. 91 с.
4. Петросян Л.А., Томский Г.В. Геометрия простого преследования. Новосибирск: Наука, 1983. 140 с.
5. Пшеничный Б.Н., Остапенко В.В. Дифференциальные игры. К.: Наукова думка, 1992. 259 с.
6. Чикрий А.А. Конфликтно управляемые процессы. К.: Наукова думка, 1992. 364 с.
7. Weintraub I.E., Pachter M., García E. An Introduction to Pursuit-evasion Differential Games. *2020 American Control Conference (ACC)*, pp. 1049-1066, DOI: 10.23919/ACC45564.2020.9147205
8. Яловець А.Л. Мультиагентне моделювання переслідування на площині: від теорії до програмної реалізації. К.: Наукова думка, 2019. 165 с.
9. Lopez V.G., Lewis F.L., Wan Y., Sanchez E.N., Fan L. Solutions for Multiagent Pursuit-Evasion Games on Communication Graphs: Finite-Time Capture and Asymptotic Behaviors. *IEEE Transactions on Automatic Control*, 2020. Vol. 65, No. 5, pp. 1911-1923, DOI: 10.1109/TAC.2019.2926554
10. Deng Z., Kong Z. Multi-Agent Cooperative Pursuit-Defense Strategy Against One Single Attacker. *IEEE Robotics and Automation Letters*, 2020. Vol. 5, No. 4, pp. 5772-5778, DOI: 10.1109/LRA.2020.3010740
11. Liang X., Zhou B., Jiang L., Meng G., Xiu Y. Collaborative pursuit-evasion game of multi-UAVs based on Apollonius circle in the environment with obstacle. *Connection Science*, 2023. Vol. 35, Iss.1, pp. 1-24, DOI: 10.1080/09540091.2023.2168253
12. Paczolay G., Harmati I. A Simplified Pursuit-evasion Game with Reinforcement Learning. *Periodica Polytechnica Electrical Engineering and Computer Science*, 2021. Vol. 65, No. 2, pp. 160-166, DOI: 10.3311/PPee.16540
13. de Souza C., Newbury R., Cosgun A., Castillo P., Vidolov B., Kulić D. Decentralized Multi-Agent Pursuit Using Deep Reinforcement Learning. *IEEE Robotics and Automation Letters*, 2021. Vol. 6, No. 3, pp. 4552-4559, DOI: 10.1109/LRA.2021.3068952
14. Kolling A., Kleiner A., Lewis M., Sycara K. Pursuit-evasion in 2.5d based on team-visibility. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, 2010, pp. 4610-4616, DOI: 10.1109/IROS.2010.5649270
15. Li S., Wang C., Xie G. Pursuit-evasion differential games of players with different speeds in spaces of different dimensions, *2022 American Control Conference*, Atlanta, GA, USA, 2022, pp. 1299-1304, DOI: 10.23919/ACC53348.2022.9867329
16. Özkahraman Ö., Ögren P. Underwater Caging and Capture for Autonomous Underwater Vehicles. *Global Oceans 2020: Singapore – U.S. Gulf Coast*, Biloxi, MS, USA, 2020, pp. 1-8, DOI: 10.1109/IEEECONF38699.2020.9389311
17. Bertram J.R., Wei P. An Efficient Algorithm for Multiple-Pursuer-Multiple-Evader Pursuit/Evasion Game. *ArXiv,* 2019, *abs/1909.04171,* DOI: 10.48550/arXiv.1909.04171
18. Chen N., Li L., Mao W. Equilibrium Strategy of the Pursuit-Evasion Game in Three-Dimensional Space. *IEEE/CAA Journal of Automatica Sinica*, 2024. Vol. 11, No. 2, pp. 446-458, DOI: 10.1109/JAS.2023.123996Ya
19. Khachumov M., Khachumov V. Modeling the Solution of the Pursuit–Evasion Problem Based on the Intelligent–Geometric Control Theory. *Mathematics*, 2023. Vol. 11, Is. 23, 4869. DOI: 10.3390/math11234869
20. Liang X., Wang H., Luo H. Collaborative Pursuit-Evasion Strategy of UAV/UGV Heterogeneous System in Complex Three-Dimensional Polygonal Environment, *Complexity*, 2020. Vol. 2020, pages 1-13, DOI: 10.1155/2020/7498740
21. Segal A., Miloh T. Barrier strategies and capture criteria in a 3D pursuit-evasion differential game. *Optimal Control Applications & Methods,* 1995. Vol. 16, Is. 5, pp. 321-340, DOI: 10.1002/j.1099-1514.1995.tb00024.x
22. Sun W., Tsiotras P., Yezzi A.J. Multiplayer Pursuit-Evasion Games in Three-Dimensional Flow Fields. *Dynamic Games and Applications*, 2019. Vol. 9, Is. 4, pp. 1188-1207, DOI: 10.1007/s13235-019-00304-4
23. Şahin İ., Kumbasar T. Catch me if you can: A pursuit-evasion game with intelligent agents in the Unity 3D game environment, *2020 International Conference on Electrical Engineering (ICEE)*, Istanbul, Turkey, 2020, pp. 1-6, DOI: 10.1109/ICEE49691.2020.9249828
24. Yan T., Jiang Z., Li T., Gao M., Liu C. Intelligent maneuver strategy for hypersonic vehicles in three-player pursuit-evasion games via deep reinforcement learning. *Frontiers in Neuroscience*, 2024. 18:1362303, DOI: 10.3389/fnins.2024.1362303
25. Яловець А.Л. До постановки задачі розпізнавання невідомого оточуючого середовища, навігації та планування шляхів аген-

том в ньому. *Проблеми програмування*. 2018. № 1. С. 113-127, DOI: 10.15407/pp2018.01.113

REFERENCES

1. Isaaks R. (1967) Differential games. Mir. 479 p. (in Russian)
2. Krasovsky N.N., Subbotin A.I. (1974) Positional differential games. Nauka. 456 p. (in Russian)
3. Petrosyan L.A., Riskhiev B.B. (1991) Pursuit on the plane. Nauka. 91 p. (in Russian)
4. Petrosyan L.A., Tomsky G.V. (1983) Geometry of simple pursuit. Nauka. 140 p. (in Russian)
5. Pshenichny B.N., Ostapenko V.V. (1992) Differential games. Naukova dumka. 259 p. (in Russian)
6. Chikriy A.A. (1992) Conflict-controlled processes. Naukova dumka. 364 p. (in Russian)
7. Weintraub I.E., Pachter M., García E. (2020) An introduction to pursuit-evasion differential games. *2020 American Control Conference (ACC)*, pp. 1049-1066, DOI: 10.23919/ACC45564.2020.9147205
8. Yalovets A.L. (2019) Multi-agent modeling of pursuit on the plane: from theory to software implementation. Naukova dumka. 165 p. (in Ukrainian)
9. Lopez V.G., Lewis F.L., Wan Y., Sanchez E.N., Fan L. (2020) Solutions for multiagent pursuit-evasion games on communication graphs: finite-time capture and asymptotic behaviors. *IEEE Transactions on Automatic Control*. Vol. 65, No. 5, pp. 1911-1923, DOI: 10.1109/TAC.2019.2926554
10. Deng Z., Kong Z. (2020) Multi-agent cooperative pursuit-defense strategy against one single attacker. *IEEE Robotics and Automation Letters*. Vol. 5, No. 4, pp. 5772-5778, DOI: 10.1109/LRA.2020.3010740
11. Liang X., Zhou B., Jiang L., Meng G., Xiu Y. (2023) Collaborative pursuit-evasion game of multi-UAVs based on Apollonius circle in the environment with obstacle. *Connection Science*. Vol. 35, Iss.1, pp. 1-24, DOI: 10.1080/09540091.2023.2168253
12. Paczolay G., Harmati I. (2021) A simplified pursuit-evasion game with reinforcement learning. *Periodica Polytechnica Electrical Engineering and Computer Science*. Vol. 65, No. 2, pp. 160-166, DOI: 10.3311/PPee.16540
13. de Souza C., Newbury R., Cosgun A., Castillo P., Vidolov B., Kulić D. (2021) Decentralized multi-agent pursuit using deep reinforcement learning. *IEEE Robotics and Automation Letters*. Vol. 6, No. 3, pp. 4552-4559, DOI: 10.1109/LRA.2021.3068952
14. Kolling A., Kleiner A., Lewis M., Sycara K. (2010) Pursuit-evasion in 2.5d based on team-visibility. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan. pp. 4610-4616, DOI: 10.1109/IROS.2010.5649270
15. Li S., Wang C., Xie G. (2022) Pursuit-evasion differential games of players with different speeds in spaces of different dimensions, *2022 American Control Conference*, Atlanta, GA, USA. pp. 1299-1304, DOI: 10.23919/ACC53348.2022.9867329
16. Özkahraman Ö., Ögren P. (2020) Underwater caging and capture for autonomous underwater vehicles. *Global Oceans 2020: Singapore – U.S. Gulf Coast*, Biloxi, MS, USA, pp. 1-8, DOI: 10.1109/IEEECONF38699.2020.9389311
17. Bertram J.R., Wei P. (2019) An Efficient algorithm for multiple-pursuer-multiple-evader pursuit/evasion game. *ArXiv, abs/1909.04171,* DOI: 10.48550/arXiv.1909.04171
18. Chen N., Li L., Mao W. (2024) Equilibrium strategy of the pursuit-evasion game in three-dimensional space. *IEEE/CAA Journal of Automatica Sinica*. Vol. 11, No. 2, pp. 446-458, DOI: 10.1109/JAS.2023.123996Ya
19. Khachumov M., Khachumov V. (2023) Modeling the solution of the pursuit–evasion problem based on the intelligent–geometric control theory. *Mathematics*. Vol. 11, Is. 23, 4869. DOI: 10.3390/math11234869
20. Liang X., Wang H., Luo H. (2020) Collaborative pursuit-evasion strategy of uav/ugv heterogeneous system in complex three-dimensional polygonal environment. *Complexity*. Vol. 2020, pages 1-13, DOI: 10.1155/2020/7498740
21. Segal A., Miloh T. (1995) Barrier strategies and capture criteria in a 3D pursuit-evasion differential game. *Optimal Control Applications & Methods*. Vol. 16, Is. 5, pp. 321-340, DOI: 10.1002/j.1099-1514.1995.tb00024.x
22. Sun W., Tsiotras P., Yezzi A.J. (2019) Multiplayer pursuit-evasion games in three-dimensional flow fields. *Dynamic Games and Applications*. Vol. 9, Is. 4, pp. 1188-1207, DOI: 10.1007/s13235-019-00304-4
23. Şahin İ., Kumbasar T. (2020) Catch me if you can: A pursuit-evasion game with intelligent agents in the Unity 3D game environment, *2020 International Conference on Electrical Engineering (ICEE)*, Istanbul, Turkey. pp. 1-6, DOI: 10.1109/ICEE49691.2020.9249828

24. Yan T., Jiang Z., Li T., Gao M., Liu C. (2024) Intelligent maneuver strategy for hypersonic vehicles in three-player pursuit-evasion games via deep reinforcement learning. *Frontiers in Neuroscience*. 18:1362303, DOI: 10.3389/fnins.2024.1362303
25. Yalovets A.L. (2018) On the problem of recognizing the unknown environment, navigating and planning paths by an agent in it. *Problems in Programming*. № 1. pp. 113-127. (in Ukrainian) DOI: 10.15407/pp2018.01.113

Одержано: 25.02.2025
Внутрішня рецензія отримана: 04.03.2025
Зовнішня рецензія отримана: 07.03.2025

Про авторів:

Романенко Ігор Олександрович,
доктор технічних наук, професор,
провідний науковий співробітник
https://orcid.org/0000-0001-5339-7900

Яловець Андрій Леонідович,
доктор технічних наук,
старший науковий співробітник,
головний науковий співробітник
http://orcid.org/0000-0001-6542-3483

Місце роботи авторів:

Інститут проблем математичних машин
і систем НАН України,
Тел.: (+38) 044 526 13 69.
E-mail: andriy.yalovets@gmail.com

УДК 517.958:57 +519.711.3 + 612.51.001 https://doi.org/10.15407/pp2025.01.013

R.D. Grygoryan, A.G. Degoda, M.V. Progonnyi

“*G_Sim*” SOFTWARE PROVIDING SIMULATIONS OF HUMAN PHYSIOLOGICAL RESPONSES TO ±Gz ACCELERATIONS

Specialized software “*G_Sim*”, providing simulations of human physiological responses to dynamic Gz accelerations, is created and tested. “*G_Sim*” is based on a previously developed and published quantitative mathematical model (QMM) that describes human hemodynamics under given Gz profiles without or with special protective tools and algorithms. “*G_Sim*” is a modern information technology realized as an autonomic executive module in the Delphi Pascal environment. By default, the biological parameters of QMM are tuned for the mean man, who is 175 cm in height and has a 70 kg mass. “*G_Sim*” has an intuitive user interface (UI) that provides the user with procedures necessary to actualize characteristics of QMM, realize a computer experiment (simulation), visualize its results in graph forms for analysis, and save the chosen data for further analysis. The actualization concerns biological data associated with human sex, anthropometrics, age, and non-biological characteristics including acceleration profiles, characteristics of the anti-G suit, breathing techniques, and muscle stressing mode. UI's special windows provide additional tunings of the basic QMM. “*G_Sim*” upgrades the traditional training techniques on centrifuges and test flights. The novel beneficial effect of “*G_Sim*” provides the future fighter pilot with realistic-like visual knowledge concerning the dynamics of physiological and protective events. Therefore, simulations will clearly show ways to optimize the combination of artificial protections to prevent negative effects (loss of vision or consciousness). Such knowledge will shorten training and minimize the anthropogenic risk of serious injuries or catastrophes during the training. Test simulations presented in the paper mainly illustrate the potential of “*G_Sim*” as an assistant informational technology.

Keywords: fighter pilot, training, risk, catastrophe, information technology.

Р.Д. Григорян, А.Г. Дегода, М.В. Прогонний

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ «*G_Sim*» ДЛЯ СИМУЛЯЦІЇ ФІЗІОЛОГІЧНИХ РЕАКЦІЙ ЛЮДИНИ НА ±Gz ПРИСКОРЕННЯ

Створено та протестовано спеціалізоване програмне забезпечення «*G_Sim*», що забезпечує моделювання фізіологічних реакцій людини на динамічні прискорення Gz. «*G_Sim*» базується на раніше розроблені та опубліковані кількісні математичні моделі (КММ), яки описують гемодинаміку людини за заданими профілями Gz без або з використанням спеціальних захисних інструментів і алгоритмів. «*G_Sim*» — сучасна інформаційна технологія, реалізована у вигляді автономного виконавчого модуля в середовищі Delphi Pascal. За замовчуванням біологічні параметри QMM налаштовані на середнього чоловіка, який має зріст 175 см і вагу 70 кг. «*G_Sim*» має інтуїтивно зрозумілий інтерфейс користувача (ІК), який надає користувачеві процедури, необхідні для актуалізації характеристик КММ, реалізації комп’ютерного експерименту (симуляції), візуалізації його результатів у вигляді графіків для аналізу та збереження вибраних даних для подальшого аналізу. Актуалізація стосується біологічних даних, пов’язаних зі статтю людини, антропометричними показниками, віком і небіологічними характеристиками, включаючи профілі прискорення, характеристики анти-G костюма, техніки дихання та режим навантаження на м’язи. Спеціальні вікна ІК забезпечують додаткові налаштування основного КММ. «*G_Sim*» вдосконалює традиційні методи навчання на центрифугах і тестових польотах. Новий корисний ефект «*G_Sim*» є в тому, що симуляції надають майбутньому пілоту винищувача реалістичні візуальні уяви щодо динаміки фізіологічних і захисних подій. Таким чином, симуляції чітко покажуть шляхи оптимізації комбінації штучних засобів захисту для запобігання негативним ефектам (втрата зору чи свідомості). Такі знання скоротять навчання та мінімізують антропогенний ризик серйозних травм або катастроф під час навчання. Тестове моделювання, представлене в статті, в основному ілюструє потенціал «*G_Sim*» в якості допоміжної, інформаційної технології.

Ключові слова: пілот-винищувач, навчання, ризик, катастрофа, інформаційні технології.

Introduction

Modern high maneuverable fighter aircraft is a source of rapid altering and often highly sustained extreme accelerations [1-3]. Both physiological [4-9] and biotechnical [10-12] problems that arose in parallel with an increase in military aircraft's maneuverability have been properly investigated [4-18].

Human physiology evolutionarily adapted to the one g Earth environment, cannot provide adequate functioning of the brain and eyes of a sitting person. These organs, very sensitive to oxygen and glucose supply, suffer in parallel with the decreasing of their input blood pressure. Under accelerations, the hydrostatic pressure increases proportionally to the acceleration value. This additional factor creates opposite effects in vessels located upper or lower the heart: in upper arteries blood inflows become difficult while the flow toward body lower regions becomes easier. In veins, alterations are opposite directions. The altered pressure gradients redistribute blood volumes worsening the circulation at the cardiovascular scale. Accelerations also alter the ventilation-perfusion ratio in lungs [13,14].

Most critical are extreme value positive (+Gz) accelerations acting in the direction of head-legs, or negative (-Gz) accelerations acting in the opposite direction [4-6]. In terminal zones (brain, eyes), the lowered circulation causes oxygen lack and worsens the pilot's vision and consciousness [9,12]. Under -Gz, the elevated local blood pressure in the eyes and brain causes rupture of microscopic vessels and hemorrhages. Both the value of Gz and the gradient of acceleration change play an essential role in these events.

Under relatively slow (0.1-0.4 g/sec) linearly increasing +Gz accelerations, a mean healthy person not using artificial protections is operable for approximately +4Gz accelerations [11]. Further elevation of the G-load causes the G-lock phenomenon usually disappearing after a break [2,6,8]. Modern fighter aircrafts can provide acceleration gradients exceeding 2 g/sec. This requires special protection algorithms and devices. Currently, typical protection algorithms include the use of special pneumatic or water-augmented anti-G suits, muscle stress, as well as breathing with a positive pressure air [1,11,17]. The adaptive protection algorithms combining multiple methods depending on the dynamics of accelerations are the most effective. So, technologies helping to optimize the use of protective methods and tools are encouraged.

Traditionally, empiric research on centrifuges is the main way for inventing more effective protections [1,5,6-8]. Mathematical models realized as special software [18-20] showed additional ways for maximizing the individual resistance of a pilot to the negative effects of accelerations. The experience in the last area was taken into account during the development of an advanced version of basic models [21-23] necessary to create our current version of "***G_Sim"*** software which is autonomic executive software oriented to PC.

The goal of this article is to inform potential users of our simulator about its purpose and possibilities.

The user interface of "*G_Sim"*

Every interaction with **"*G_Sim"*** is provided by the user interface (UI). Its general view presented in Fig.1. indicates that **"*G_Sim"*** is oriented to problems associated with dynamic accelerations that appear either when employing a professional centrifuge or during the piloting of military fighter aircraft.

The mathematical model of cardiovascular physiology of a healthy and physically well-trained human sitting in a standard aviation chair is the basis of our simulator [22,23]. Fig.1. also shows that standard protection tools are also modeling subjects.

In the upper left sector of Fig.1., one can see eight special icons that provide the user with all the procedures necessary to prepare and execute a single computer experiment (simulation).

The icon containing a picture of a sited human and the abbreviation "SETS" is the main one clicking which the user opens a window shown in Fig.2.

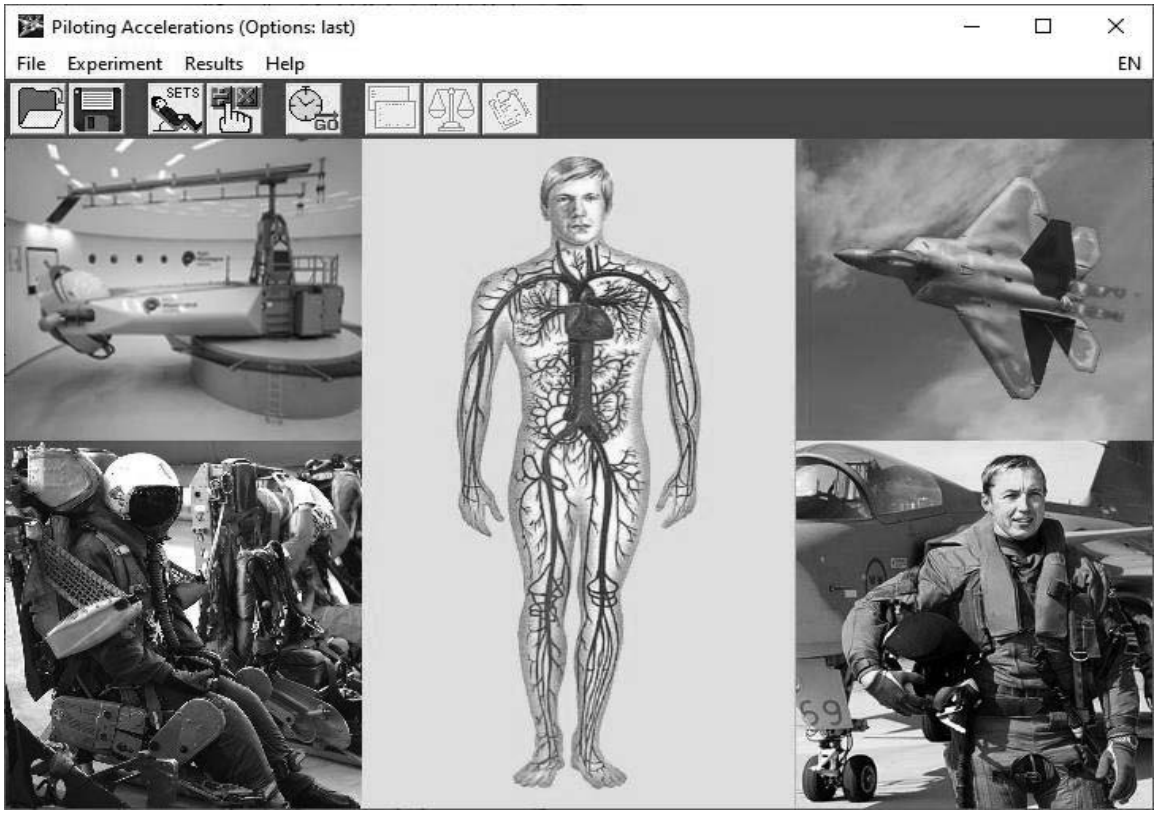


Fig. 1. General view of the user interface of ***"G_Sim"***.

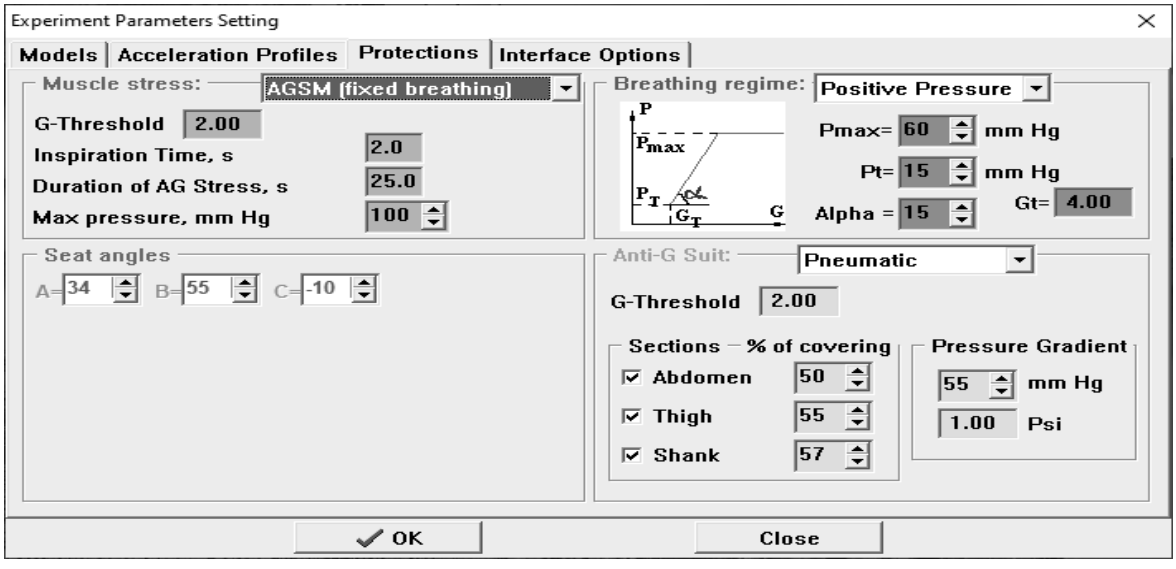


Fig. 2. The main window form to prepare a single simulation. This image shows the expanded content of the operations that can be accessed by clicking on the menu bar "Protections".

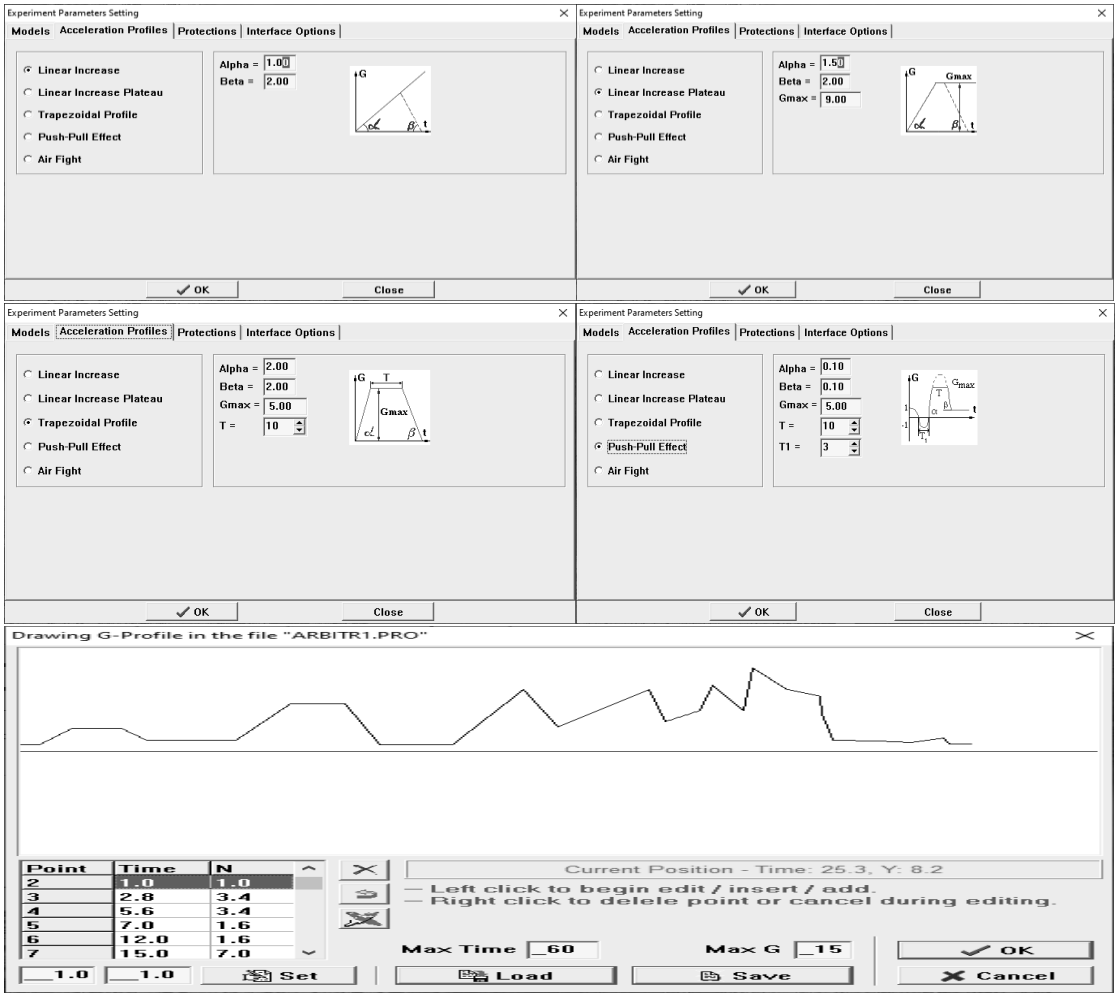
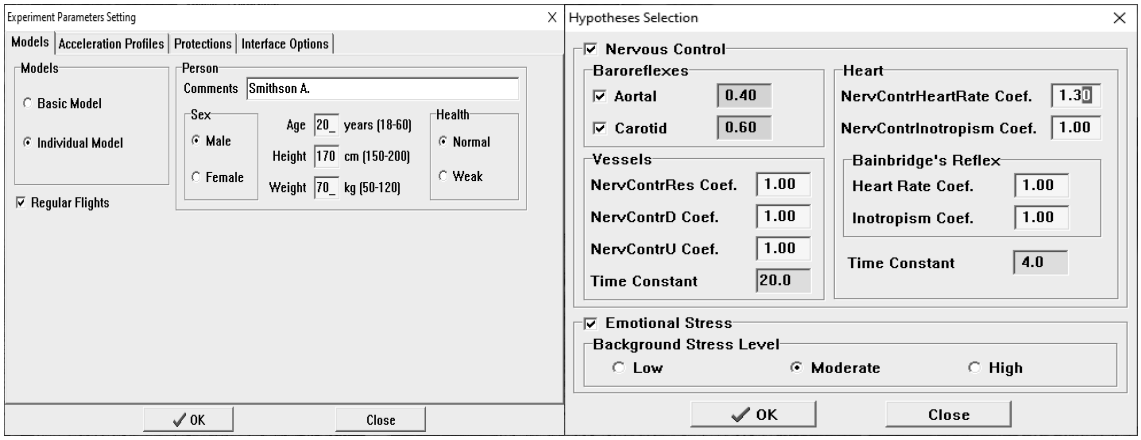


Fig. 3. The window form provides settings of parameters that determine the acceleration profile. Using the bottom-located form the user can construct an arbitrary acceleration profile imitating complex combat maneuvers.

By clicking on the menu bar "Models" or "Interface Options", the user can actualize the physiological model.



A B

Fig. 4. Special window forms provide settings of parameters that determine actual parameters of QMM: A) basic or personal model including the health level; B) activities of physiological mechanisms controlling the circulation.

The icon "GO" located in the central area of Fig. 1. starts the program's calculation according to the actualized set of parameters. According to the algorithm, the calculation is over if special events (e.g., G-LOC) happen or the time limit is used. Then, "G-Sim" builds graphs presenting the dynamics of model characteristics. They include both physiological and technical data. The physiological data concern blood pressures, flows, and volumes in certain body parts. The technical data concerns specific parameters of protection. Theoretically, the data set could provide advanced experts with additional capabilities for investigating new algorithms for protection optimization.

In this article, we illustrated only a part of the information. The main window to illustrate the most important information contains three sections. Each combines a special sub-set of variables (see Figures 5-9).

Fig.5. represents the basic data concerning a relaxed healthy human sitting in an armchair but without using any protection. The bottom section presents the acceleration dynamics. The middle section presents the dynamics of the pressure PExt provided by a compressor and six specific pressures (in this simulation, pressures in three sections of the pneumatic anti-G suit are not presented but are calculated and can be illustrated using specific activators). PExtThr, PMuscle, and PBrLiq represent pressures in the thorax, body muscles, and liquor respectively. Hemodynamic variables are collected in the upper section. In this case, end-systolic (APs) and end-diastolic (APd) are not shown. MAP is the mean pressure in the aortic arch, CO is the cardiac output, SV is the stroke volume, HR is the heart rate, MCAP, PES, and CVP represent mean pressures in the carotid sinus, eye arteries, and central vein respectively. Vertical dotted lines indicate time moments for acceleration start and maximal levels. In this simulation, neither G-LOC nor vision loss happened: the simulation scenario was realized totally.

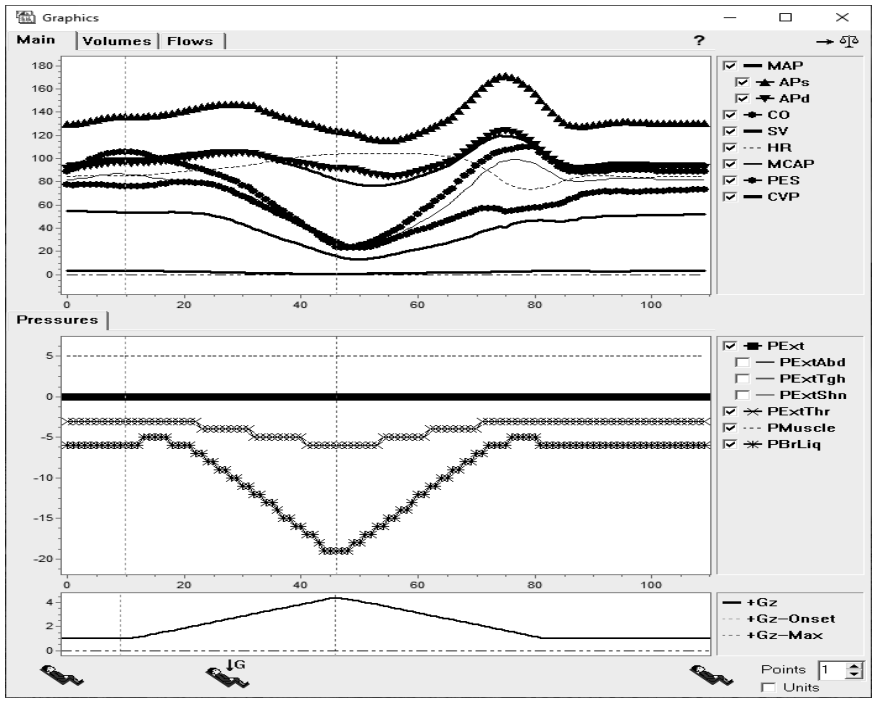


Fig. 5. The basic simulation illustrates the physiological responses of a relaxed healthy human to a slow altering (0,1 g/sec) linear profile acceleration. The person sitting in an armchair does not use protection. Before G-onset (marked with a first vertical dotted line) parameters indicate a practically steady-state mode. At the 46th second of a load (marked with a second vertical dotted line), at a value of G=4,35g, the program automatically activated break because of the G-LOC event.

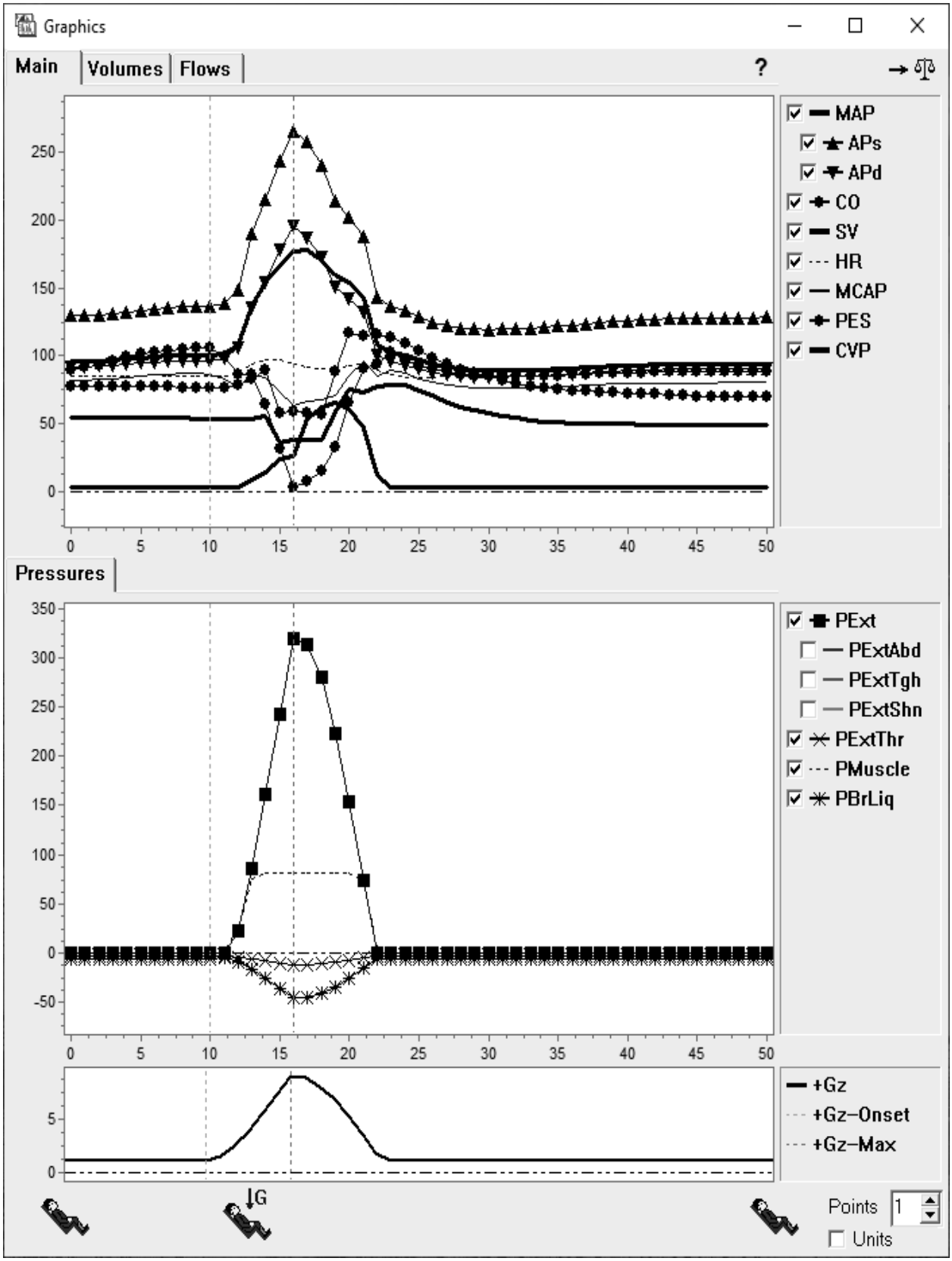


Fig. 6. A simulation scenario with a trapezoidal G-profile using a standard pneumatic anti-G suit, natural breathing, and moderate muscle stress.

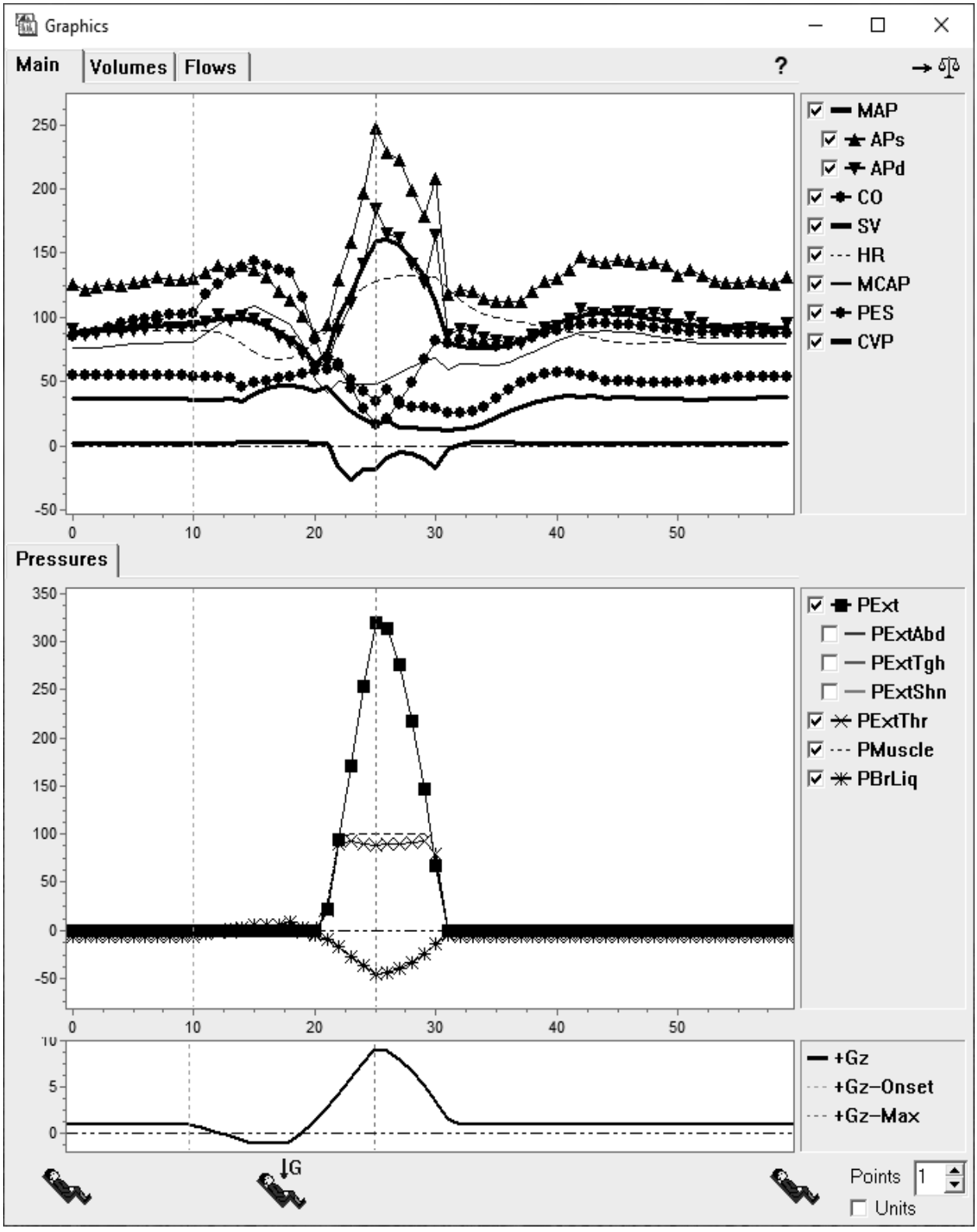


Fig. 7. A simulation of a "Push-Pull" scenario using standard pneumatic anti-G suit, natural breathing, and AGSM-technique with maximal muscle stress of 100 mm Hg accompanied with inspiration time of 2 sec and duration of AG-stress of 20 sec.

As Fig. 7. illustrates, our "mean man" resisted up to 9g accelerations for a 20 sec plateau. Pay attention that end-systolic (APs) and end-diastolic (APd) pressures are also shown.

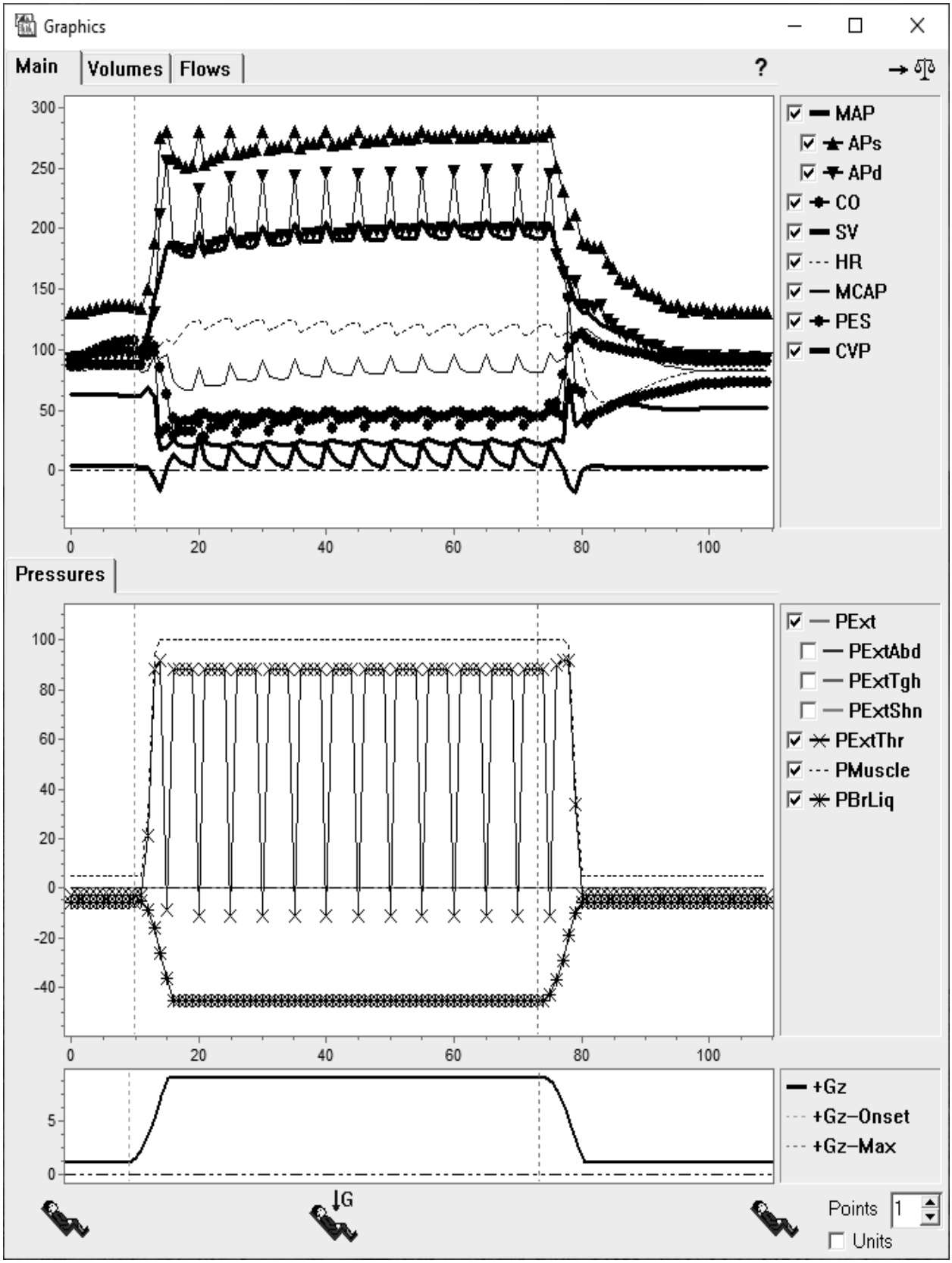


Fig. 8. A simulation of a trapezoidal acceleration scenario with a long-lasting plateau using a hydraulic anti-G suit, natural breathing, and special technique of AGSM (sharp inspirations of 2 sec, maximal muscle stress of 100 mm Hg for 3 sec, and sharp expirations of 2 sec).

Discussion

Not all the information concerning the capabilities of our "G-Sim", in particular, describing functionalities of the icons of UI was presented in this article. In addition to the space limit, another reason is that the version of "G-Sim" used in this publication is not yet the final software. We continue to work on upgrading software to make it maximally useful and convenient in practice.

Although the mean man model used in this "G-Sim" already provides the student-pilot with important visualized dynamics of physiological and technical data. Every pilot has specific anatomical, physiological, and psychological individualities that potentially can modify the pilot's resistance to negative effects of accelerations. Therefore, we are working on algorithms that, being not very complex, could provide the individualizations of basic mathematical models. Principally, we hope to achieve acceptable results using relatively simple algorithms that correct initial parameters of BMM mainly using passport and anthropological data (namely, such data is reflected in the window form in Figure 4A).

Another aspect of upgrading our "***G-Sim***" we see in imitating characteristic phenomena, caused by a deterioration of the eyes and brain oxygen supply. We already have created a model and program modules visually imitating: 1) the narrowing of the field of peripheral vision including the loss of vision; 2) loss of consciousness as an extreme manifestation (G-lock).

Certainly, the main goal of our "G-Sim" is to facilitate the pilot's acquiring the needed skills. In this context, an essential role does play the factor of dynamics. As physical events develop to speed, in-time counteracts are extremely important to provide effective resistance. "***G-Sim***" is the single technology using which the student-pilot can imitate every thinkable scenario and find the most effective combination of algorithms for maximizing the protective effect.

An additional use of "***G-Sim***" is that it can be used to provide "post-factum" simulations for analysis and understanding non-trivial causes of failures.

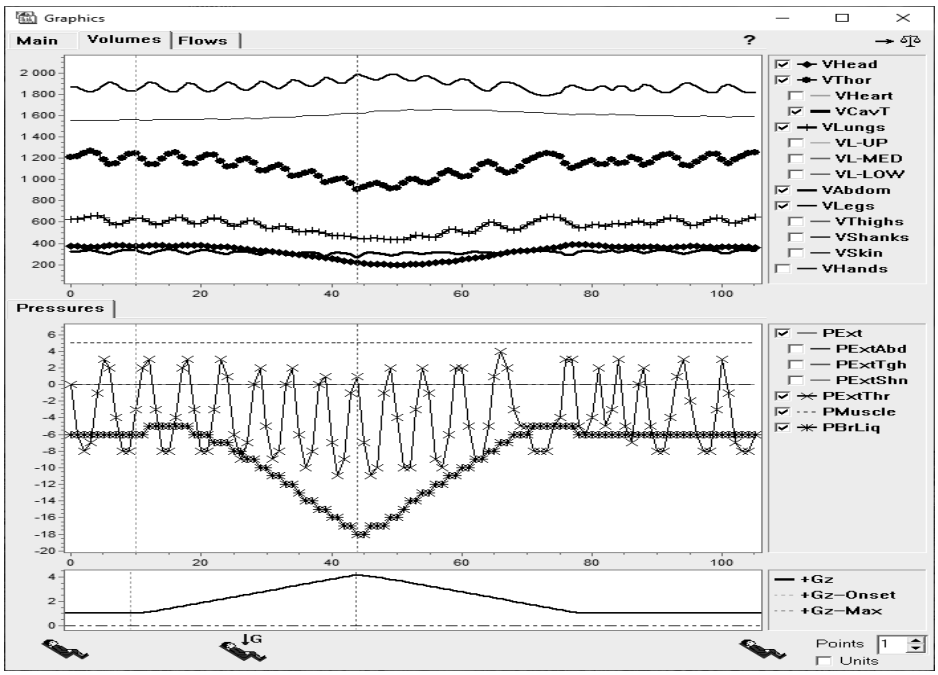


Fig. 9. The simulation scenario described in Fig.5. This case the dynamics of blood volumes in certain body sections are illustrated.

Conclusion

Combat maneuvers of modern fighter aircraft originate extreme accelerations negatively influencing on pilot's physiology and operability. Until recently, empirical investigations were the only way to develop and test

protective methods and tools providing fighter pilots functionality under combat maneuvers. The main tools used for acquiring student pilot initial skills necessary to resist the negative effects of dynamic extreme accelerations were centrifuges. The skilling process of student pilots of modern fighter aircraft is not duly formalized yet. Our special computer simulator "***G-Sim***" provides the user with a user-friendly intuitive interface for construction and execution of a computer experiment (a simulation) that visualizes additional dynamic variables concerning characteristics of both human physiology and protections under arbitrarily formed acceleration profiles. By comparing human physiological responses under different simulated scenarios (without use of protections, with use of their different combinations), student-pilots and their instructors can optimize the individually optimal tactics for maximizing the resistance and performance capability of the future fighter pilot.

References

1. Burton, R.R.,Whinnery, J.E. Biodynamics: Sustained acceleration. In Fundamentals of Aerospace Medicine, 3rd ed.; DeHart, R.L., Davis, J.R., Eds.; Lippincott Williams & Wilkins: Philadelphia, PA, USA, 2002; pp. 122–153.
2. Slungaard E., McLeod J., Green, N.D.C., Kiran A., Newham D.J., Harridge S.D.R. Incidence of g-induced loss of consciousness and almost loss of consciousness in the Royal Air Force. Aerosp. Med. Hum. Perform. 2017, 88, 550–555.
3. Newman D.G. The cardiovascular system at high Gz. In High G Flight: Physiological Effects and Countermeasures, 1st ed.; Newman, D.G., Ed.; Ashgate: Farnham, UK, 2015; pp. 57–72.
4. Park M., Yoo S., Seol H., Kim C., Hong Y. Unpredictability of fighter pilots' G duration by anthropometric and physiological characteristics. Aerosp. Med. Hum. Perform. 2015, 86, 307–401.
5. Yun, C.; Oh, S.; Shin, Y.H. AGSM proficiency and depression are associated with success of high-G training in trainee pilots. Aerosp. Med. Hum. Perform. 2019, 90, 613–617.
6. Polock R.D., Hodkinson, P.D., Smith T.G. Oh G: The x,y and z of human physiological responses to acceleration. Experimental Physiology, 2021,106,2367–2384. https://doi.org/10.1113/EP089712
7. Albery W. B. Acceleration in other axes affects +Gz tolerance: Dynamic centrifuge simulation of agile flight. Aviation, Space, and Environmental Medicine, 2004,75(1), 1–6.
8. Burton R., Whinnery J. Operational G-induced loss of consciousness: Something old; something new. Aviation, Space, and Environmental Medicine, 1985,56(8), 812–817.
9. Cao X.-S., Wang Y.-C., Xu L., Yang C.-B., Wang B., Geng J., Gao Y., Wu Y. H., Wang X. Y., Zhang S., Sun X.-Q. Visual symptoms and G-induced loss of consciousness in 594 Chinese Air Force aircrew— A questionnaire survey. Military Medicine, 2012, 177(2), 163–168. https://doi. org/10.7205/milmed-d-11-00003.
10. Chung K. Y. Cardiac arrhythmias in F-16 pilots during aerial combat maneuvers (ACMS): A descriptive study focused on G-level acceleration. Aviation, Space, and Environmental Medicine, 2001,72(6), 534– 538.
11. Eiken O., Bergsten, E., Grönkvist M. G-Protection mechanisms afforded by the anti-G suit abdominal bladder with and without pressure breathing. Aviation, Space, and Environmental Medicine, 2011. 82(10), 972–977. https://doi.org/10.3357/ASEM.3058.2011
12. Eiken O., Keramidas M. E., Taylor N. A. S., Grönkvist M., Gronkvist M. Intraocular pressure and cerebral oxygenation during prolonged headward acceleration. European Journal of Applied Physiology, 2017,117(1), 61–72. https://doi.org/10.1007/s00421-016-3499-3
13. Grönkvist M., Bergsten E., Eiken O. Lung mechanics and transpulmonary pressures during unassisted pressure breathing at high Gz loads. Aviation, Space, and Environmental Medicine, 2008,79(11), 1041–1046. https://doi.org/10.3357/ASEM.2371.2008.
14. Henderson A. C., Sá R. C., Theilmann R. J., Buxton R. B., Prisk G. K., Hopkins S. R. The gravitational distribution of ventilation-perfusion ratio is more uniform in prone than supine posture in the normal human lung. Journal of Applied Physiology, 2013, 115(3), 313–324. https://doi.org/10.1152/JAPPLPHYSIOL.01531.2012

15. MacDougall J. D., McKelvie R. S., Moroz D. E., Buick, F. The effects of variations in the anti-G straining maneuver on blood pressure at +Gz acceleration. Aviation, Space, and Environmental Medicine, 1993,64(2), 126–131.
16. Sundblad P., Kölegård R., Migeotte P. F., Delière Q., Eiken O. The arterial baroreflex and inherent G tolerance. EuropeanJournal of Applied Physiology, 2016,116(6), 1149–1157. https://doi.org/10.1007/s00421-016-3375-1
17. Whiny J.E., Forster E. The +Gz-induced loss of consciousness curve. Extreme Physiology and Medicine, 2013,2(1),19. https://doi.org/10.1186/2046-7648-2-19
18. Grygoryan R.D., Kochetenko E.M., Informational technology for modeling of fighters medical testing procedures by centrifuge accelerations. Selection &Training Advances in Aviation: AGARD Conference Proceedings 588; Prague, 1996. May 25-31, PP3,1-12.
19. Grygoryan R.D., High sustained G-tolerance model development.STCU#P-078 EOARD# 01-8001 Agreement: Final Report. 2002. 66p.
20. Grygoryan R.D. Problem-oriented computer simulators for solving of theoretical and applied tasks of human physiology. Problems of programming. 2017, №3, 102-111.
21. Grygoryan R.D. Modeling of mechanisms providing the overall control of human circulation. Advances in Human Physiology Research, 2022,4, 5 – 21, https://doi.org/10.30564/ahpr.v4i1.4763.
22. Grygoryan R.D. Problems associated with creating special software for simulating of human physiological responses to dynamic ±Gz accelerations. Проблеми програмування, 2024; 1: 30-37, http://doi.org/10.15407/pp2024.01.30.
23. Grygoryan R.D., Degoda A.G. A mathematical model of human hemodynamics for use in special software simulating pilots' physiological responses to sustained ±Gz accelerations. 2024; 90: 4-15, DOI: 10.5281/zenodo.11357990.

Отримано: 25.02.2025
Внутрішня рецензія отримана: 06.03.2025
Зовнішня рецензія отримана: 09.03.2025

About authors:

Grygoryan Rafik,
Department chief, PhD,
D-r in biology
http://orcid.org/0000-0001-8762-733X.

Degoda Anna,
Senior scientist, PhD.
http://orcid.org/0000-0001-6364-5568.

Progonnyi Mykola,
Scientist
https://orcid.org/0000-0002-8320-3465

Place of work:

Institute of software systems of National Academy of Sciences of Ukraine, 03187, Kyïv, Acad. Glushkov avenue, 40,
E-mail:
rgrygoryan@gmail.com,
anna@silverlinecrm.com,
progonny@gmail.com

УДК 004.91 http://doi.org/10.15407/pp2025.01.024

Б.О. Семьонов, С.Д. Погорілий

ДОСЛІДЖЕННЯ ЗАСТОСУВАННЯ ТЕХНОЛОГІЙ GPGPU ТА TPU ДЛЯ МЕТОДУ ЗАБЕЗПЕЧЕННЯ ЯКОСТІ КОМЕНТАРІВ У СИСТЕМАХ КОНТРОЛЮ ВЕРСІЙ

У роботі обґрунтовано актуальність вирішення задачі забезпечення якості описів до внесених змін у вихідних текстах програм для систем контролю версій. Для здійснення фільтрації коментарів використовуються методи машинного навчання: нейронні мережі різної архітектури. Доцільним є використання нейронних мереж у зв'язку з необхідністю пошуку описів до внесених змін, які відображають їхню мету. Створено рекурентні нейронні мережі та здійснено їх навчання на множині описів до внесених змін, отриманих за допомогою спеціального програмного інтерфейсу GitHub REST API. Для покращення швидкодії навчання застосовано різні програмно-апаратні платформи на кшталт CPU, TPU та GPGPU. Проведено аналіз точності моделей за допомогою метрик: точності (Accuracy) та середнього гармонійного (F1-score).

Ключові слова: GPGPU, RNN, TPU, опис внесених змін, репозиторій, система контролю версій.

B.O. Semonov, S.D. Pogorilyy

RESEARCH OF THE APPLICATION OF GPGPU AND TPU TECHNOLOGIES FOR ENSURING COMMENT QUALITY IN VERSION CONTROL SYSTEMS

The study substantiates the relevance of solving the issue of ensuring the quality of descriptions for changes made in source code files within version control systems. Machine learning methods, particularly neural networks of various architectures, are employed for comment filtering. Neural networks are deemed appropriate due to the necessity of identifying descriptions that accurately reflect the purpose of the changes made. Recurrent neural networks were developed and trained on a dataset of change descriptions obtained through the GitHub REST API. To enhance training performance, various hardware and software platforms such as CPU, TPU, and GPGPU were utilized. The accuracy of the models was analyzed using metrics like Accuracy and the harmonic mean (F1-score).

Keywords: commit message, GPGPU, repository, RNN, TPU, version control system.

Вступ

У сучасному цифровому світі, де розвиток програмного забезпечення стає невід'ємною частиною багатьох галузей, системи контролю версій є ключовим інструментом для ефективної розробки та управління вихідними текстами програм. Постійні зміни ринку вимагають від розробників не лише швидкості та якості, а й систематизованого та надійного підходу до керування версіями застосунків [7].

Завдяки системам контролю версій розробники можуть ефективно працювати над проєктами, вносити зміни та експериментувати з новими функціями, водночас маючи можливість у разі необхідності повертатися до попередніх версій. Це дозволяє уникнути втрат даних, забезпечує стабільність роботи системи та сприяє кращій співпраці в команді розробників.

Опис внесених змін у системі контролю версій є важливим елементом для розуміння, відстеження та збереження історії змін у кодовій базі. Кожне повідомлення про внесені зміни є відображенням конкретної зміни, внесеної розробником. Деталізований та розгорнутий опис внесених змін допомагає не лише поточним, а й майбутнім членам команди легко розібратися в тому, як і чому змінено вихідний текст програми [3].

Зрештою деталізовані описи внесених змін роблять проєкт більш документованим та полегшують процеси технічного супроводу, допомагають відслідковувати

етапи його розвитку, логіку ухилення рішень та, кінець кінцем, роблять проєкт прозорішим.

Створення методу забезпечення якості коментарів до внесених змін у вихідних текстах програм для систем контролю версій є актуальним у зв'язку із зростанням обсягу проєктів та ускладненням структури кодової бази.

Враховуючи те, що до сучасних проєктів можуть залучатися розробники із різним досвідом та стилями програмування, фільтр описів внесених змін, який оцінюватиме зміст та контекст опису, буде сприяти розумінню іншими членами команди внесених у вихідний текст програм змін та буде спрощувати технічну підтримку.

Отже, опис внесених змін – це повідомлення, зміст якого відповідає загальним правилам написання повідомлень про внесені зміни, є лаконічним, описує характер змін, їхні ефект та причину.

1. Правила щодо оформлення повідомлення про внесені зміни [6]:
   а. опис повинен мати заголовок і може мати тіло. Ці частини мають бути розділені порожнім рядком;
   б. заголовок не має бути довгим, не більше 50-70 символів;
   в. дієслова у заголовку повинні бути доконаного виду.
2. Правила щодо подання інформації в повідомленні про внесені зміни:
   а. має бути описана причина зроблених змін;
   б. який ефект має це повідомлення про внесені зміни;
   в. якщо внесені зміни виправляють якусь проблему, то необхідно її вказати;
   г. подання інформації має бути таким, щоб фахівець, який не розуміється на проблемі чи структурі коду, збагнув, що було зроблено і який вплив це має на проєкт (програму) Приклади описів наведено в Таблиці 1.

Постановка задачі

Всі, наведені у попередньому розділі правила написання повідомлень про внесені зміни, можна опрацювати засобами будь-якої мови програмування. Проте для правила 2.г потрібно використовувати саме машинне навчання, бо формальних правил для визначення, що було зроблено і чому, не існує – для цього потрібне «людське» розуміння контексту та змісту повідомлення про внесені зміни. Звідси випливає постановка задачі: створити метод забезпечення якості повідомлень про внесені зміни вихідних текстів програм у системах контролю версій, який аналізуватиме опис внесених розробником змін у вихідний текст програми та повертатиме мітку, чи відповідає цей коментар до внесених змін на питання «що було зроблено і чому?», тобто «так» чи «ні».

Таблиця 1

Повідомлення про внесені зміни, які відповідають правилу 2.г

№	Приклад опису
1	Add array identifiers for generating routes. This allows for resources with multi-column keys to generate the related routes by only passing an instance of the object in question.
2	Increases the select timeout in start reactor() endless loop. This small patch greatly reduces CPU time (for instance for backgroundrb).
3	Add a ’variance’ method and a ’sum’ method to the Array class to stay DRY.
4	Fixed autocomplete with scrollbar. IE has problems when the ul has a scrollbar and the user clicks there. The activate event is thrown and the list disappears.

Формалізація задачі фільтра. Нехай **X** – множина описів внесених змін (генеральна сукупність), кожне із цих повідомлень про внесені зміни описується m-вимірним вектором слів:

$$\vec{x} = \{x_1, x_2, \ldots, x_m\}, \vec{x} \in \mathbf{X}, \quad (1)$$

де m – розмірність вектору слів.

Y – множина можливих відгуків (міток) у вигляді M-вимірного вектору:

$$\vec{y} = \{y_1, \ldots, y_M\}, \vec{y} \in \mathbf{Y}, \quad (2)$$

де M = 2 – це кількість відгуків (класів), які потрібно отримати: перший відгук – чи задовольняє поточне повідомлення вимоги, другий – не задовольняє. Звідси випливає, що шуканий фільтр – це задача бінарної класифікації: Y = {0, 1}.

Отже, модель фільтра повідомлень до внесених змін можна описати таким сюр'єктивним, але не ін'єктивним, відображенням:

$$f : \mathbf{X} \to \mathbf{Y} \quad (3)$$

Збір та підготовка навчальних корпусів повідомлень

Для збору даних для навчання було використано один із найбільших веб-сервісів для розміщення IT-проєктів та спільної розробки програмного забезпечення GitHub, в основі якого лежить популярна система контролю версій Git. Цей веб-сервіс має спеціальний програмний інтерфейс для застосунків, який називається GitHub REST API [4]. Через те, що дослідження пов’язане з перевіркою змісту повідомлень (commit messages) про внесені зміни (differences, скорочено «diffs») для систем контролю версій, знадобилися такі інтерфейси:

1. отримання переліку репозиторіїв (проєктів);
2. отримання повідомлень до внесених змін для кожного репозиторію.

Наведені вище інтерфейси мають тип GET http-запиту і повинні формувати та передавати на сервер відповідні GET-параметри та http-заголовки. У свою чергу відповідь від сервісів надходить у текстовому форматі структурованих даних JSON.

Вказані у Таблиці 2 http-запити мають однакові набори http-заголовків, а саме:

1. 'Accept': 'application/vnd.github+json' – тип відповіді від програмного інтерфейсу GitHub REST API;
2. 'Authorization': 'Bearer [token]' – авторизація користувача API,

   де token – спеціальний набір символів, який можна згенерувати в налаштуваннях користувача вебсервісу GitHub;
3. 'X-GitHub-Api-Version': '2022-11-28' – версія програмного інтерфейсу GitHub REST API.

Після завантаження було виконано ручне маркування цих даних, тобто віднесення кожного повідомлення до відповідного класу згідно з правилом 2.г. Такий процес довготривалий, тому в цій роботі використовується 8 тис. повідомлень з 1 млн. завантажених.

Аналіз промаркованих описів внесених змін показав, що дані є нерівномірними. В отриманій вибірці описи, які не відповідають правилу 2.г, кількісно переважають повідомлення, які відповідають вимогам. На Рис. 1 зображено розподіл навчальних корпусів повідомлень про внесені зміни. Такий розподіл впливає на точність навчання нейронних моделей, тому наступний методи було використано задля боротьби з вказаною проблемою: зсув порогу (threshold moving) з вибором коефіцієнту 0.57.

Перед подачею повідомлень про внесені зміни на вхід досліджуваних нейронних мереж було виконано [11]:

1. нормалізацію (стандартизацію) повідомлень, а саме: приведення літер тексту у малі;
2. токенізацію повідомлень на навчальній вибірці;
3. векторизацію повідомлень. Під час створення об'єкту векторизації повідомлень було зафіксовано довжину вихідної послідовності токенів, тому що форма вхідного для нейронної мережі тензору має бути фіксованою. А також обмежено розмір словника, чим нехтують найменш вживані токени для прискорення обчислень.

Таблиця 2

Особливості сервісів програмного інтерфейсу GitHub REST API

№	Назва сервісу	URL-посилання	GET-параметри	«Корисні» поля для дослідження
1	Отримання переліку репозиторіїв	https://api.github.com/repositories	**since** – параметр, значення якого повинне бути цілим числом та відповідає ідентифікатору репозиторія. Він не є обов'язковим. Якщо цей параметр присутній, то у відповіді буде повернуто перелік репозиторіїв, ідентифікатори яких більші за вказаний у параметрі.	**id** – ідентифікатор репозиторію; **full_name** – назва репозиторію; **description** – детальний опис репозиторію.
2	Отримання повідомлень до внесених змін для кожного репозиторію	https://api.github.com/repos/[full_name]/commits?per_page=100, де full_name – назва репозиторію з пункту № 1	**per_page** – кількість повідомлень про внесені зміни у відповіді на один запит. Необов'язковий параметр. За замовчуванням встановлено 30 повідомлень. **sha** – символьна послідовність (відбиток) повідомлення про внесені зміни, з якого потрібно починати пошук.	**sha** – символьна послідовність (відбиток) повідомлення про внесені зміни; **message** – повідомлення про внесені зміни. Знаходиться в JSON-об'єкті «commit»; **date** – дата та час, коли були внесені зміни до репозиторію. Знаходиться в JSON-об'єкті «author», який, у свою чергу, розміщений у JSON-об'єкті «commit». **sha** з JSON-масиву **parents** – символьна послідовність (відбиток) повідомлення про внесені зміни попередніх внесених змін.

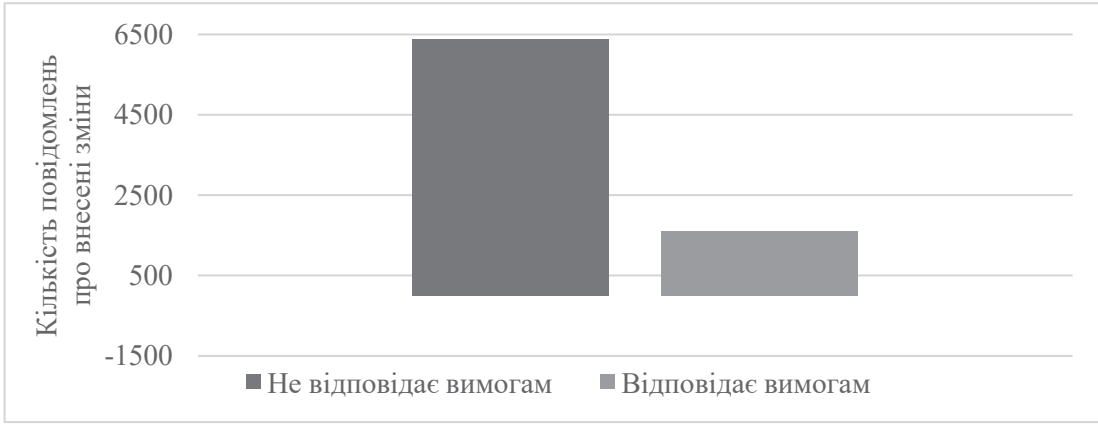


Рис. 1. Розподіл повідомлень про внесені зміни

Метод забезпечення якості коментарів до внесених змін у системах контролю версій

Перший фільтр (Рис. 2) було побудовано на основі моделі з використанням повнозв'язного (Dense) шару, який складається з:

1. двох вагових шарів (weighted layers):
   а. вхідного (Embedding) шару;
   б. вихідного повнозв'язного шару (Dense) [14] з кількістю нейронів, яка дорівнює кількості бажаних відгуків (M = 2);
2. проміжного шару без ваг (Global Average Pooling) [14].

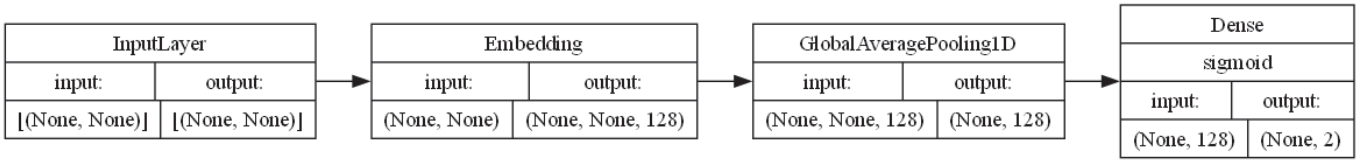


Рис. 2. Архітектура моделі з повнозв'язним шаром без методів регуляризації

Як алгоритм оптимізації ваг обрано AdamW [9], що так само є алгоритмом Adam зі спеціальним варіантом регуляризації на основі пониження ваг (weight decay). Обраний алгоритм демонструє швидше навчання і краще узагальнення, основними кроками якого є:

1. ініціалізація параметрів:
   а. встановлення початкових значень для параметрів моделі θ_0 (вектор ваг) випадковим чином;
   б. ініціалізація першого моменту (оцінка середнього градієнта) $m_0 = 0$ та другого моменту (оцінка дисперсії градієнта) $v_0 = 0$;
   в. встановлення гіперпараметрів:
   η – швидкість навчання (learning rate);
   β_1 і β_2 – коефіцієнти згасання для першого та другого моментів;
   ε – деяке мале значення для запобігання діленню на нуль;
   λ – коефіцієнт вагового спаду (weight decay);
2. обчислення градієнта: для кожної ітерації t обчислюється градієнт функції втрат:
$$g_t = \nabla_\theta f(\theta_t), \quad (4)$$
де $f(\theta)$ – це функція втрат,
g_t – градієнт для поточних параметрів;
3. оновлення моментів:
   а. оновлення першого моменту (оцінка середнього градієнта):
$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \quad (5)$$
   б. оновлення другого моменту (оцінка дисперсії градієнта):
$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \quad (6)$$
4. коригування зміщення моментів: для того, щоб компенсувати зміщення на початкових етапах (оскільки $m_0 = 0$ та $v_0 = 0$), відбувається коригування:
$$m_t = \frac{m_t}{1-\beta_1^t}, \quad (7)$$
$$v_t = \frac{v_t}{1-\beta_2^t} \quad (8)$$
5. оновлення параметрів з урахуванням вагового спаду та адаптивної швидкості навчання:
$$\theta_t = \theta_{t-1} - \eta\left(\frac{m_t}{\sqrt{v_t + \varepsilon}} + \lambda\theta_{t-1}\right), \quad (9)$$
де ваговий спад $\lambda\theta_{t-1}$ застосовується окремо від основного процесу оновлення градієнта, що робить AdamW відмінним від класичного Adam;
6. повторення кроків 2 – 5 для кожної ітерації t, поки не буде досягнуто зупинки (залежно від кількості епох або критеріїв зупинки).

У моделі другого фільтру описів внесених змін (Рис. 3) для зменшення прояву явища перенавчання використано методи регуляризації:

1. введено додатковий шар Dropout з коефіцієнтом 0.3 – випадкове виключення нейронів шару із заданою

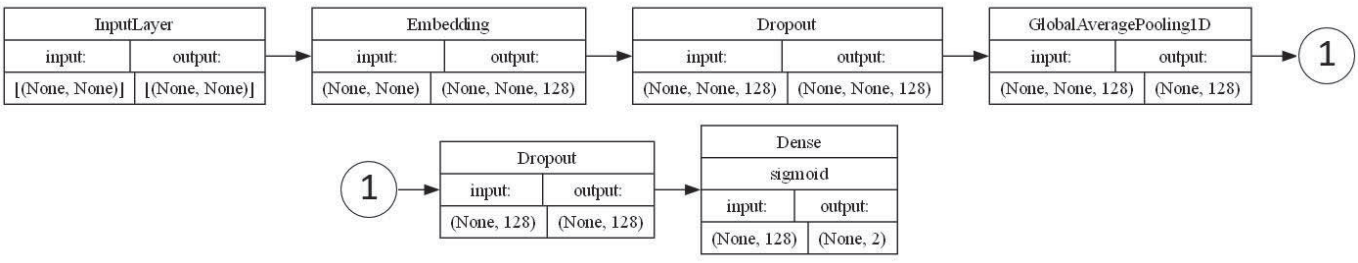


Рис. 3. Архітектура моделі з повнозв'язним шаром та методами регуляризації

ймовірністю p під час кожної ітерації навчання, метою якого є перешкоджання спільній адаптації ваг нейронів шару, яка у свою чергу призводить до перенавчання [10];
2. використано метод ранньої зупинки.

Окрім того, до частини, яка видобуває ознаки тексту для подальшої класифікації, було додано різні варіанти рекурентних шарів, в результаті чого вдалося побудувати 5 різних моделей:

1. один LSTM-шар з 48-ма нейронами (Рис. 4);
2. один двонаправлений (bidirectional) LSTM/GRU-шар з 24-ма нейронами (загалом 48 нейронів) (Рис. 5);
3. два двонаправлені (bidirectional) LSTM/GRU-шари з 12-ма нейронами кожний шар (загалом 48 нейронів) (Рис. 6).

Слід зауважити, що для навчання моделей було використано методологію, яка базується на основі перевірної підмножини. Вона полягає у тому, що весь доступний набір маркованих даних поділяється на частини, що не перетинаються, а саме: з навчальної множини (training set) було виділено 20% на перевірну підмножину (validation subset), а решту – для навчання моделей-кандидатів.

Для імплементації наведених вище моделей було використано мову програмування Python (версії 3.10.12) та застосовано бібліотеку TensorFlow (2.15.0), тому що вона має підтримку GPGPU та TPU технологій, які є однією зі складових дослідження. У Таблиці 3 наведено порівняльну характеристику між TensorFlow та іншими популярними бібліотеками для машинного навчання, такими як PyTorch [12], Keras [8] та Scikit-learn [13]. Кожна з цих бібліотек має свої унікальні можливості та підходить для різних типів завдань.

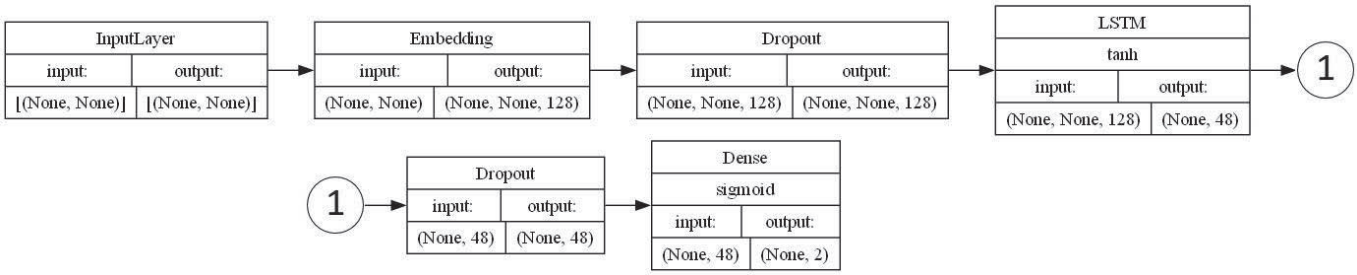


Рис. 4. Архітектура моделі з рекурентним LSTM-шаром та методами регуляризації

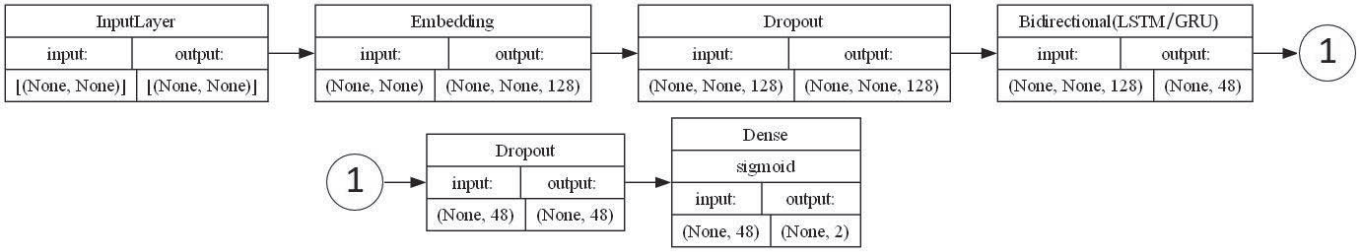


Рис. 5. Архітектура моделі з одним рекурентним BiLSTM/BiGRU-шаром та методами регуляризації

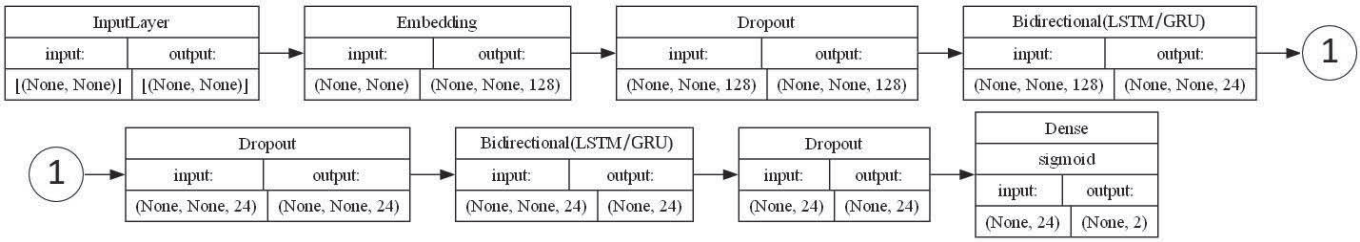


Рис. 6. Архітектура моделі з двома рекурентними BiLSTM/BiGRU-шарами та методами регуляризації

Таблиця 3

Особливості бібліотек для машинного навчання

Характеристика	**TensorFlow**	**PyTorch**	**Keras**	**Scikit-learn**
Рік випуску	2015	2016	2015	2007
Розробник	Google	Facebook (Meta)	Google (підмодуль TensorFlow)	Інститут Франсуа Шоле
Мова програмування	Python, C++, Java, Go, JavaScript, Swift	Python, C++	Python	Python, C++
Основне використання	Глибоке навчання, нейронні мережі	Глибоке навчання, нейронні мережі	Інтерфейс для TensorFlow/високорівневе API	Класичні ML-алгоритми, регресія, класифікація
Підтримка CPU/GPU	Так (підтримує GPU через CUDA, TPU)	Так (GPU через CUDA)	Так (через TensorFlow або Theano)	Лише CPU (GPU підтримується через обгортки)
Модульність	Висока (TensorFlow 2.0 спрощений)	Висока (динамічні обчислювальні графи)	Висока (зручний API поверх TensorFlow)	Середня (для традиційних ML-моделей)
Динамічний граф	Ні (але є функція Eager Execution у TF 2.0)	Так (основа PyTorch)	Ні (покладається на TensorFlow)	Ні
Простота використання	Відносно складна, але зростає з TF 2.0	Простий для прототипування, інтуїтивний	Дуже простий (високий рівень абстракції)	Дуже простий (для класичних задач ML)
Гнучкість	Висока (низький рівень контролю)	Висока (простий контроль над обчисленнями)	Обмежена (високий рівень абстракції)	Середня (зосереджена на класичних алгоритмах)
Обчислювальні графи	Статичні (але є динамічна можливість)	Динамічні (створюються під час виконання)	Статичні (через TensorFlow або Theano)	Ні
Розподілені обчислення	Так, відмінна підтримка (TPU, кластеризація)	Обмежена підтримка	Через TensorFlow	Ні
Підтримка мобільних пристроїв	Так (TensorFlow Lite)	Ні	Так (через TensorFlow Lite)	Ні
Експорт моделей	TensorFlow SavedModel, HDF5, TF.js, TF Lite	TorchScript, ONNX	HDF5, TensorFlow SavedModel	Pickle, ONNX
Спільнота та ресурси	Велика, багато офіційної документації, підтримка від Google	Велика, активна спільнота, багато прикладів	Велика спільнота через TensorFlow/Keras	Дуже велика для класичного ML
Продуктивність	Висока продуктивність, особливо на великих моделях	Висока продуктивність на GPU	Висока (через TensorFlow)	Висока продуктивність для традиційного ML
Використання в індустрії	Широко використовується, особливо у великих проєктах (Google, Uber, Airbnb)	Використовується для наукових досліджень і прототипування	Широко використовується для швидкої розробки	Дуже популярний у наукових дослідженнях і стартапах

Як середовище виконання було обрано *Google Colab* [5], тому що воно надає користувачеві (досліднику, розробнику) «в хмарі» апаратні ресурси технологій GPGPU та TPU, а також в ньому можна використовувати всі наведені бібліотеки для машинного навчання.

Під час використання платформи *Google Colab* було визначено наступні переваги:

1. Доступ до потужних GPU і TPU. Як відомо, графічні та тензорні процесори значно прискорюють тренування моделей глибокого навчання. *Google Colab* дозволяє використовувати ці обчислювальні потужності, що важливо для проєктів, які потребують великих ресурсів;
2. Це дає змогу тренувати моделі, які потребують багато обчислень, без необхідності купувати дорогий апаратний ресурс;
3. Хмарне середовище: *Google Colab* працює у браузері, тому користувачеві не потрібно нічого встановлювати локально на свій комп'ютер. Всі обчислення виконуються на серверах *Google*, а дані та написані сценарії автоматично зберігаються на хмарному носії *Google Drive*, що полегшує збереження та доступ до проєктів;
4. Інтеграція з *GitHub*: ця можливість дозволяє відкривати, редагувати та зберігати вихідні тексти програм безпосередньо в репозиторіях *GitHub*;
5. Легкий обмін та співпраця: користувачі можуть ділитися своїми сценаріями через посилання, як це відбувається з *Google Docs* або *Google Sheets*. Інші ж користувачі можуть редагувати або переглядати ваші зміни вихідних текстів у режимі реального часу. Це зручно для командної роботи, наукових досліджень або навчання, коли існує необхідність працювати над одним проєктом одночасно з кількома користувачами;
6. Підтримка основних бібліотек для машинного навчання. *Google Colab* має попередньо встановлені популярні бібліотеки, такі як *TensorFlow*, *Keras*, *PyTorch*, *Scikit-learn*, *Pandas*, *NumPy* та інші. Це дозволяє швидко почати роботу, не гаючи часу на встановлення пакетів;
7. Інтерактивне *Python*-середовище. *Colab* використовує *Jupyter Notebook* у хмарі, що дозволяє писати *Python*-сценарії, виконувати його по блоках, а також додавати візуалізації, текстові блоки та графіки. Це інтерактивне середовище ідеально підходить для експериментів з даними та розробки моделей машинного навчання, наукових досліджень і навчання;
8. Підтримка інших мов програмування. Хоча основною мовою в *Google Colab* є *Python*, також існує можливість виконувати сценарії, написані іншими мовами, за допомогою спеціальних команд. Наприклад, R, JavaScript, SQL тощо;
9. Розширення для візуалізації: Colab підтримує бібліотеки для візуалізації, такі як Matplotlib, Seaborn, Plotly та інші, що дозволяє створювати графіки, діаграми та візуалізації результатів безпосередньо в Jupyter-блокнотах;
10. Безперервне навчання моделей, тобто існує можливість виконувати процеси навчання моделей протягом кількох годин на серверах Google Colab, що зручно для довгих тренувань, особливо у роботі з великими нейронними мережами

Аналіз архітектур паралельних обчислень та їх застосування для навчання моделей фільтра

Основною особливістю сучасних графічних відеоадаптерів, які використовують графічні процесори (GPU), є наявність набору потокових мультипроцесорів (SM), що використовувалися раніше лише в алгоритмах і задачах, пов'язаних з обробкою графічних зображень. Програмні технології (інтерфейси), що застосовуються розробни-

ками програмного забезпечення для створення програм такого напрямку, використовують пам'ять відеоадаптера для розміщення структур даних, таких як текстури, буфери, визначають конвеєр обробки, кожен етап якого відповідає за свої специфічні дії, а саме: растеризацію, інтерполяцію, блендинг, теселяцію [2] тощо.

Технологія обчислень загального призначення на графічних процесорах (GPGPU) ґрунтується на використанні великої кількості процесорів GPU, що працюють паралельно, для обробки даних за допомогою алгоритмів загального призначення (наукових чи інших, але не обов'язково пов'язаних з обробкою зображень).

Потоковий процесор на GPU має простішу структуру порівняно з обчислювальним вузлом CPU. Тобто такі вузли менш універсальні і виконують менший набір функцій, аніж вузли процесора. Проте, оскільки їхня кількість велика, то для певних типів задач можна значно підвищити швидкодію. Найкращого прискорення вдається досягти для алгоритмів, що підтримують концепцію паралелізму за даними (один паралельний потік обробляє свою область у пам'яті). Тому зазвичай GPGPU застосовують у наступних сферах: обробка зображень (Reduction, Histogram, Fast Fourier Transform, Summed Area Table); обробка відеоданих (Transcode, Digital Effects, Analysis); лінійна алгебра; моделювання (Technical, Finance, Academic, Some Databases) [1] тощо.

TPU (Tensor Processing Unit) – це спеціалізований інтегральний процесор, розроблений компанією Google, який оптимізований для виконання операцій глибокого навчання, зокрема, обчислень з тензорами в рамках машинного навчання. Він використовується для прискорення роботи моделей штучного інтелекту, особливо в системах, що працюють з великими обсягами даних і потребують високої продуктивності.

У Таблиці 4 наведено порівняльну характеристику технологій GPGPU та TPU:

Отже, основна відмінність полягає в тому, що TPU розроблені спеціально для швидкої обробки великих обсягів даних, які використовуються в машинному навчанні (зокрема, нейронних мережах), тоді як GPU є універсальнішими і застосовуються як у графічних, так і в обчислювальних задачах, включно із тренуванням моделей машинного навчання. Кожна з технологій має свої переваги, і вибір між ними визначається конкретними вимогами до продуктивності та характером обчислювальних задач.

Ефективність застосування технологій GPGPU та TPU до задачі забезпечення якості коментарів до внесених змін у вихідних текстах програм для систем контролю версій показано в Таблиці 5.

Результати цього експерименту показали, що найбільшого прискорення навчання моделей вдалося досягти саме з використанням GPU.

Оцінка методу забезпечення якості коментарів

Для оцінки якості забезпечення коментарів до внесених змін систем контролю версій використано такі метрики [13]:

1. частка правильних відповідей (*Accuracy*):

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \quad (10)$$

де TP (True Positive) – вірні позитивні відповіді;

TN (True Negative) – вірні негативні відповіді;

FP (False Positive) – невірні позитивні відповіді;

FN (False Negative) – невірні негативні відповіді.

2. частка вірних позитивних відповідей серед усіх позитивних відповідей класифікатора (*Precision*):

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (11)$$

3. частка вірних спрацювань на позитивних об'єктах (*Recall*):

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (12)$$

4. середнє гармонійне (F1-score):

$$\text{F}_1 = 2\frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (13)$$

Таблиця 4

Порівняльна характеристика технологій GPGPU та TPU

Характеристика	TPU	GPU
Призначення	Оптимізовано для виконання операцій з тензорами, орієнтованих на обчислення в нейронних мережах (машинне навчання, зокрема TensorFlow)	Широко застосовується для паралельних обчислень, графіки, рендерингу і тренування нейронних мереж
Виробник	Google	Nvidia, AMD
Архітектура	Спрямована на матричні операції (використовує матричні множники)	Паралельна обробка, тисячі ядер для обчислень
Тип обчислень	Високоефективні для операцій з низькою точністю (FP16, INT8)	Гнучка точність обчислень (FP32, FP64), підходить для широкого спектру обчислень
Програмне забезпечення	Спеціалізоване для TensorFlow, Google Cloud	Підтримує TensorFlow, PyTorch, Keras та інші фреймворки
Швидкість обробки	Висока продуктивність при тренуванні нейронних мереж	Залежить від моделі, проте найсучасніші GPU мають високу швидкість, але відстають від TPU у певних задачах
Енергоспоживання	Оптимізовані для низького енергоспоживання при виконанні ML задач	Високе енергоспоживання, особливо у потужних моделях для машинного навчання
Ціна	Доступний лише як хмарна послуга Google (не доступний для індивідуального продажу)	Можна придбати як окремі пристрої різного рівня продуктивності
Застосування	Виключно для машинного навчання, нейронних мереж	Широке застосування: графіка, симуляції, машинне навчання, рендеринг
Пам'ять (тип і розмір)	Спеціалізована швидка пам'ять для ML (HBM)	GDDR5, GDDR6, HBM, різні об'єми пам'яті
Гнучкість у використанні	Менш універсальні, спеціалізовані для ML	Висока універсальність, використовується в багатьох сферах

Таблиця 5

Порівняльна таблиця часу виконання навчання моделей за допомогою CPU, TPU та GPU

Модель	**CPU (Intel Xeon з 2-ма ядрами)**	**TPU другого покоління з 8-ма ядрами**	**GPU (Nvidia Tesla T4 з 320-ма тензорними ядрами)**
Dense (без методів регуляризації)	44 хв 7 с	9 хв 47 с	2 хв 26 с
Dense (з регуляризацією)	57 хв 52 с	7 хв 33 с	1 хв 55 с
LSTM (48 нейронів)	59 хв 34 с	15 хв 56 с	1 хв 4 с
BiLSTM (1 шар з 24-ма нейронами)	1 год 10 хв 59 с	11 хв 38 с	1 хв 33 с
BiLSTM (2 шари з 12-ма нейронами кожний)	1 год 14 хв 34 с	16 хв 56 с	2 хв 35 с
BiGRU (1 шар з 24-ма нейронами)	1 год 3 хв 19 с	10 хв 36 с	1 хв 34 с
BiGRU (2 шари з 12-ма нейронами кожний)	1 год 5 хв 55 с	18 хв 53 с	2 хв 49 с

Хоча варто врахувати, що для фільтра повідомлень про внесені зміни важливіша саме повнота (*Recall*), тому що краще

відхилити повідомлення та порекомендувати розробнику доповнити його або перефразувати.

Отже, результати роботи створених моделей наведені у Таблиці 6.

Таблиця 6

Порівняння ефективності побудованих моделей для імплементації фільтра повідомлень про внесені зміни

Модель	**Метрика**	**Без використання методів, що долають проблему нерівномірних даних**	**Зсув порогу (threshold moving)**
Dense (без методів регуляризації)	Accuracy F1-score	0.786 0.622	0.788 0.616
Dense (з регуляризацією)	Accuracy F1-score	0.816 0.648	**0.817** 0.637
LSTM (48 нейронів)	Accuracy F1-score	0.801 0.445	0.801 0.445
BiLSTM (1 шар з 24-ма нейронами)	Accuracy F1-score	0.796 0.664	0.798 0.655
BiLSTM (2 шари з 12-ма нейронами кожний)	Accuracy F1-score	0.794 **0.677**	0.796 0.675
BiGRU (1 шар з 24-ма нейронами)	Accuracy F1-score	**0.808** **0.651**	0.811 0.629
BiGRU (2 шари з 12-ма нейронами кожний)	Accuracy F1-score	0.791 0.635	0.790 0.630

Динаміку навчання моделей представлено на Рис. 7-9, а саме зображено графіки значення точності та середнього гармонійного залежно від епохи.

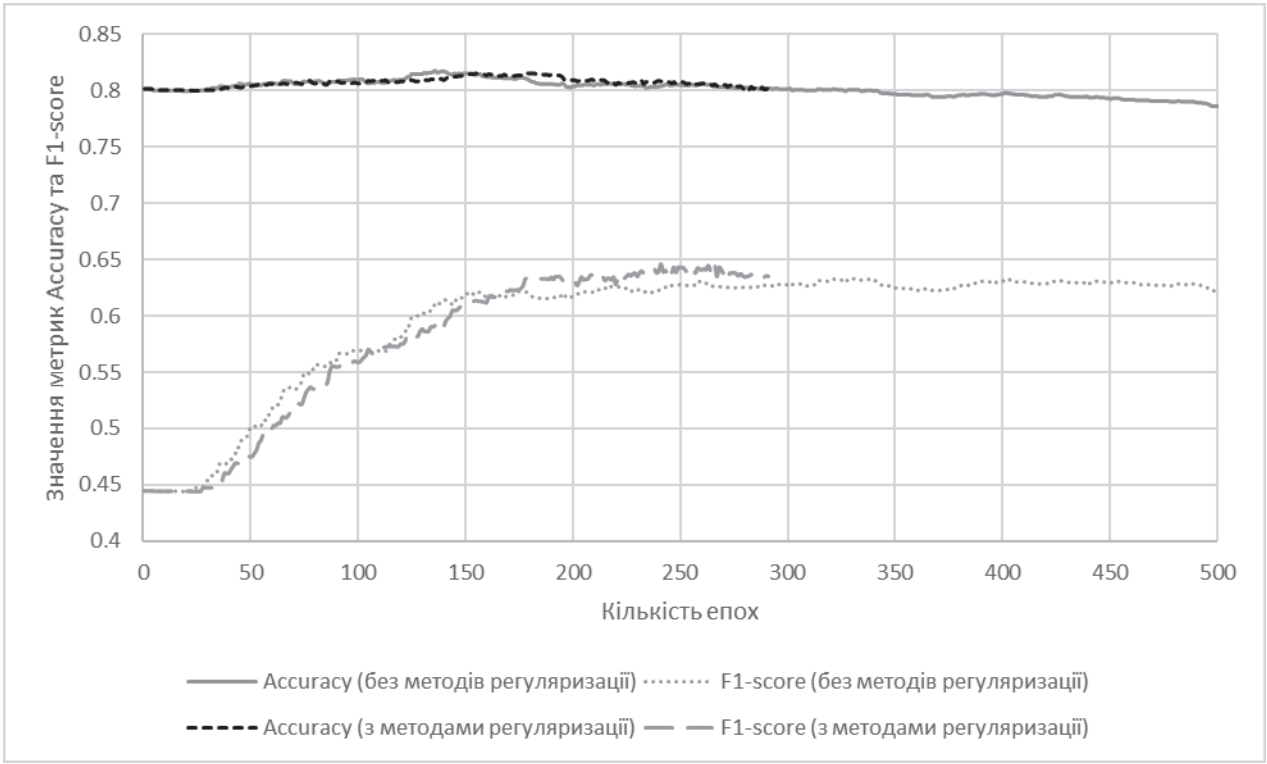


Рис. 7. Графік залежності значення різних метрик оцінювання якості навчання нейронних мереж з Dense-шаром від порядкового номера епохи навчання

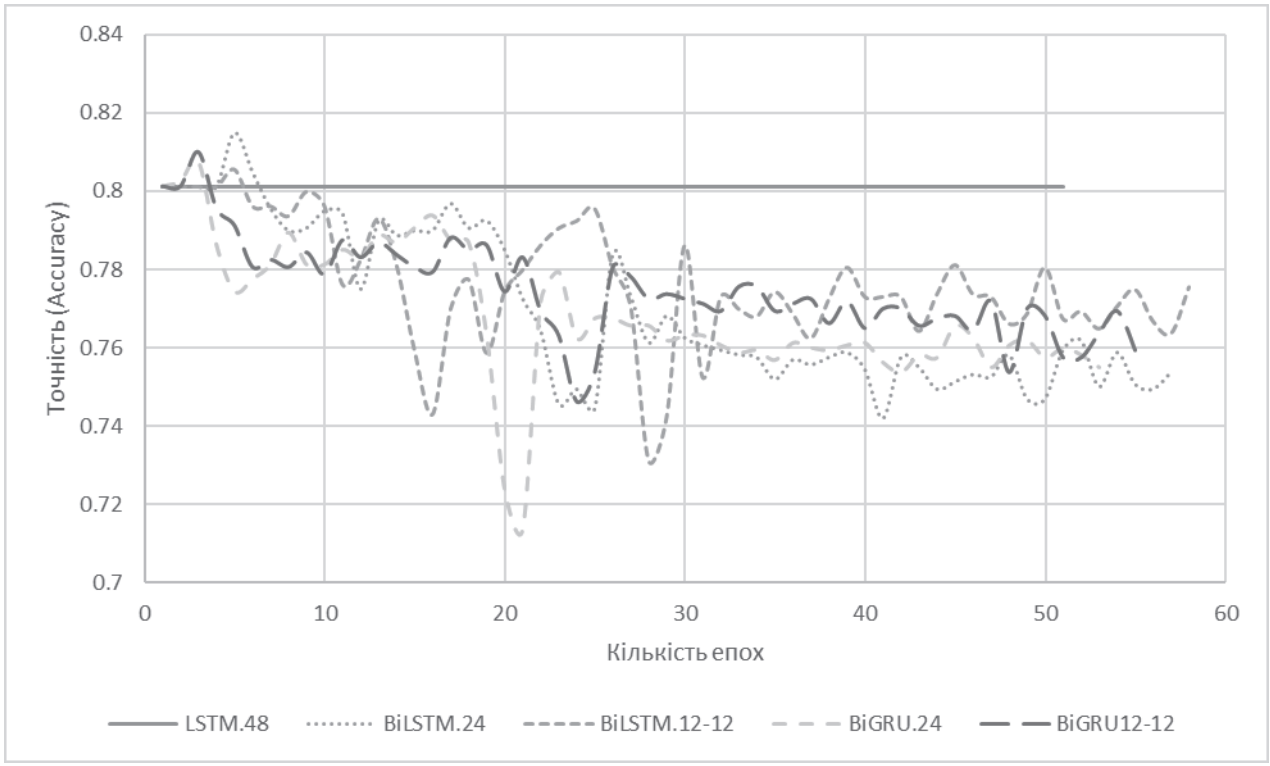


Рис. 8. Графік залежності значення точності рекурентних нейронних мереж від порядкового номера епохи навчання

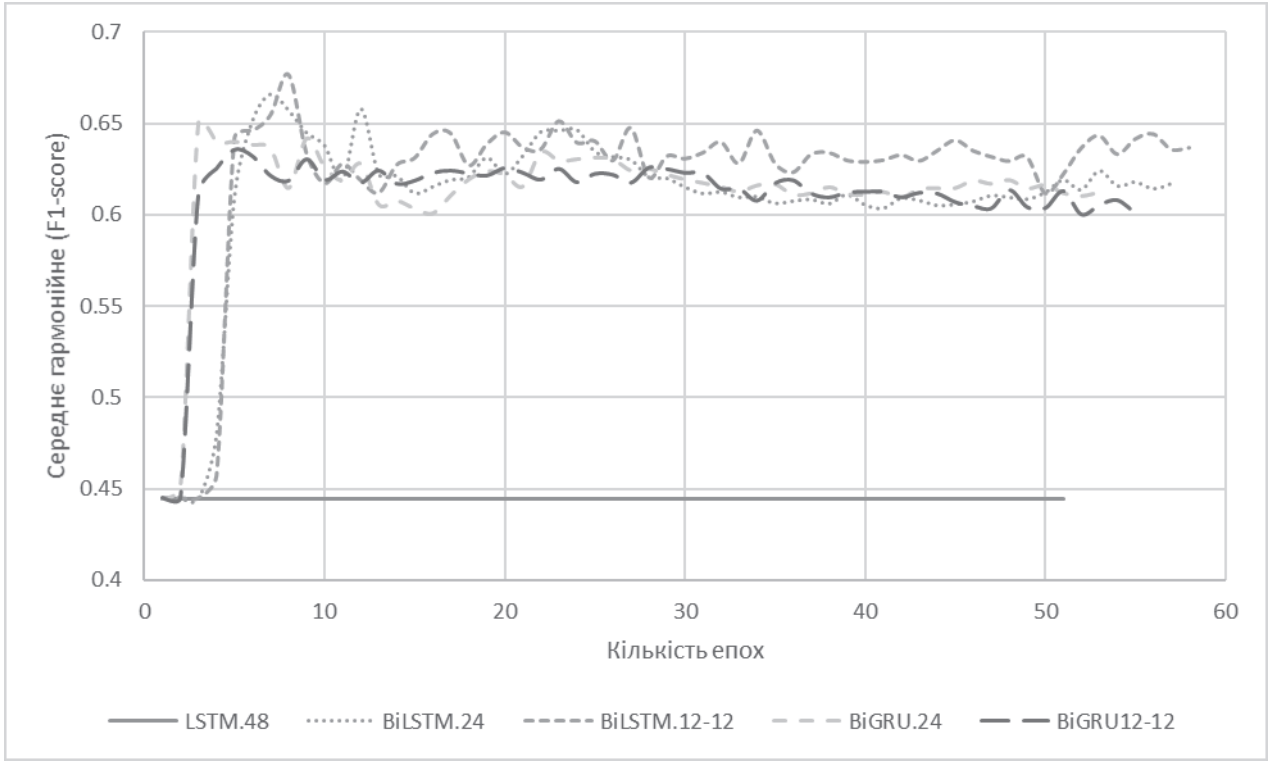


Рис. 9. Графік залежності значення метрики середнього гармонійного рекурентних нейронних мереж від порядкового номера епохи навчання

Графіки точності RNN-мереж мають дещо стрибкоподібний характер через особливості рекурентних нейронних мереж, тому що вони «працюють» із послідовними даними, і точність може різко змінюватися залежно від того, наскільки добре модель вловила послідовні зв'язки в певних наборах даних. Якщо модель починає «вчитися» на певних зразках і покращує точність для конкретних послідовностей, то вона може втрачати здатність узагальнювати інші зразки, що призводить до стрибків.

Висновки

У роботі запропоновано різні підходи до імплементації фільтра повідомлень до внесених змін. Цей фільтр є одним з етапів аналізу, обробки та підготовки даних для методу, який буде мати можливість створювати повідомлення про внесені зміни на їхній основі.

Проведено порівняльний аналіз основних програмно-апаратних платформ, запропонованих у хмарному середовищі Google Colab задля прискорення навчання моделей фільтра, а саме: CPU, TPU та GPU. Часові заміри тривалості навчання показують доцільність використання тензорного процесору (TPU) від корпорації Google та графічного адаптеру (GPU) від Nvidia у порівнянні з центральним процесором (CPU) від Intel. Результати показують домінування графічного адаптеру Nvidia Tesla T4 над Google TPU v2 у 15 разів, а над Intel Xeon – у 45 разів. У свою чергу TPU перевершує CPU у 6 разів.

Виконано оцінку методу забезпечення якості описів до внесених змін. З таблиці результатів видно, що найкращий результат має RNN-мережа, яка містить один двонаправлений (bidirectional) GRU-шар з 24-ма нейронами для видобутку ознак тексту: середнє гармонійне становить 65.1 %, а повнота – 80.8 %. Окрім того, така архітектура мережі навчається з методами регуляризації за 1 хвилину та 34 секунди за допомогою GPU. Хоча послідовна та інші рекурентні моделі (окрім, LSTM) також мають порівняні результати, як щодо точності, так і щодо швидкості.

Показано, що середовище *Google Colab* є потужним інструментом для швидкого прототипування та розробки моделей машинного навчання. Завдяки своїм можливостям (GPU, TPU та інтеграції з хмарними сервісами) він стає одним із найзручніших інструментів для студентів, дослідників та інженерів програмного забезпечення.

Література

1. Погорілий С.Д., Семьонов Б.О. Дослідження паралельних алгоритмів мовою Python з використанням різних платформ. *Наукові записки НаУКМА*. Вип. 198, 2017. С. 14–21.
2. Погорілий С.Д., Семьонов Б.О. Дослідження програмної бібліотеки Linpack на архітектурі CUDA мовою програмування Python. *Наукові праці ДонНТУ*. Вип. 23, № 2. С. 98–106.
3. Buse R., Weimer W. Automatically documenting program changes. *Proceedings of the IEEE/ACM International Conference on Automated Software Engineering*(2010).
4. GitHub Docs GitHub REST API. *GitHub Docs*. URL: https://docs.github.com/en/rest?apiVersion=2022-11-28 (дата звернення: 07.10.2024).
5. Google Colaboratory – Google. *research.google.com*. 2024. URL: https://research.google.com/colaboratory/faq.html (дата звернення: 07.10.2024).
6. How to Write a Git Commit Message. *cbeams*. 30.08.2014. URL: https://cbea.ms/git-commit/#seven-rules (дата звернення: 07.10.2024).
7. Jiang S., Armaly A., McMillan C. Automatically generating commit messages from diffs using neural machine translation. *Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering*(2017).
8. Keras Home - Keras Documentation. *Keras.io*. 2019. URL: https://keras.io/ (дата звернення: 07.10.2024).
9. Loshchilov I., Hutter F. Decoupled weight decay regularization. *Proceedings of the ICLR*(2019).
10. N.K. Nissa Text Messages Classification using LSTM, Bi-LSTM, and GRU. *Medium*. 23.08.2022. URL: https://nzlul.medium.com/theclassification-of-text-messages-using-lstm-bi-lstm-and-gruf79b207f90ad (дата звернення: 07.10.2024).
11. N.V. Otten How To Use Text Normalization Techniques In NLP With Python [9 Ways. *Spot Intelligence*. 25.01.2023. URL: https://spotintelligence.com/2023/01/25/text-normalization-techniquesnlp/ (дата звернення: 07.10.2024).
12. PyTorch PyTorch documentation — PyTorch master documentation. *Pytorch.org*. 2019. URL: https://pytorch.org/docs/stable/index.html (дата звернення: 07.10.2024).
13. Scikit-Learn scikit-learn: machine learning in Python — scikit-learn 0.16.1 documentation.

Scikit-learn.org. 2019. URL: https://scikit-learn.org/ (дата звернення: 07.10.2024).
14. TensorFlow TensorFlow. *TensorFlow*. 2019. URL: https://www.tensorflow.org/ (дата звернення: 07.10.2024).

Одержано: 26.12.2024
Внутрішня рецензія отримана: 08.01.2025
Зовнішня рецензія отримана: 08.01.2025

Про авторів:

[1]*Семьонов Богдан Олександрович*,
аспірант.
https://orcid.org/0009-0001-3692-9415.

[2]*Погорілий Сергій Дем'янович*,
доктор технічних наук, професор.
https://orcid.org/0000-0002-6497-5056.

Місце роботи авторів:

[1]Київський національний
університет імені Тараса Шевченка,
тел. +380 (66) 451-97-00
E-mail: bohdan.semonov@gmail.com

[2]Київський національний
університет імені Тараса Шевченка,
Тел. +380 (66) 434-27-86
E-mail: sdp7799@gmail.com

УДК 004.82 https://doi.org/10.15407/pp2025.01.038

Г.Ю. Проскудіна, К.О. Кудім

ОГЛЯД ГЛОБАЛЬНИХ СЛУЖБ АГРЕГАЦІЇ РЕСУРСІВ ВІДКРИТОГО ДОСТУПУ ТА ЇХНІХ ВИМОГ ДО ПОСТАЧАЛЬНИКІВ ДАНИХ

У роботі представлено огляд сучасних глобальних агрегаторів документів відкритого доступу. Проаналізовані їхні кількісні характеристики, такі як кількість зібраних описів документів та повних текстів, кількість постачальників даних, наявність інтерфейсу прикладного програмування для отримання даних. Проаналізовано склад і види їхніх постачальників даних, такі як інституційні репозитарії, відкриті журнали, видавництва, наукові репозитарії препринтів, тематичні електронні бібліотеки, а також системи, які в свою чергу теж є агрегаторами. Досліджено також яку саме інформацію про документи збирають ці агрегатори, як вона представлена в інтерфейсі користувача, а також яка інформація збирається про їхніх постачальників даних та яким чином вона представлена у інтерфейсі користувача. Як саме відбувається взаємодія агрегатора з постачальниками даних, які протоколи обміну даних підтримуються, з якою частотою відбувається оновлення зібраних даних. Також сучасні агрегатори на базі зібраних корпусів даних, використовуючи методи машинного навчання, методи бібліометрії, вебометрики, альтиметрії, семантометрії надають науковцям цілий ряд корисних сервісів. Ми як розробники низки наукових електронних бібліотек з відкритим доступом вже зареєстровані як провайдери даних у деяких з цих систем. Тому знайомі з їхніми вимогами у практичній площині. В цій роботі ми спробували дещо узагальнити ці вимоги.
Ключові слова: відкритий доступ, провайдер сервісів, провайдер даних, протокол OAI-PMH.

G.Yu. Proskudina, K.O. Kudim

OVERVIEW OF GLOBAL OPEN ACCESS RESOURCE AGGREGATION SERVICES AND THEIR REQUIREMENTS FOR DATA PROVIDERES

The paper presents an overview of modern global aggregators of open access documents. Their statistical characteristics are analysed, such as the number of collected document descriptions and full texts, the number of data providers, and the availability of application programming interface to obtain data. The types of data providers, such as institutional repositories, open journals, publishers, scientific repositories of preprints, thematic digital libraries, and systems that are also aggregators, are analysed. We also investigate what kind of information about documents these aggregators collect and how it is presented in the user interface, as well as what information is collected about their data providers and how it is presented in the user interface. How the aggregator interacts with data providers, what data exchange protocols are supported, and how often the collected data is updated. Also, modern aggregators based on collected data corpora, using machine learning methods, bibliometrics, webometrics, altmetrics, semantometrics, provide a range of useful services to researchers. As developers of a certain number of scientific digital libraries, we are already registered as data providers in some of these systems. Therefore, we are familiar with their requirements in the practical sense. In this paper, we have attempted to summarise these requirements.
Key words: open access, service provider, data provider, OAI-PMH protocol.

Вступ

В ході виконання частини проєкту НАНУ «Відкрита наука» були проведені дослідження із створення системи інтеграції (харвестера, агрегатора) ресурсів із різних відкритих академічних джерел з метою отримання зручного і потужного інструменту пошуку та доступу до інформації для користувачів. Також проводяться дослідження з метою інтеграції цих зібраних наукових ресурсів України у світовий/європейський науковий інформаційний простір.

Зазвичай, перед системою інтеграції постають три основні задачі:

1. Збір та інтеграція описової інформації (метаданих) з різних джерел електронних ресурсів;

2. Організація пошуку і видачі відповідних ресурсів;

3. Передача метаданих з власного харвестера іншим харвестерам.

Проаналізувавши низку програмних систем, на платформі яких можна розгорнути агрегатор наукових робіт, ми зупинилися на програмній системі VuFind, що була розроблена в університеті Вілланова, США, про яку ми розповіли в роботі [1].

Наразі ми вже розгорнули таку систему[1] і почали її експлуатацію. До неї підключено близько десяти електронних бібліотек (провайдерів даних), з яких було зібрано близько 300 тис метаданих документів, в основному це – статті. Ми індексуємо метадані документів усіх видів академічно релевантних ресурсів – таких як журнали, інституційні репозитарії, цифрові колекції тощо, які надають інтерфейс OAI та використовують протокол Open Archives Initiative Protocol for Metadata Harvesting[2] (OAI-PMH) для надання свого контенту [2]. Зібрані і проіндексовані дані зберігаються на серверах Інституту програмних систем НАНУ. Апробується підключення та передача метаданих з нашої системи до інших харвестерів. Вивчаються можливості системи VuFind із перегляду, пошуку та завантаження статей, а також відпрацьовуються інструкції для кінцевого користувача та постачальників даних, визначаються схеми метаданих основних видів наукових інформаційних ресурсів НАНУ.

Зараз продовжується робота із налагодження цієї системи інтеграції, відпрацьовуються алгоритми завантаження та оновлення даних, регламент роботи агрегатора, вимоги до провайдерів даних щодо підключення до нашого харвестра.

Тому вивчення світового досвіду з експлуатації таких агрегаторів ресурсів є необхідним кроком у доведенні нашої системи до сучасних зразків, аби зробити максимально доступною для суспільства наукову інформацію, що сприятиме розвитку освітньої, наукової, науково-технічної та інноваційної діяльності.

У першій частині ми даємо основні роз'яснення щодо значення термінів, які використовуються в наших документах.

У другій частині представлено огляд сучасних глобальних агрегаторів ресурсів відкритого доступу. Проаналізовані їхні кількісні характеристики, такі як кількість зібраних описів документів та повних текстів, кількість постачальників даних, наявність API отримання даних. Проаналізовано склад і види постачальників даних, такі як інституційні репозитарії, відкриті журнали, видавництва, наукові репозитарії препринтів, тематичні електронні бібліотеки, а також системи, які в свою чергу теж є агрегаторами. Досліджено також, яку саме інформацію про документи збирають ці агрегатори, як вона представлена в інтерфейсі користувача, а також яка інформація збирається про постачальників даних, а також, як вона представлена у інтерфейсі користувача. Яким чином відбувається взаємодія агрегатора з постачальниками даних, які протоколи обміну даних підтримуються, з якою частотою відбувається оновлення зібраних даних. Також сучасні агрегатори на базі зібраних корпусів даних, використовуючи методи машинного навчання, методи бібліометрії, вебометрики, альтиметрії, семантометрії надають цілий ряд корисних сервісів науковцям. Ми як розробники низки наукових електронних бібліотек з відкритим доступом вже зареєстровані як провайдери даних у деяких із цих систем. Тому знайомі з їхніми вимогами, як провайдерів сервісів до своїх провайдерів даних у практичній площині.

У третьому розділі ми спробували дещо узагальнити ці вимоги.

1. Термінологія

Цей розділ містить роз'яснення щодо значення термінів, які використовуються в нашій роботі.

Відкритий доступ (ВД). Будапештська ініціатива відкритого доступу (Budapest

1 https://harvester.nas.gov.ua/

2 https://www.openarchives.org/OAI/openarchivesprotocol.html

Open Access Initiative, BOAI), яке визначає ВД як «вільну доступність у загальнодоступному інтернеті, що дозволяє будь-яким користувачам читати, завантажувати, копіювати, поширювати, друкувати, шукати або посилатися на повні тексти цих статей, сканувати їх для індексації, передавати їх як дані програмному забезпеченню або використовувати їх для будь-яких інших законних цілей[3]».

Дублінське ядро. Схема метаданих для опису інформаційних ресурсів репозиторіїв провайдерів даних.

Електронна бібліотека (ЕБ). Це розподілена документальна інформаційно-пошукова система, що функціонує на основі повнотекстових репозиторіїв (баз даних) і надає можливість створювати, зберігати та використовувати різноманітні колекції електронних документів інформаційних ресурсів (документів) у глобальній мережі комп'ютерів у зручному для користувачів вигляді.

Запис. Сукупність метаданих, що описують окремий науковий ресурс, розташований у репозитарії. Цей термін зазвичай використовується для позначення опису цифрового об'єкта, такого як текст, зображення, відео тощо. У харвестері термін **запис метаданих** використовується для позначення метаданих наукової публікації, тобто її назву, авторів, анотацію, деталі фінансування проєкту тощо, і термін **повнотекстовий запис** для позначення запису, що містить посилання у метаданих на повний текст інформаційного ресурсу. У харвестері запис є інформаційним ресурсом.

Інформаційна (автоматизована) система. Організаційно-технічна система, в якій реалізується технологія зберігання та обробки інформації з використанням технічних і програмних засобів.

Інформаційний ресурс (ресурс). Електронний документ або їх сукупність у автоматизованих інформаційних системах (ЕБ, архівах, базах даних тощо).

Користувач. Фізична особа, якій надається відкритий доступ до пошуку та перегляду інформаційних ресурсів харвестера, що мають відношення до записів метаданих та похідної від них інформації (наприклад, статистична інформація), та виконання переходу на домашню сторінку знайденого документа у репозитарії провайдера даних.

Персональний електронний кабінет. Індивідуальна персоніфікована веб-сторінка, за допомогою якої користувач здійснює роботу з інформаційними ресурсами, представленими у харвестері.

Провайдер/постачальник даних. Це служба, що підтримує створення і ведення одного чи більше репозиторіїв (бази документів, архівів, електронних бібліотек), здійснює публікацію своїх ресурсів, а також надає доступ до своїх метаданих для їхнього використання в інших системах. Зазвичай, провайдер даних надає вільний доступ до своїх метаданих і, можливо, але не обов'язково, надає вільний доступ до повних текстів своїх документів з ЕБ чи з інших інформаційних ресурсів. Провайдер даних може мати самостійний веб-інтерфейс для організації пошуку, перегляду і доступу до своїх ресурсів, а також інші сервіси, що надаються кінцевим користувачам. Провайдер даних самостійно вирішує питання про відкритість своїх інформаційних ресурсів і доступність до них. Зокрема, провайдер даних може прийняти рішення про інтеграцію усіх або частини своїх інформаційних ресурсів на рівні метаданих у харвестері і для цього організує експорт відповідних метаданих у форматі узгодженого протоколу.

Протокол OAI-PMH. Протокол OAI-PMH збору (харвестінгу) визначає механізм збору записів, що містять метадані інформаційних ресурсів, з репозитаріїв провайдерів даних. Він надає простий спосіб такого представлення метаданих, яке робить їх доступними для систем-агрегаторів інформаційних ресурсів. Зібрані в такий спосіб метадані можуть бути представлені в будь-якому форматі, обраному співтовариством організацій, що вирішили об'єднати свої зусилля для створення інтегрованої федеративної інформаційної системи.

3 https://www.budapestopenaccessinitiative.org

Репозитарій. Електронний архів або сховище для довготривалого, постійного та надійного накопичення, зберігання, управління та розповсюдження інформаційних ресурсів. **Інституційний репозитарій** має відношення до наукових інформаційних ресурсів, отриманих у результаті досліджень певної наукової установи. В **тематичних репозитаріях** інформаційні ресурси фокусуються на певних галузях знань. **Типи репозитаріїв** визначаються типами інформаційних ресурсів (журнальні статті, препринти, тези, книги, монографії, дослідницькі дані, зображення, карти, аудіо, відео і т.ін.). **Централізовані репозитарії** уможливлюють інтегроване зберігання інформаційних ресурсів різних типів і галузей знання у великому обсязі. **Спільнотні репозитарії** створюються спільнотами дослідників, учених або наукових установ з метою обміну даними та співпраці у конкретній галузі.

Цільова сторінка. Сторінка HTML у харвестері, куди зазвичай потрапляють під час доступу до наукової публікації, та з якої можна отримати доступ до пов'язаних ресурсів.

Харвестер. Програмний інструмент для автоматичного збору метаданих наукових періодичних видань НАН України та інформаційних ресурсів установ НАН України, редагування метаданих у харвестері, передачі метаданих у інші харвестери тощо.

2. Чинні служби агрегації відкритого доступу та бази даних публікацій

Наразі існує низка доступних агрегаційних служб відкритого доступу (Табл. 1), прикладами яких є BASE[4], OpenAIRE[5], CORE[6], Unpaywall[7], Paperity[8], SHARE[9].

BASE (Bielfield Academic Search Engine) – глобальна служба збору метаданих. Вона збирає репозитарії та журнали

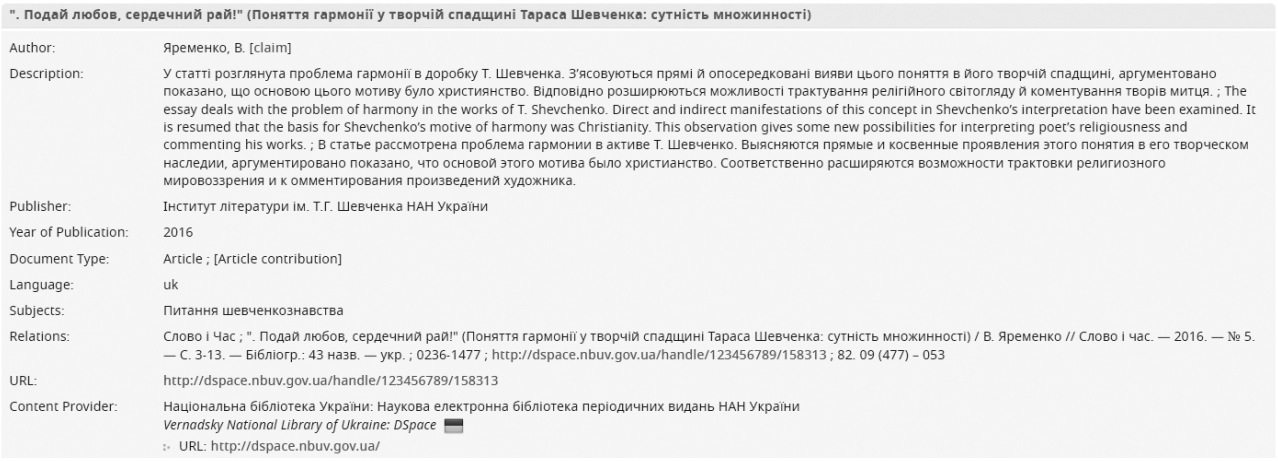
Таблиця 1. Порівняння служб агрегації відкритого доступу

Назва	Записи метаданих [/постачальники даних]	Записи з повним текстом	Записи з повнотекстовим посиланням	API	Набір даних	Синхронізація даних
CORE	298 млн / 11 139	32.8 млн	139 млн	Так	Так	Так■
BASE	362 млн / 11 399	0	~60%	Так	Ні	Ні
OpenAIRE	149 млн	19 млн*	н/в	Ні	Ні	Ні
Unpaywall	50 млн / 50 тис (+ через Crossref і DOAJ)	0	50 млн	Так	Так♦	Так●
Paperity	10.5 млн / 17 158 журналів	10.5 млн	н/в	Ні	Ні	Ні
SHARE	57 млн	0	н/в	Ні	Ні	Ні

Примітка: * Повні тексти, розміщені на OpenAIRE, недоступні для завантаження. ♦ Набір даних, наданий Unpaywall, містить лише посилання, а не повні метадані чи повний текст, як у випадку з CORE. ■ CORE використовує механізм FastSync для синхронізації даних. ● Unpaywall забезпечує синхронізацію даних як частину преміум-сервісу.

4 https://base-search.net/
5 https://www.openaire.eu/
6 https://core.ac.uk/
7 http://unpaywall.org/
8 https://paperity.org/
9 https://share.osf.io/

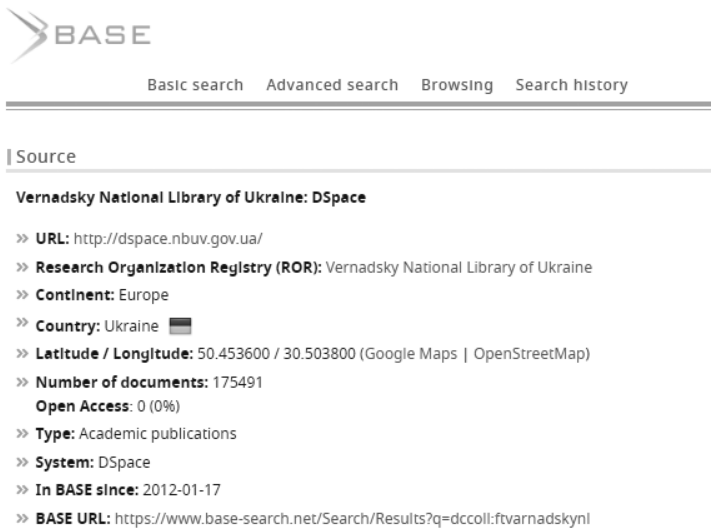
по протоколу OAI-PMH і надає зібраний вміст через API та набір даних. Наукова електронна бібліотека періодичних видань НАН України, https://dspace.nbuv.gov.ua, яку ми підтримуємо, підключена до цього агрегатора у якості провайдера даних. На рис. 1–2 наведені приклади, яким чином тут представлені статті та сам провайдер даних (https://www.base-search.net/about/en/about_source.php?id=2328).



". Подай любов, сердечний рай!" (Поняття гармонії у творчій спадщині Тараса Шевченка: сутність множинності)

Author:	Яременко, В. [claim]
Description:	У статті розглянута проблема гармонії в доробку Т. Шевченка. З'ясовуються прямі й опосередковані вияви цього поняття в його творчій спадщині, аргументовано показано, що основою цього мотиву було християнство. Відповідно розширюються можливості трактування релігійного світогляду й коментування творів митця. ; The essay deals with the problem of harmony in the works of T. Shevchenko. Direct and indirect manifestations of this concept in Shevchenko's interpretation have been examined. It is resumed that the basis for Shevchenko's motive of harmony was Christianity. This observation gives some new possibilities for interpreting poet's religiousness and commenting his works. ; В статье рассмотрена проблема гармонии в активе Т. Шевченко. Выясняются прямые и косвенные проявления этого понятия в его творческом наследии, аргументировано показано, что основой этого мотива было христианство. Соответственно расширяются возможности трактовки религиозного мировоззрения и к омментирования произведений художника.
Publisher:	Інститут літератури ім. Т.Г. Шевченка НАН України
Year of Publication:	2016
Document Type:	Article ; [Article contribution]
Language:	uk
Subjects:	Питання шевченкознавства
Relations:	Слово і Час ; ". Подай любов, сердечний рай!" (Поняття гармонії у творчій спадщині Тараса Шевченка: сутність множинності) / В. Яременко // Слово і час. — 2016. — № 5. — С. 3-13. — Бібліогр.: 43 назв. — укр. ; 0236-1477 ; http://dspace.nbuv.gov.ua/handle/123456789/158313 ; 82. 09 (477) – 053
URL:	http://dspace.nbuv.gov.ua/handle/123456789/158313
Content Provider:	Національна бібліотека України: Наукова електронна бібліотека періодичних видань НАН України *Vernadsky National Library of Ukraine: DSpace* URL: http://dspace.nbuv.gov.ua/

Рис. 1. Приклад представлення опису документу у агрегаторі BASE



BASE

Basic search Advanced search Browsing Search history

Source

Vernadsky National Library of Ukraine: DSpace

- **URL:** http://dspace.nbuv.gov.ua/
- **Research Organization Registry (ROR):** Vernadsky National Library of Ukraine
- **Continent:** Europe
- **Country:** Ukraine
- **Latitude / Longitude:** 50.453600 / 30.503800 (Google Maps | OpenStreetMap)
- **Number of documents:** 175491
  Open Access: 0 (0%)
- **Type:** Academic publications
- **System:** DSpace
- **In BASE since:** 2012-01-17
- **BASE URL:** https://www.base-search.net/Search/Results?q=dccoll:ftvarnadskynl

Рис. 2. Приклад опису провайдера даних у агрегаторі BASE

OpenAIRE – це мережа постачальників даних відкритого доступу, які підтримують політику відкритого доступу. Незважаючи на те, що в минулому проєкт був зосереджений на європейських репозитаріях, нещодавно він розширився за рахунок інституційних і предметних репозитаріїв поза межами Європи. Основним напрямком OpenAIRE є допомога Європейській Раді в моніторингу дотримання її політики відкритого доступу. Дані OpenAIRE доступні через API. Наукова електронна бібліотека періодичних видань зареєстрована у цій системі з 2018 року з рівнем сумісності v2.0 з інструкціями OpenAIRE (рис. 3). На даний час є питання щодо оновлення нашої системи до рівня сумісності v3.0 або v4.0. Оскільки нова інтегрована платформа каталогу EOSC Portal Catalog та Marketplace тепер обслуговуватиме лише зареєстровані джерела даних OpenAIRE, які сумісні з версіями 3.0 та 4.0 інструкцій OpenAIRE. Це дасть нам змогу отримувати додаткові послуги, наприклад, буде можливість отримати звіти про використання досліджень. OpenAIRE збирає дані про використання, а потім об'єднує їх і надає стандартизовані звіти.

Рис. 3. Наукова електронна бібліотека періодичних видань у агрегаторі OpenAIRE

Paperity – перший мультидисциплінар-ний агрегатор журналів і статей відкритого доступу, який був запущений 2014 року. Paperity збирає як метадані, так і повні тексти. Наразі агрегатор містить 10 508 682 повнотекстових статей та 17 158 журналів (рис. 4) в усіх галузях досліджень: від науки, технологій, медицини до соціальних наук, гуманітарних наук і мистецтва. Метою Paperity є надання читачам легкого і необмеженого доступу до тисяч журналів із сотень дисциплін в одному центральному місці; допомога авторам охопити цільову аудиторію, ефективніше поширювати відкриття та максимізувати вплив досліджень; підвищення популярності журналів, збільшення їх читацької аудиторії та заохочення подання нових рукописів. Paperity має наступні особливості:

Доступність. Усі статті, які проіндексовані Paperity, мають відкритий доступ і доступні з повним текстом. Це свідомий вибір. Вчені втомилися від платних платформ і вимагають легкого доступу до потрібної їм літератури, особливо якщо ця література створена ними самими. Наукова комунікація має бути відкритою, адже відкритість – це не про вартість, а про свободу наукових досліджень.

Уніфікація. Сьогодні науковці потребують широкого доступу до літератури з різних галузей, навіть з-поза меж їхньої основної дослідницької сфери. Сучасні дослідження стали міждисциплінарними і найбільш новаторські відкриття відбуваються на перетині різних дисциплін. Академічні служби мають наздоганяти цей процес. Paperity – це шлях до ефективнішої наукової комунікації.

Цілісність та висока якість корпусу. Paperity гарантує, що індексується дійно наукова література. Завдяки унікальній тех-

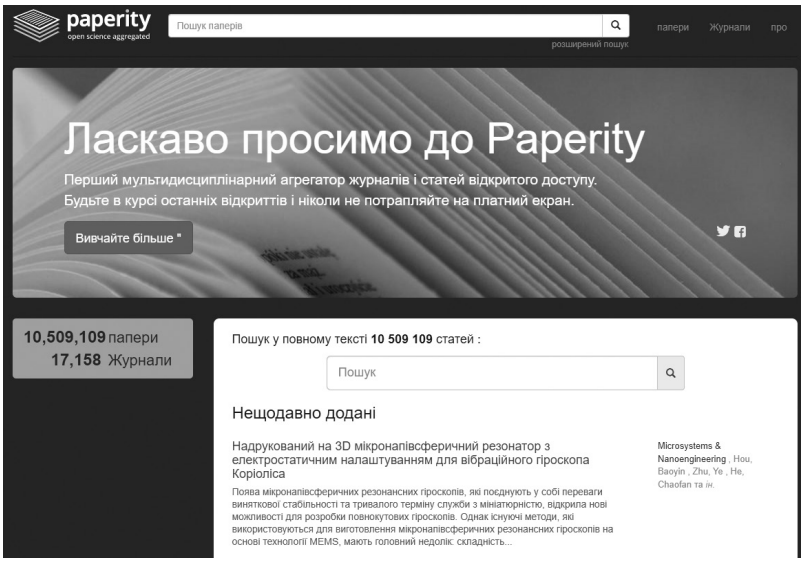


Рис. 4. Домашня сторінка агрегатора Paperity

нології тут збирають детальні метадані про кожну публікацію і пильно стежать за тим, щоб не забруднювати колекцію нерелевантними записами. У Paperity ніколи не знайти студентських завдань, бізнес-презентацій або кулінарних рецептів, класифікованих як наукові роботи.

Unpaywall – це безкоштовна база даних наукових статей відкритого доступу з API та розширенням для браузера. Розроблена некомерційною організацією Impactstory[10], яка опікується проблемами відкритого доступу в науці. Тобто це не агрегатор, а скоріше збір вмісту з Crossref щоразу, коли безкоштовна версія доступна для читання. Обробляє як метадані, так і повний текст, але не розміщує їх. Розкриває виявлені посилання на документи через API. Значну частину даних інструмент отримує з бази даних під назвою oaDOI, яка індексує більше сотні мільйонів документів, яким присвоєно цифрові ідентифікатори об'єктів (DOI).

Unpaywall – це також і плагін для веб-браузера (наприклад, Firefox або Chrome), який визначає потрібний вам документ, а потім перевіряє, чи доступний він безкоштовно будь-де в Інтернеті. І коли ви

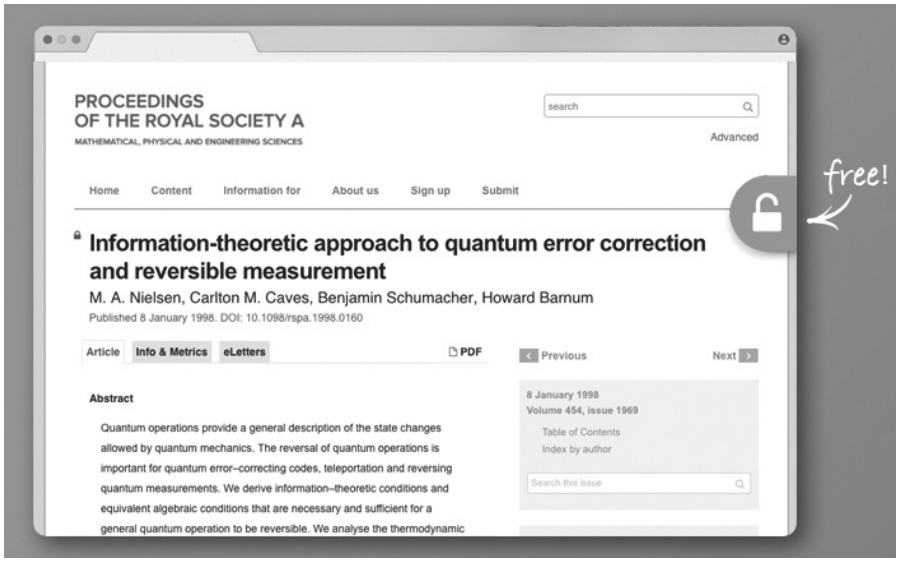


Рис. 5. Робота розширення Unpaywall

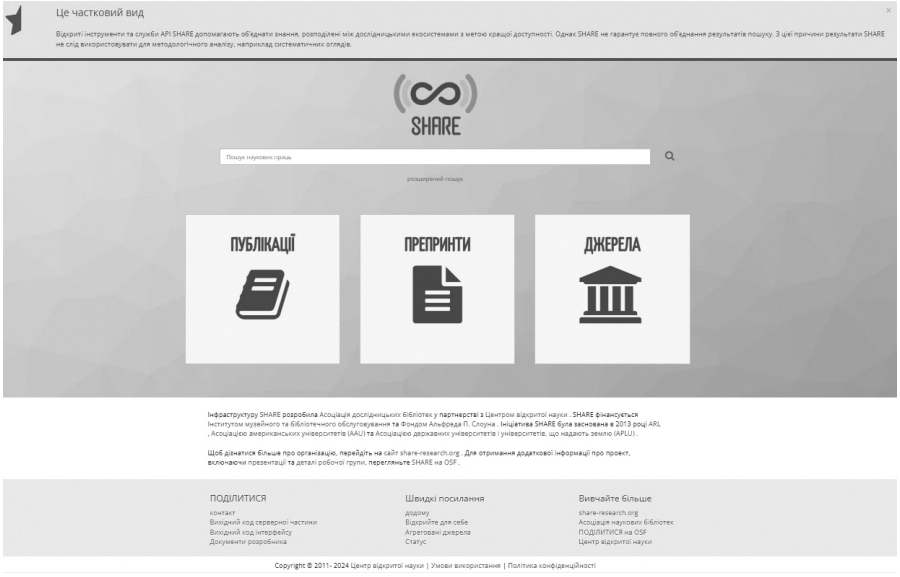


Рис. 6. Дослідницька екосистема спільного доступу SHARE

[10] https://impactstory.org/about

перейдете на сторінку, яка підсумовує чи показує частину статті, з'явиться маленький значок замка (рис. 5), який повідомляє, чи можливо його отримати десь ще безкоштовно. Якщо стаття доступна, сірий значок «замок» Unpaywall стане зеленим і «розблокується». Клацнувши на нього, користувач отримує доступ до PDF-файлів, яке спрощує пошук PDF-файлів із відкритим доступом.

SHARE (SHared Access Research Ecosystem), дослідницька екосистема спільного доступу – це збирач вмісту відкритого доступу з репозиторіїв США, https://aims.fao.org/news/share-making-research-widely-accessible-discoverable-and-reusable. Незважаючи на те, що SHARE збирає як метадані, так і повний текст, він не розміщує останній.

CORE (COnnecting REpositories) об'єднує наукові статті у відкритому доступі від тисяч постачальників даних з усього світу, включно з інституційними та предметними репозитаріями, журналами відкритого та гібридного доступу. CORE є найбільшою колекцією літератури з відкритого доступу – на момент написання цієї статті вона забезпечує єдину точку доступу до наукової літератури, зібраної від понад десяти тисяч постачальників даних з усього світу, і ця колекція постійно зростає. Вона надає кілька способів доступу до своїх даних як для користувачів, так і для машин, включно із безкоштовним API і повним дампом своїх даних. На наш погляд, CORE – найцікавіший агрегатор з точки зору розвитку нашої системи у майбутньому. Наразі ми маємо дві свої бібліотеки, зареєстровані у CORE (рис. 7–9). Це вищезгадана бібліотека періодичних видань НАНУ, що працює на програмному забезпеченні DSpace[11], і електронна бібліотека Інституту програмних систем НАНУ, що працює на програмному забезпеченні EPrints[12].

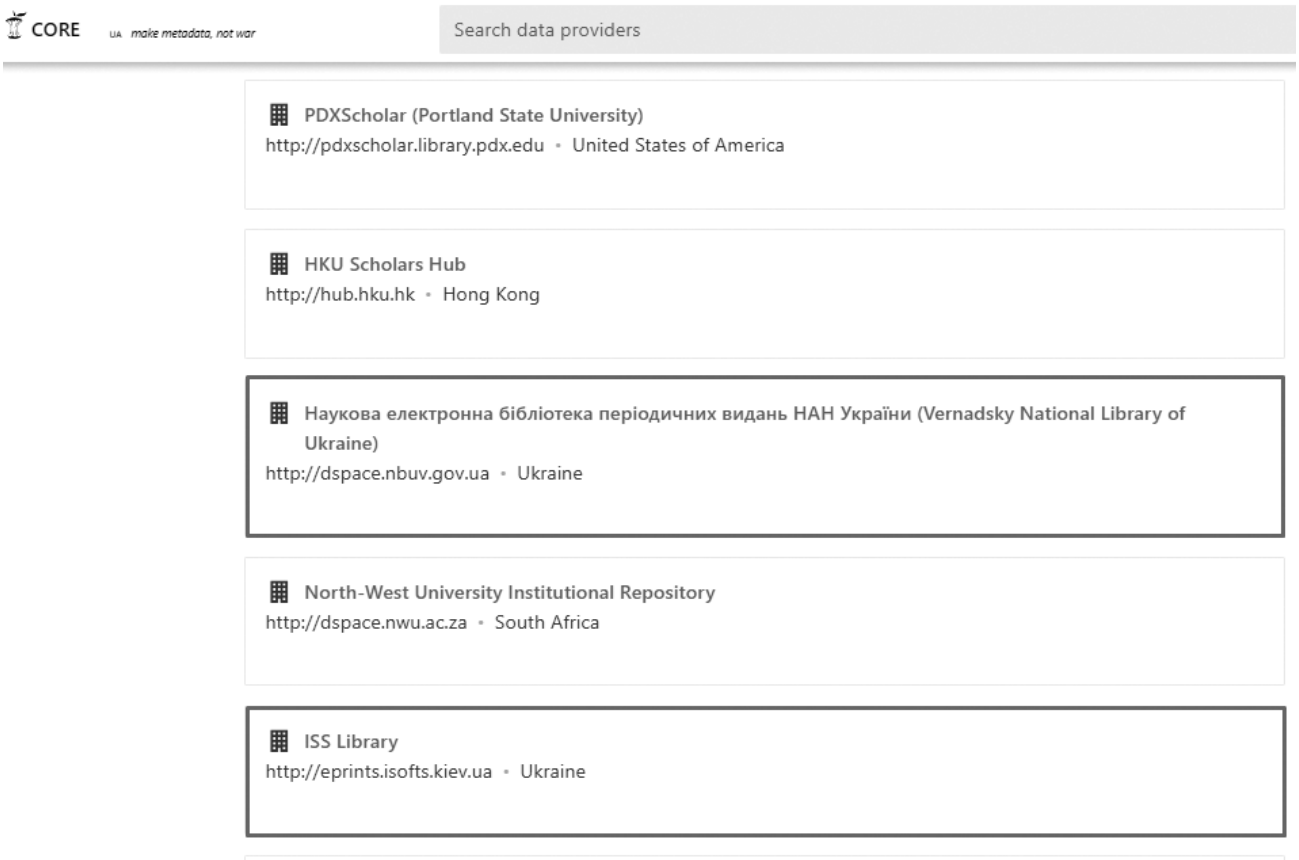


Рис. 7. Наші бібліотеки у списку постачальників даних у агрегаторі CORE

11 https://dspace.lyrasis.org/
12 https://www.eprints.org/uk/

Рис. 8. Наукова електронна бібліотека періодичних видань НАН України у агрегаторі CORE[13]

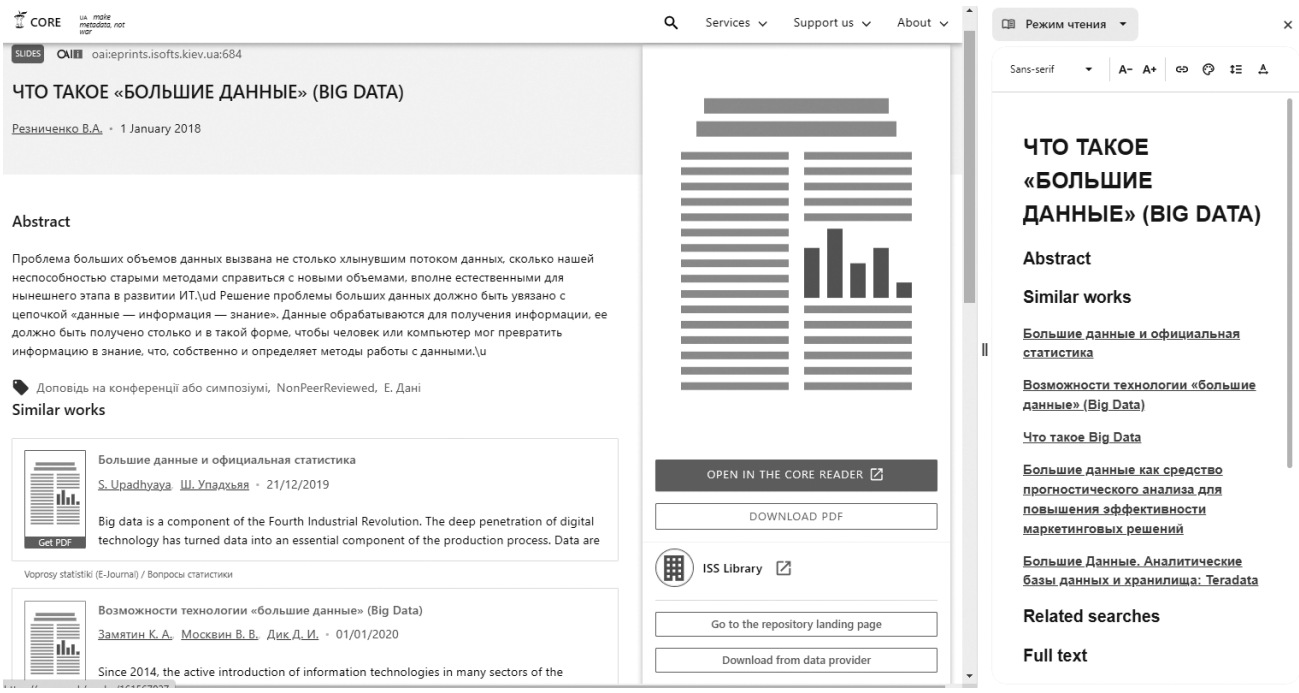


Рис. 9. Сторінка представлення повного тексту документу з ISS Library у агрегаторі CORE

13 https://core.ac.uk/data-providers/944

Джерела даних CORE. Станом на квітень 2024 року CORE агрегував контент із 11 139 джерел даних. Ці джерела даних включають інституційні репозитарії, академічні видавництва (Elsevier, Springer), журнали відкритого доступу, предметні репозиторії, в тому числі ті, що містять електронні версії (arXiv, ZENODO, PubMed Central) та агрегатори (наприклад, DOAJ). Декілька найбільших джерел даних CORE наведено в Таблиці 2. При підрахунку загальної кількості постачальників даних у CORE агрегатори і видавці розглядаються як одне джерело даних, не зважаючи на те, що деякі з них, в свою чергу, агрегують дані з багатьох джерел. Повний список усіх постачальників даних можна знайти на веб-сайті CORE, https://core.ac.uk/data-providers.

Таблиця 2. Найбільші постачальники даних у CORE

Назва, url	Кількість документів / повних текстів
Crossref, https://crossref.org/	118.155.624 / 1.532.012
CiteSeerX, http://citeseerx.ist.psu.edu	6.540.649 / 80.766
Directory of Open Access Journals, https://doaj.org/	5.463.188 / 990.721
Zenodo, https://zenodo.org	3.321.673 / 51,927
Elsevier - Publisher Connector	1.682.191 / 840.998

Примітка. Агрегатори вмісту, такі як DOAJ і Elsevier, представлені як одне джерело даних, незважаючи на те, що вони самі є агрегаторами і збирають дані з багатьох джерел.

Crossref. Як видно з Таблиці 2, серед провайдерів даних CORE, однією з основних баз даних публікацій є Crossref, авторитетний покажчик ідентифікаторів DOI (The Digital Object Identifier).

DOI – це обов'язковий міжнародний цифровий ідентифікатор наукової публікації. DOI визначає постійне місце знаходження наукової роботи (об'єкта) в інтернеті, її назву та метадані[14]. DOI на сьогодні є обов'язковою складовою сучасної системи наукової комунікації. Він полегшує процедуру та облік цитування, пошуку та локалізації наукової публікації. Ви можете присвоїти DOI будь-якій публікації, наприклад: науковій статті, монографії, главі монографії, дисертації, автореферату, рецензії, підручнику, методичним розробкам/рекомендаціям, звіту, препринту і навіть окремо таблиці, рисунку, схемі, зображенню тощо. DOI призначається публікації раз і назавжди. Це забезпечує стабільність посилання на публікацію в Інтернеті та спрощує пошук потрібної інформації.

Отже, основною функцією бази даних публікацій Crossref є збереження метаданих, пов'язаних з кожним DOI. Метадані, які зберігає Crossref, включають записи як ВД, так і не ВД. Crossref не зберігає повний текст публікації, але для багатьох публікацій надає посилання на повні тексти. Хоча Crossref надає API, він не пропонує свої дані для масового завантаження та не надає служби синхронізації даних.

Zenodo – це безкоштовний і відкритий цифровий архів, створений CERN і OpenAIRE, який дає змогу дослідникам ділитися і зберігати свої наукові результати в будь-якому обсязі, форматі та з усіх галузей досліджень. На Zenodo можна знайти різноманітні типи даних:

1. Датасети: набори даних, які дослідники можуть завантажувати та ділитися зі спільнотою.
2. Публікації: наукові статті, дисертації, звіти та інші публікації.
3. Програмне забезпечення: відкритий вихідний код, бібліотеки, інструменти та програми.

[14] https://sciencen.org/o/doi/

4. Презентації: презентації, семінари та доповіді.

Якщо планується використовувати Zenodo, слід ознайомитися з розділом Get started, щоб дізнатися, як створити обліковий запис, завантажити файли та орієнтуватися в інтерфейсі.

Інші бази публікацій. Окрім сервісів агрегації ВД, існує низка інших сервісів для пошуку та завантаження наукової літератури, Таблиця 3.

Решту послуг із Таблиці 3 можна згрупувати приблизно в такі дві категорії: 1) індекси цитування, 2) академічні пошукові

Таблиця 3. Порівняння баз публікацій

Назва	Вільний доступ	Розміщується повний текст	API	Набір даних	Синхронізація даних
CORE	Так	Так	Так	Так	Так
Crossref	Так	Ні	Так	Ні	Ні
Scopus	Ні	Так	Так	Ні	Ні
Web of Science	Ні	Так	Так	Ні	Ні
Google Scholar	н/в*	Ні	Ні	Ні	Ні
Semantic Scholar	Так	Так	Так	Так	Ні
Dimensions	Так	Так	Так	Ні	Ні
1findr	Ні	Так	Ні	Ні	Ні

Примітка. Під «вільним доступом» тут розуміють, чи є доступ до бази даних вільним за допомогою автоматизованих методів (наприклад, через API). *Google Scholar не надає засобів програмного доступу до своїх даних.

системи та наукові графи. Двома основними індексами цитування є Scopus від Elsevier[15] і Web of Science від Clarivate[16], які є послугами передплати преміум-класу. Google Scholar, найвідоміша академічна пошукова система, не надає API для доступу до своїх даних і не дозволяє сканувати свій веб-сайт. Semantic Scholar[17] є відносно новою академічною пошуковою службою, метою якої є створення «інтелектуальної академічної пошукової системи» [4]. Dimensions[18] – це сервіс, орієнтований на аналіз даних. Він об'єднує публікації, гранти, політичні документи та показники. 1findr[19] – це служба індексування рефератів. Він надає посилання на повний текст, але не містить API чи набору даних для завантаження.

Процес збору документів у агрегаторі CORE. Процес збору можна описати як послідовність етапів, на кожному етапі виконується певна дія, і коли вихідні дані одного етапу передаються на вхід наступному [3]. Вхідними даними для цього процесу є набір постачальників даних, а кінцевим виходом є система, заповнена записами дослідницьких робіт. Основні типи завдань, які наразі виконуються в рамках системи збору документів, наступні:

Завантаження метаданих: метадані, надані постачальником даних за протоколом OAI-PMH, завантажуються та зберігаються у файловій системі (зазвичай як XML-файли). Процес завантаження є послідовним, тобто репозитарій зазвичай надає від 100 до 1000 записів метаданих на запит і маркер продовження. Потім цей маркер використовується для надання наступної партії записів. У результаті повне збирання може займати чимало часу (годин-днів) для великих постачальників даних. Такий процес

15 https://www.elsevier.com/solutions/scopus
16 https://clarivate.com/webofsciencegroup/solutions/web-of-science/
17 https://www.semanticscholar.org/
18 https://www.dimensions.ai/
19 https://1findr.1science.com/home

було впроваджено для забезпечення стійкості до низки комунікаційних збоїв.

Витяг метаданих аналізує, очищає та узгоджує завантажені метадані та зберігає їх у внутрішній структурі даних. Процес узгодження та очищення розглядає той факт, що різні постачальники даних описують одну й ту саму інформацію по-різному (синтаксична неоднорідність), а також мають різні інтерпретації для однієї й тієї самої інформації (семантична неоднорідність).

Завантаження повного тексту: Використовуючи посилання, отримані з метаданих, CORE намагається завантажити та зберегти повні тексти публікацій. Цей процес є нетривіальним. Як було сказано вище, OAI-PMH наразі є стандартним протоколом для обміну даними між сховищами. Хоча OAI-PMH спочатку був розроблений лише для збору метаданих, через його широке застосування та відсутність альтернатив він використовувався як точка входу для збору і повного тексту. Збір повного тексту досягається шляхом витягу URL-адрес із записів метаданих, зібраних за допомогою OAI-PMH. А потім витягнуті URL-адреси використовуються для визначення розташування фактичного ресурсу. Однак протокол OAI-PMH безпосередньо підтримує лише збирання метаданих, тобто необхідно реалізувати додаткові функції, щоб використовувати його для збору вмісту. Розташування посилань на повні тексти у метаданих не стандартизовано, і записи метаданих зазвичай містять кілька посилань. З метаданих не зрозуміло, яке з цих посилань вказує на описане представлення ресурсу, і в багатьох випадках жодне з них не робить цього безпосередньо. Тому всі можливі посилання на сам ресурс потрібно витягнути з метаданих і перевірити, щоб визначити правильний ресурс. Крім того, OAI-PMH не сприяє перевірці того, що виявлений ресурс справді є описаним ресурсом. Подолання цих проблем сприяло прийняттю формату метаданих RIOXX[20] або рекомендацій OpenAIRE[21]. Однак питання однозначного зв'язку запису метаданих і описаного ресурсу залишається актуальним.

Витяг інформації. Звичайний текст із завантажених документів витягується та обробляється для створення напів-структурованого представлення. Цей процес включає низку завдань добування інформації, таких, наприклад, як витяг посилань.

Збагачення працює шляхом доповнення метаданих і повного тексту, отриманих від постачальників даних, додатковими даними з різних джерел. Деякі збагачення виконуються безпосередньо конкретними завданнями в процесі виконання збору документів, зокрема, визначення мови і типу документа (стаття, презентація, дипломна робота, інший). Ці збагачення зазвичай передбачають застосування моделей машинного навчання та інструментів на основі правил для збору додаткової інформації про записи. І для певного запису виконуються лише один раз. Решта збагачень, які включають зовнішні набори даних, виконуються ззовні, незалежно від процесу збору, і вбудовуються в набір даних. Це здійснюється шляхом збору різноманітної інформації з великих сторонніх наукових наборів даних. Інформація включає метадані, які не обов'язково змінюються, як-от, ідентифікатор DOI, а також метадані, які зазнають змін, наприклад, кількість цитувань. Саме через це такі збагачення виконуються періодично, тобто всі записи в CORE проходять цей процес повторно через певні проміжки часу. Початкове відображення запису здійснюється за допомогою DOI у разі його доступності. Однак, оскільки більшість записів з репозиторіїв не мають DOI, відбувається процес зіставлення з базою даних Crossref, використовуючи підмножину полів метаданих, включаючи назву, авторів і рік публікації. Після того, як зіставлення виконано, можна узгодити поля, а також зібрати широкий спектр додаткових корисних даних із відповідних зовнішніх баз даних, тим самим збагачуючи запис CORE. Такі дані включають ідентифікатори ORCID, інформацію про цитування, додаткові посилання на вільно

20 https://rioxx.net/

21 https://guidelines.openaire.eu/

доступні повні тексти, інформацію про галузь дослідження тощо.

Індексування. Останнім кроком збирання даних є індексування зібраних даних. Отриманий індекс забезпечує роботу служб CORE, включно із пошуком, API і FastSync.

Сервіси інтелектуального аналізу даних. Особливий інтерес для нас становлять сервіси, які надаються/ плануються самими агрегаторами (або із залученням сторонніх організацій) своїм користувачам на основі сучасних досліджень інтелектуального аналізу тексту та даних наукової літератури. Тут ми просто наводимо їх список: виявлення плагіату нещодавно надісланих публікацій; семантометрики [5]; підбір публікаціям відповідних рецензентів; аналіз науково-дослідних трендів; рекомендації щодо роботи зі співавторами, заходами проведення тощо; ідентифікація різних версій одної і тої самої статті (наприклад, препринти та постпринти); визначення того, чи відповідає публікація умовам проведення заходу; визначення важливості, настанов або типу цитування; узагальнення результатів досліджень; побудова індексу цитувань; витяг або добування важливих слів або фраз; категоризація публікацій за сферами досліджень; визначення типу публікації; витяг цитат для бібліометрії.

3. Вимоги та рекомендації щодо індексування вмісту постачальників даних

Майже всі розглянуті у попередньому розділі служби агрегації документів відкритого доступу мають більш-менш стандартні процедури реєстрації та вимоги до провайдерів даних. У цьому розділі стисло описано основні вимоги та вказівки щодо найкращого налаштування репозитаріїв для індексування у таких агрегаторах.

Підтримка протоколу OAI-PMH. Майже всі сучасні системи агрегації використовують цей протокол для регулярного збору та оновлення інформації, яку вони зберігають від своїх постачальників даних. Обов'язково слід переконатися, що ваш репозиторій видимий через OAI-PMH. Щоб пройти індексацію, постачальники даних мають надати інформацію про свої ресурси через OAI-PMH.

Доволі поширеною проблемою є те, що кінцева точка OAI-PMH постачальника даних неправильно налаштована або не функціонує. Це може статися навіть тоді, коли інші функції репозитарію працюють без проблем. Тому важливо, щоб репозитарії стежили за тим, щоб їхня кінцева точка OAI-PMH залишалася функціональною. Це можна зробити кількома способами.

Валідація через інструменти перевірки OAI-PMH. Постачальники даних, які розглядають можливість реєстрації в агрегаторі можуть використати інструмент перевірки Open Archives OAI-PMH, https://www.openarchives.org/Register/ValidateSite, рис. 10. Інструмент очікує базову URL-адресу OAI-PMH і проводить низку перевірок, щоб оцінити коректність конфігурації кінцевої точки. Якщо валідатор повертає помилки, їх потрібно буде виправити до реєстрації в агрегаторі. Рекомендується виконати цей крок до спроби реєстрації в агрегаторі.

Реєстрація постачальників даних у агрегаторі. Відвідайте сторінку реєстрації постачальника даних у та введіть URL-адресу OAI-PMH свого репозитарію. Якщо URL-адреса OAI-PMH є дійсною, репозитарій буде зареєстровано за умови проходження додаткових перевірок, описаних нижче. Важливо зазначити, що перевірка правильності конфігурації кінцевої точки OAI-PMH не гарантує, що вона правильно відображає всі записи метаданих із системи репозитарію. Це можна перевірити лише після того, як агрегатор спробує отримати дані з репозитарію.

Постачальники даних, які вже зареєстровані в агрегаторі. Зареєстровані постачальники даних можуть отримати доступ до інформаційної панелі агрегатора, як, наприклад, у системі CORE, рис. 11. Панель управління надає доступ до звіту про збір даних, який містить таку інформацію, як останній випадок успішного збору даних, кількість знайдених та проіндексованих метаданих і повнотекстових записів (якщо повні тексти збираються агрегатором), будь-які помилки або попередження, що виникли в процесі збирання даних.

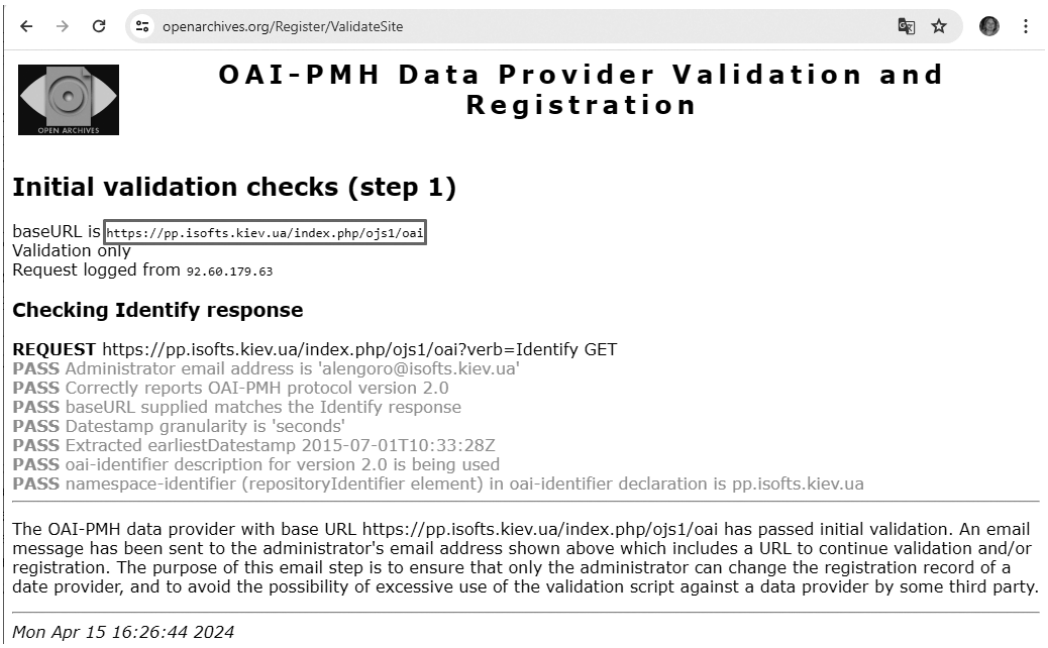


Рис. 10. Зразок перевірки OAI-PMH журналу «Проблеми програмування»

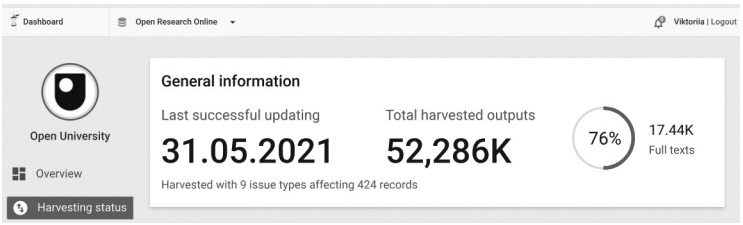


Рис. 11. Показ збору інформації агрегатором CORE репозитарію Відкритого університету

Нерідко кількість проіндексованих записів метаданих у агрегаторах може дещо відрізнятися від кількості, яку постачальник даних може бачити у своїй системі репозитарію, оскільки інформаційна панель надає незалежний зовнішній погляд на репозитарій. Відхилення можуть виникати через низку причин, зокрема, тому, що ваш репозитарій не відкриває всі записи через інтерфейс OAI-PMH, деякі записи відключені або їхні метадані не відповідають мінімальним вимогам до якості. Тому постачальникам даних рекомендується регулярно перевіряти і виправляти будь-які помилки та попередження, що з'являються на інформаційній панелі.

Реєстрація репозитарію у відкритих реєстрах. Агрегатори використовують визнані відкриті реєстри репозитаріїв, як-от OpenDOAR[22], для виявлення нових постачальників даних з відкритим доступом. Тому бажано, щоб ваш репозитарій був попередньо включений до такого міжнародного списку репозитаріїв і щоб базова URL-адреса OAI-PMH підтримувалася в актуальному стані в цьому реєстрі. Наприклад, для реєстрації будь-якого провайдера даних у агрегаторі OpenAIRE, попередня реєстрація у реєстрі OpenDOAR є обов'язковою вимогою. Нерідко зустрічаємо і таку ситуацію, коли одні агрегатори автоматично індексують інші агрегатори, як, наприклад, серед провайдерів даних агрегатора CORE є агрегатор DOAJ (Каталог журналів відкритого доступу), кілька серверів препринтів та наукових праць, які мають ідентифікатори публікацій, зареєстрованих у Crossref.

22 http://v2.sherpa.ac.uk/opendoar/

Конфігурація метаданих. Для того, щоб агрегатор міг успішно збирати дані із репозитарію, як було сказано вище, він повинен мати кінцеву точку, сумісну зі стандартом OAI-PMH. Більшість поширених програм, на яких будуються репозитарії, такі як EPrints, DSpace або Open Journal Systems (OJS) підтримують OAI-PMH. Крім того, для успішного індексування важливо, щоб метадані записів у репозитарії були правильними та у підтримуваному форматі.

Обов'язкова підтримка стандартів метаданих. Агрегатори можуть підтримувати кілька форматів метаданих для опису наукових ресурсів [1]. Як правило, метадані повинні бути представлені в одному з наступних профілів.

i. Dublin Core / Extended Dublin Core (мінімум). Дублінське ядро є однією з найпростіших і найпоширеніших схем метаданих. Розширена версія була формалізована у вигляді термінів метаданих DCMI 2019 року, з приблизно 70 полями даних. Хоча Dublin Core та Extended Dublin Core є достатніми для індексації, наприклад, у агрегаторі BASE, ця схема надає лише обмежені можливості для опису наукових ресурсів порівняно з OpenAIRE Guidelines та Rioxx.

ii. OpenAIRE Guidelines була створена для підтримки стратегії відкритого доступу Європейської Комісії та для задоволення вимог інфраструктури OpenAIRE. Ця нова версія настанов відповідно до розширення цілей ініціативи OpenAIRE та її інфраструктури має ширшу сферу застосування. Впроваджуючи ці настанови, менеджери репозитаріїв дозволять авторам, які розміщують публікації в їхніх репозитаріях, відповідати вимогам Європейської Комісії щодо відкритого доступу.

iii. RIOXX. Агрегатор CORE рекомендує репозитаріям використовувати формат метаданих RIOXX (The Research Outputs Metadata Schema), як найбільш придатний профіль метаданих для опису результатів наукових досліджень. Ця схема метаданих була розроблена для інституційних репозитаріїв з метою обміну метаданими про наукові ресурси, які вони містять. Спочатку розроблена для задоволення вимог до звітності Дослідницьких рад Великої Британії (Research Councils UK, RCUK)[23], RIOXX також виявилася корисною як стандарт для обміну метаданими між репозиторіями та мережевими сервісами, такими як великомасштабні агрегатори метаданих, такі як CORE. RIOXX фокусується на застосуванні узгодженості до полів метаданих, що використовуються для ідентифікаторів наукових результатів, людей і організацій, спонсорів досліджень і проєктів/грантів, і призначений для підтримки послідовного відстеження наукових публікацій з відкритим доступом у різних наукових системах. RIOXX має кілька переваг над іншими схемами. Зокрема, ця схема була розроблена з особливим акцентом на забезпеченні ефективного обміну даними між репозиторіями і машинними агентами, що робить збір даних швидшим і точнішим завдяки уникненню неоднозначності у зв'язках метаданих, що описують ресурси, такі як повні тексти, набори даних, програмне забезпечення, і самі ресурси. Вона надає значні можливості для опису широкого спектру наукових властивостей, корисних для звіту та аналізу досліджень, таких як ідентифікатори грантів, ідентифікатори проєктів, ідентифікатори спонсорів, інформація про ліцензування тощо.

Схема RIOXX сумісна з іншими протоколами машинного доступу до наукових документів, зокрема, Signposting[24] та ResourceSync[25], і логічно інтегрується з ними. Вона забезпечує механізми для розрізнення ресурсів, що знаходяться під управлінням репозитарію, від ресурсів, які знаходяться під зовнішнім управлінням, що дозволяє CORE надавати перевагу посиланням на репозитарій, де це можливо (і доречно), замість зовнішніх посилань на веб-сайт видавця. CORE надає валідатор[26] метаданих для постачальників даних, які підтримують RIOXX.

3 https://www.ukri.org/

4 https://signposting.org/

5 https://www.openarchives.org/rs/1.1/resourcesync

6 https://core.ac.uk/membership-documentation#rioxx-validator

Кодування символів. Весь вміст інтерфейсу OAI-PMH (заголовки, імена авторів, анотації) кодується в UTF-8. Інші кодування або дублювання кодувань спричиняють помилки у відображенні результатів пошуку з вашого джерела.

Розділення кількох записів у полі метаданих. Якщо у полі метаданих вказані кілька записів (наприклад, ім'я автора та його ідентифікатор ORCID), розділяйте їх пробілами, крапкою з комою та пробілами. Таке розділення дозволяє індексувати інформацію окремо і зробити її доступною для пошуку.

Повнотекстова конфігурація. Як вже було сказано вище, низка потужних можливостей деяких агрегаторів заснована на здатності індексувати повнотекстовий контент. До них належать повнотекстовий пошук, рекомендації повнотекстового контенту, виявлення версій і дубльованого контенту та інші. Підтримувані повнотекстові формати – тільки дійсні файли PDF, DOC або DOCX, які містять текст, що витягується. Якщо PDF-файл це відскановане зображення, то використання документа в агрегаторах буде обмежено. Для успішного індексування повного тексту статті на нього має бути посилання одним з наступних способів.

i. Пряме посилання на повний текст. Рекомендується, аби постачальники даних надавали однозначне посилання на повний текст у метаданих. Правильна стратегія посилання залежить від використовуваної схеми метаданих. Там, де використовується Дублінське ядро, рекомендується надавати пряме посилання на повний текст у першому входженні dc:identifier.

<dc:identifier>http://oro.open.ac.uk/37823/1/jcdl2019_v7.pdf</dc:identifier>

Система EPrints за замовчуванням дотримується цієї рекомендації. Такий підхід значно зменшує навантаження, яке агрегатор накладає на репозитарій під час індексації. Якщо це неможливо, то агрегатор може індексувати вміст, якщо в метаданих є чітко визначене посилання, яке вказує на машинодоступний документ (або PDF, або формат DOC/DOCX).

ii. Непряме посилання на повний текст. Агрегатор автоматично визначає, коли постачальник даних не вказує посилання на повний текст безпосередньо, як було описано вище. У таких випадках індексатор відвідає цільову сторінку і збере з неї посилання так само, як це зробив би користувач, намагаючись знайти посилання на документ на сторінці.

Щоб переконатися, що правильний повний текст був знайдений, тобто повний текст, що відповідає запису метаданих, агрегатор запускає процес зіставлення заголовка запису з заголовком, що міститься в PDF-файлі. Цей процес не є на 100% точним, спричиняє додаткове навантаження на сервер і вимагає більше часу на обробку одного документа. Це не працює для документів, які потребують розпізнавання тексту, і тому вони будуть відкинуті.

Висновки

У роботі визначена термінологія та проаналізована низка доступних і популярних агрегаційних служб та баз даних публікацій відкритого доступу BASE, OpenAIRE, CORE та ін. Здійснена спроба узагальнити їхні вимоги та рекомендації щодо збору ресурсів та індексування вмісту постачальників даних. Наші ЕБ, які ми створили і підтримуємо протягом уже майже 20 років, присутні у цих агрегаторах. Тому деякий практичний досвід ми маємо, і це дозволяє нам виконати проєкт НАНУ «Відкрита наука» зі створення агрегатора публікацій, який у подальшому буде підключено до більш потужних європейських або світових агрегаторів наукових публікацій.

Література

1. Проскудіна Г.Ю., Кудім К.О., Резніченко В.А. VUFIND: відкрите рішення для інтеграції бібліотечних колекцій // Проблеми програмування. − 2023. − № 4 – С. 15–26. https://pp.isofts.kiev.ua/ojs1/article/view/590
2. Резніченко В.А, Новицький О.В., Проскудіна Г.Ю. Інтеграція наукових електронних бібліотек на основі протоколу OAI-PMH // Проблеми програмування. −

2007. − № 2 – С. 97–112. http:// dspace.nbuv.gov.ua/hadle/123456789/291
3. Кнот П., Херрманнова Д., Канселлієрі М. та ін. CORE: Глобальна служба агрегації документів відкритого доступу. Nature Scientific Data 10, 366 (2023). https://www.nature.com/articles/s41597-023-02208-w
4. Ammar, W. et al. Construction of the Literature Graph in Semantic Scholar. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers): 84–91 (2018).
5. Кнот П., Херрманнова Д.. К вопросу о семантометрии: новый критерий для оценки вклада научной публикации на основе семантического сходства // Международный форум по информатизации. — 2015. — Т. 40, № 1. — С. 3-8.

References

1. Proskudina G.Yu., Kudim K.A., Reznichenko V.A. 2023. VuFind: an open solution for integrating library collections . Problems in programming. no. 4, pp. 15–26. (in Ukrainian). Available from: https://pp.isofts.kiev.ua/ojs1 / article/view/590 [Accessed 17/10/2024].
2. Reznichenko V.A.,. Novytskyi O.V., Proskudina G.Yu., 2007. OAI-PMH protocol-based integration of scientific digital libraries. Problems in programming, no. 2, pp. 97–112. (in Ukrainian). Available from: http://dspace.nbuv.gov.ua/handle/123456789/291 [Accessed 17/10/2024].
3. Knoth, P., Herrmannova, D., Cancellieri, M. et al. CORE: A Global Aggregation Service for Open Access Papers. Sci Data 10, 366 (2023). https://doi.org/10.1038/s41597-023-02208-w [Accessed 17/10/2024]
4. Ammar, W. et al. Construction of the Literature Graph in Semantic Scholar. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers): 84–91 (2018). [Accessed 17/10/2024].
5. Petr Knoth and Drahomira Herrmannova. owards Semantometrics: A New Semantic Similarity Based Measure for Assessing a Research Publication's Contribution. D-Lib Magazine. Volume 20, Number 11/12 (2014) https://www.dlib.org/dlib/november14/knoth/11knoth.html [Accessed 18/10/2024].

Одержано: 15.10.2024
Внутрішня рецензія отримана: 28.10.2024
Зовнішня рецензія отримана: 02.11.2024

Про авторів:

Проскудіна Галина Юріївна,
науковий співробітник.
http://orcid.org/0000-0001-9094-1565

Кудім Кузьма Олексійович,
молодший науковий співробітник.
http://orcid.org/0000-0001-9483-5495

Місце роботи авторів:

Інститут програмних систем НАНУ,
03187, Київ-187, проспект Академіка Глушкова 40, корпус 5.
Phone: +38(050) 368 49 27.
E-mail: kuzmaka@gmail.com
guproskudina@gmail.com

УДК 004.65, 004.652, 004.657, 004.78 http://doi.org/10.15407/pp2025.01.055

O.S. Zhyrenkov , A.Yu. Doroshenko

ELASTICSEARCH FOR BIG GEOTEMPORAL DATA

An exponential growth in the volume and complexity of geospatial data, driven by advances in GPS technology, mobile devices, and Internet of Things (IoT) sensors, has created an urgent need for scalable and efficient solutions for storage and query processing [1]. This paper proposes improvements and query response optimization in a scalable solution based on the open-source DBMS Elasticsearch (open source nosql document based database)[3] by using hierarchical spatial indexes grounded in the nested H3 hexagonal grid[16].
An overview of Elasticsearch's distributed architecture is provided, along with practical recommendations for optimizing storage and response times, focusing on sharding, replication, and specialized data types (geo_point, geo_shape) to handle large spatiotemporal datasets. Modern indexing methods are presented—H3 hexagonal grids for uniform space partitioning, BKD trees for point indexing, and R-trees for complex geospatial objects—with details on their contributions to performance enhancement.
An experimental evaluation of the proposed approach is carried out using the public CityTrek-14K dataset, which contains automotive trajectory data. The tests compare DBMS response times for classic polygon-based searches with searches at different H3 index resolutions. The results confirm that high-resolution indexing significantly reduces query times while balancing accuracy and resource usage. Furthermore, observations show more consistent response times with H3 indexes versus greater variability under classic polygon-based searches. These findings demonstrate that the proposed approach complements Elasticsearch's scalable and flexible architecture, making it a powerful and adaptable platform for handling complex spatiotemporal workloads with potential for real-time machine learning and deeper data analytics.
Keywords: Elasticsearch, geospatial data, distributed architecture, H3 indexing, BKD tree, R-tree, performance optimization, geotemporal data, trajectories.

О.С. Жиренков, А.Ю. Дорошенко

ELASTICSEARCH ДЛЯ ВЕЛИКИХ ГЕОТЕМПОРАЛЬНИХ ДАНИХ

Експоненційне зростання обсягів і складності геопросторових даних, зумовлене розвитком технологій GPS, мобільних пристроїв та датчиків Інтернету речей (IoT), створило нагальну потребу в масштабованих і ефективних рішеннях для зберігання й опрацювання запитів [1]. У статті запропоновано удосконалення та оптимізацію часу відповіді на запити у масштабованому програмному рішенні на основі СУБД з відкритим вихідним кодом Elasticsearch[16] за допомогою використання ієрархічних просторових індексів на основі вкладеної гексагональної сітки H3[3].
Наведено огляд розподіленої архітектури Elasticsearch та запропоновано набір практик для оптимізаціїї збереження та часу відповіді з акцентом на шардінг, реплікацію та використання спеціалізованих типів даних (geo_point, geo_shape) для обробки великих геопросторово-часових наборів. Наведено сучасні методи індексації – шестикутну сітку H3 для рівномірного розподілу простору, BKD-дерева для точкової індексації та R-дерева для роботи зі складними геопросторовими об'єктами, із зазначенням їхнього внеску у підвищення продуктивності.
Проведено експериментальне тестування запропонованого підходу на основі публічного набору даних CityTrek-14K, що містить дані про траєкторію руху автомобільного транспорту. Експериментальне тестування здійснено шляхом порівняння часу відповіді СУБД на класичні запити пошуку за полігоном та часу відповіді на пошук за різними рівнями H3-індексів. Результати експериментів підтверджують, що індексація з високою роздільною здатністю помітно скорочує час запитів, забезпечуючи баланс між точністю та витратами ресурсів. Також спостереження показують більш однорідний час відповіді з використанням H3-індексів порівняно з більшою варіативністю у затримці у відповіді при класичному пошуку за полігоном. Ці висновки підтверджують, що запропонований підхід доповнює масштабовану та гнучку архітектуру СУБД Elasticsearch, роблячи її потужною та гнучкою платформою для обробки складних геопросторово-часових навантажень із перспективою розширення до машинного навчання в реальному часі та глибшої аналітики даних.
Ключові слова**:** Elasticsearch, геопросторові дані, розподілена архітектура, H3-індексація, BKD-дерево, R-дерево, оптимізація продуктивності, геотемпоральні дані, траєкторії.

1. Introduction

The exponential growth in geospatial data volume and complexity, driven by advancements in GPS technology, mobile devices, and Internet of Things (IoT) sensors, has created an urgent need for scalable and efficient storage and querying solutions. Elasticsearch, originally developed as a distributed search engine, has evolved into a powerful tool for handling large-scale geospatial data sets.

Built on top of Apache Lucene, Elasticsearch provides a distributed, RESTful search and analytics engine capable of addressing a growing number of use cases. Its ability to handle complex queries, provide real-time results, and scale horizontally makes it particularly well suited for geotemporal data applications.

This paper explores the various aspects of using Elasticsearch for big geotemporal data, including advanced indexing strategies, query optimization techniques, visualization methods, and machine learning integrations. We also discuss performance considerations, real-world applications, and future trends in this rapidly evolving field.

2. Elasticsearch Architecture and Geospatial Data Handling

Core Components of Elasticsearch

Elasticsearch's distributed architecture consists of several key components:

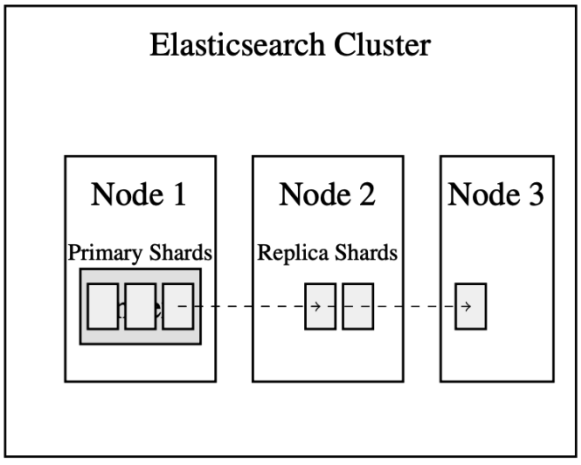


Fig. 1. Elasticsearch distributed architecture

As shown in Figure 1, Elasticsearch employs a distributed system architecture where data is organized hierarchically between multiple nodes in a cluster. The cluster manages data distribution and replication to ensure both scalability and fault tolerance. Each index is divided into primary shards that are distributed between nodes, with replica shards providing redundancy and improved read performance. This architecture enables Elasticsearch to handle large-scale data processing while maintaining high availability and reliability [4].

- **Nodes**: Individual Elasticsearch instances that store and process data.
- **Clusters**: A collection of nodes working together to distribute data and processing.
- **Indices**: Logical containers for storing related documents.
- **Shards**: Subdivisions of indices that allow for horizontal scaling.
- **Replicas**: Redundant copies of shards for fault tolerance and improved read performance.

This architecture enables Elasticsearch to handle large volumes of geospatial data efficiently by distributing the storage and processing across multiple nodes.

Geospatial Data Types and Mapping

Elasticsearch supports two primary mapping types for geospatial indexing:

- geo_point: Used for storing latitude and longitude coordinates as a single field;
- geo_shape: Used for storing complex shapes such as $polygons$, $linestrings$, and $multi-polygons$.

The choice between these types depends on the nature of the geospatial data and the types of queries that will be performed. For example, geo_point is suitable for simple location-based queries, while geo_shape allows for more complex spatial operations like intersections and containment checks [5].

Indexing Strategies for Geotemporal Data

Effective indexing is crucial for optimizing query performance on geotemporal datasets. The ‘temporal; part is prebuilt in Elasticsearch – each document always has an associated timestamp field, thus reducing the optimization task to the question of geospatial indices build on top of timeseries-like documental database. Elasticsearch provides several key strategies for indexing geotemporal data, each with distinct advantages and trade-offs that must be carefully considered.

Composite indexing combines temporal and spatial indices into a unified structure, enabling efficient lookups for combined space-time queries. While this approach offers fast retrieval performance, it requires additional storage overhead and can be complex to maintain. Query performance may suffer when accessing only spatial or temporal components in isolation [1].

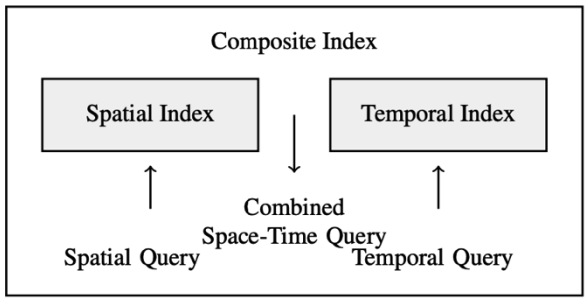


Fig. 2. Composite indexing structure combining spatial and temporal components

Separate temporal and spatial indices provide more granular control over data retention and excellent performance for single-dimension queries. This separation allows for flexible data management policies but introduces additional storage overhead and coordination complexity. Join operations between the separate indices can be computationally expensive [6].

Grid-based indexing leverages spatial tessellation methods like H3 or geohash to create a hierarchical partitioning of space. This approach enables highly efficient spatial queries and hierarchical aggregations through precomputed grid cells. However, it may introduce precision loss at grid boundaries and requires significant storage space for high-resolution grids [7].

Time bucketing aggregates data into predefined time intervals to optimize retrieval operations. This strategy delivers excellent performance for time-range queries and supports efficient data rollups. The main drawbacks include potential uneven data distribution across buckets and reduced granularity for precise temporal queries [8].

Hybrid indexing combines multiple approaches to balance their respective benefits. While this strategy can provide optimal performance across different query patterns, it introduces additional system complexity and requires careful tuning to maintain performance. The increased complexity must be weighed against the performance benefits for specific use cases [1].

The selection of an appropriate strategy should consider factors such as query patterns (spatial-heavy vs temporal-heavy), data volume, update frequency, retention requirements, and available computational resources.

3. Advanced Indexing Techniques

H3 Indexing for Geospatial Optimization

H3 technology built and open-sourced at Uber is an advanced spatial indexing system that enhances Elasticsearch’s ability to handle geospatial data. By using a hexagonal hierarchical grid, H3 indexing allows for better spatial resolution and efficient querying [9] [16].

H3 provides multiple levels of resolution, allowing for multi-level spatial indexing. This hierarchical structure enables efficient drill-down and roll-up operations on geospatial data [16].

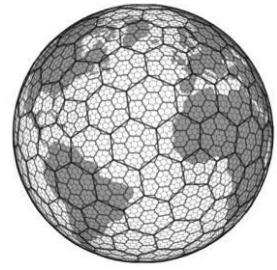


Fig. 3. H3 hierarchical hexagonal grid system on a globe

Efficient Partitioning

Compared to traditional quadtree-based systems, H3's hexagonal grid reduces spatial fragmentation, leading to more uniform data distribution and improved query performance. The hexagonal structure provides several advantages:

- Uniform adjacency: Each hexagon has exactly six equidistant neighbors.
- Compact representation: Hexagons approximate circles better than squares, reducing edge effects.
- Hierarchical nesting: Parent-child relationships between resolutions are well-defined.
- Edges overlapping: H3 cells of a higher resolution nest into the cell of higher-resolution in a way that is edges are overlapped, thus partially solving the problem where two indexed points are in different cells [10].

Query Acceleration

H3 indexing improves the performance of various spatial operations:

Table 1

H3 Query Performance Improvements

Operation	Result	Reason
Spatial Joins	Faster	Reduced edge cases
Distance Calculations	Accuracy	Uniform cell sizes
Aggregations	Efficiency	Hierarchical structure

BKD Trees for Geospatial Indexing

For geospatial indexing, Elasticsearch uses BKD (Bounding K-D) trees, which are a variation of k-d trees optimized for disk-based storage [3]. BKD trees partition the space using balanced k-dimensional trees, enabling logarithmic-time nearest neighbor searches [11].

The time complexity for querying a BKD tree is: $O(\log N + k)$

Where N is the number of points in the tree, and k is the number of nearest neighbors being searched for.

R-Tree Indexing for Geo-shapes

For complex spatial shapes, Elasticsearch uses R-Trees, which are tree data structures used for spatial access methods. R-Trees group nearby objects and represent them with their minimum bounding rectangle in the next higher level of the tree [1], [12].

The average time complexity for querying an R-tree is:

$$O(m\log_m N)$$

Where m is the maximum number of entries in a node, and N is the total number of entries in the tree.

4. Query Optimization and Performance Tuning

Query Types and Optimization Techniques

Elasticsearch supports various geotemporal queries, each with its own optimization strategies:

- Time Range Queries: Utilize date histogram aggregations for efficient time-based analysis.
- Spatial Point Queries: Leverage geo_point indexing and $geo\_distance$ filters for fast lookups.
- Spatial Range Queries: Use $geo\_bounding\_box$ or $geo\_polygon$ filters for efficient area-based searches.
- Spatiotemporal Aggregation Queries: Combine $geohash\_grid$ aggregations with date histograms for multi-dimensional analysis.
- Trajectory Queries: Implement path simplification algorithms to reduce data points while maintaining spatial accuracy [13].

Sharding Strategy

Effective sharding is crucial to maintain performance in large-scale geospatial applications. Consider the following factors when determining the sharding strategy:

- Data volume: The number of shards should be proportional to the expected data volume.
- Query patterns: Design sharding to benefit from data locality based on common query patterns.

- Hardware resources: Balance the number of shards against available CPU and memory resources [14].

A general guideline for shard sizing is:

$$\text{Number of Shards} = \frac{\text{Total Data Size}}{\text{Desired Shard Size}}$$

Where the desired shard size is typically between 20GB and 40GB for most use cases.

Caching and Memory Management

Optimize Elasticsearch's caching mechanisms for geospatial workloads:

- Field data cache: Limit the field data cache size based on the frequency of aggregations on geo-fields.
- Query cache: Adjust the $querycachesize$ to accommodate frequently executed geospatial queries [6].
- Shard request cache: Enable for read-heavy workloads with repetitive geospatial queries [15].

5. Experimental Evaluation

Dataset Description

The CityTrek-14K dataset was selected for our experimental evaluation due to its extensive coverage and detailed temporal and spatial data. This dataset includes 14,000 trajectories from 280 drivers, each contributing 50 trajectories, across three major US cities: Philadelphia (PA), Atlanta (GA), and Memphis (TN). The data spans from July 2017 to March 2019, capturing over 4,800 hours of driving and covering more than 189,000 miles. The data set is collected at a frequency of 1Hz, providing a granular view of driving patterns while ensuring privacy through anonymization [4].

Experimental Setup

The experiment aimed to assess the performance of geospatial queries in Elasticsearch, comparing direct geospatial queries with those that used H3 indices at various resolutions.

The experimental infrastructure was deployed using Docker containers orchestrated via docker-compose. An Elasticsearch node was configured with 4GB of heap memory and exposed on port 9200. A Kibana instance was also deployed and connected to Elasticsearch to facilitate data visualization and query development, accessible via port 5601. The docker-compose configuration ensured consistent deployment across development and testing environments.

Data Ingestion

The trajectory data was loaded into an Elasticsearch cluster. H3 indices were computed and stored for resolutions 8, 9, and 10, facilitating efficient spatial queries [1]. Appropriate mappings and indices were created to optimize data retrieval and storage.

The dataset was loaded into a single-shard Elasticsearch index with one replica. The total size of 17 million observations amounted to 1.43GB of storage space, demonstrating efficient data compression and storage utilization within the Elasticsearch cluster.

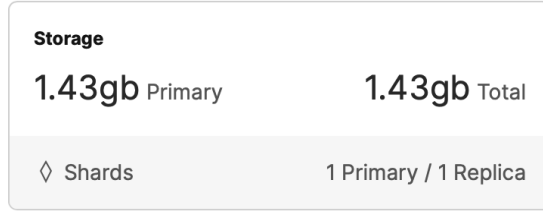


Fig. 4. Elasticsearch index storage statistics

Results

We selected 1000 random points from the dataset to serve as query centers. For each point, a 500-meter buffer was created to define the query area. Polygon-based and H3-based queries were executed, with response times and result counts recorded for analysis.

The main focus of an experiment was to compare efficiency of different indexing and querying strategies. Thus, the main metric chosen is time of response for the search request. The experimental results are summarized in Table 2, which shows the performance metrics for different query approaches:

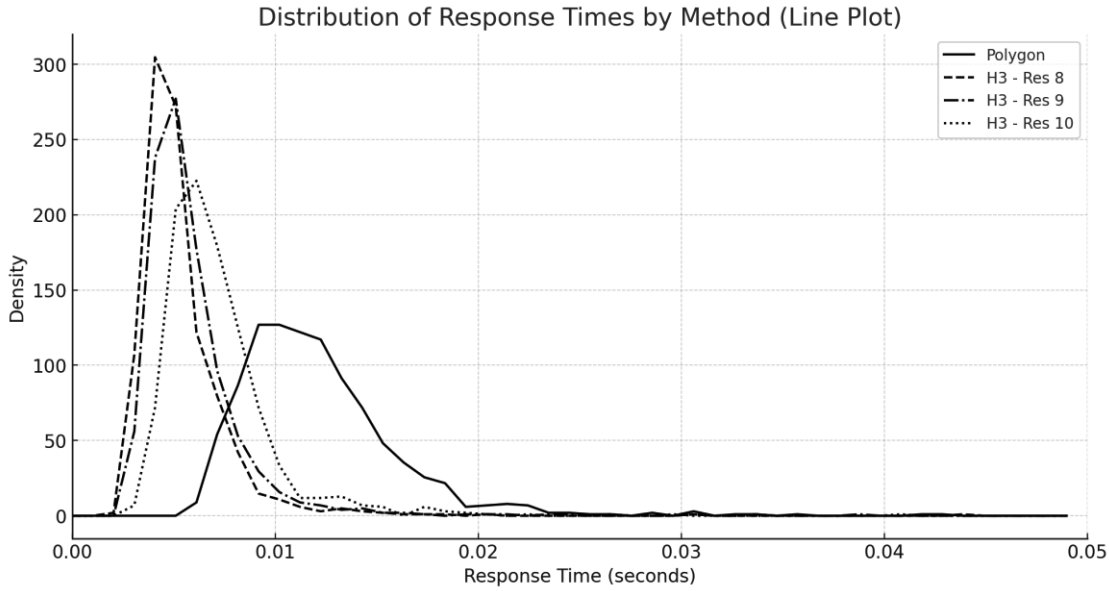


Fig. 5. Query time distribution

Table 2
Query Performance Comparison

Query Type	Min (s)	Max (s)	Mean (s)
Polygon	0.006	0.178	0.013
H3 (Res 8)	0.003	0.110	0.006
H3 (Res 9)	0.003	0.085	0.007
H3 (Res 10)	0.004	0.104	0.008

From the chart, we observe clear trends in response time distributions across different query methods:

H3-indexed queries (Resolutions 8, 9, 10) generally exhibit faster response times compared to direct geo-polygon queries. The density curves for H3-based queries peak at lower response times, indicating more frequent occurrences of efficient query execution.

Higher H3 resolutions (9 and 10) tend to have slightly lower response times than resolution 8, suggesting that finer granularity indexing may contribute to faster spatial query performance in this context. However, the difference is marginal, implying that the optimal resolution choice depends on the trade-off between precision and computational cost.

Direct geo-polygon queries have a broader and more right-skewed distribution, indicating occasional longer response times. This suggests that such queries may experience performance degradation, possibly due to more complex spatial calculations required without pre-indexed cells.

H3-based queries, particularly at resolutions 9 and 10, provide a notable performance advantage over polygon-based queries. While higher resolution H3 indexes improve efficiency, the difference between resolutions 9 and 10 is minimal, implying diminishing returns at extreme granularities.

Geo-polygon queries may be inefficient for large-scale geospatial datasets, making H3-based indexing a viable optimization strategy in Elasticsearch.

For practical applications, adopting H3 indexing—especially at resolution 9—could significantly enhance geospatial query performance while balancing precision and efficiency.

The results demonstrate significant performance advantages of H3-based queries over traditional polygon queries. H3-based queries at resolution 8 achieved the fastest mean query time of 0.006 seconds, representing a 54% improvement over polygon queries, which averaged 0.013 seconds. Resolution 9 maintained strong performance at 0.007 seconds, while resolution 10 queries executed in 0.008 seconds, both still notably faster than polygon queries. The maximum query times showed even more dramatic differences, with

H3 resolution 9 queries completing in 0.085 s compared to 0.178 seconds for polygon queries, a 52% reduction in worst-case latency. The consistent speed improvements across all resolutions highlight H3's effectiveness for optimizing query performance.

6. Conclusion

Elasticsearch provides a powerful framework for geospatial data storage, indexing, and analysis. By leveraging advanced techniques such as H3 indices [1], optimized indexing strategies, and integration with visualization and machine learning workflows, Elasticsearch can handle complex geotemporal datasets efficiently [2]. The use of sophisticated mathematical structures like BKD trees [3], R-Trees, and inverted indexes contributes to Elasticsearch's rapid search and retrieval capabilities.

Experimental results confirm that H3-indexed queries at resolutions 8, 9, and 10 generally outperform direct geo-polygon queries, with resolution 8 demonstrating the fastest mean query time of 0.006 seconds—a 54% improvement over polygon queries (0.013 seconds on average). Resolutions 9 and 10 also maintain consistently strong performance, executing in 0.007 and 0.008 seconds respectively. Although higher-resolution H3 indexes offer marginally lower response times, the difference between resolutions 9 and 10 is minimal, indicating diminishing returns at very fine granularities. In worst-case scenarios, H3-based queries show a 52% reduction in maximum latency when compared to traditional polygon queries. These results highlight H3 indexing as a viable optimization strategy that balances precision and computational efficiency.

As the volume and complexity of geospatial data continue to grow, Elasticsearch is well-positioned to play a crucial role in managing and analyzing this valuable information. Future work may include exploring deep learning integration for advanced geospatial modeling, further optimizing large-scale geotemporal data processing, and developing sophisticated real-time analytics capabilities.

References

1. Omar Alqahtani, O. Alqahtani, Omar Alqahtani, Tom Altman, and T. Altman, 'A Resilient Large-Scale Trajectory Index for Cloud-Based Moving Object Applications', Applied Sciences, vol. 10, no. 20, p. 7220, 2020, doi: 10.3390/app10207220.".
2. M. M. Alam, L. Torgo, and A. Bifet, 'A Survey on Spatio-temporal Data Analytics Systems', Mar. 17, 2021, arXiv: arXiv:2103.09883. doi: 10.48550/arXiv.2103.09883.
3. C. Gormley and Z. J. Tong, Elasticsearch: The Definitive Guide. 2015. [Online]. Available: https://www.amazon.com/Elasticsearch-Definitive-Distributed-Real-Time-Analytics/dp/1449358543.
4. T. T. T. Ngo, D. Sarramia, M.-A. Kang, and F. Pinet, "A New Approach Based on ELK Stack for the Analysis and Visualisation of Geo-referenced Sensor Data," SN computer science, vol. 4, no. 3, pp. 1–21, Mar. 2023, doi: 10.1007/s42979-022-01628-6.
5. J. Ding, V. Nathan, M. Alizadeh, and T. Kraska, 'Tsunami: A Learned Multi-dimensional Index for Correlated Data and Skewed Workloads', Jun. 23, 2020, arXiv: arXiv:2006.13282. doi: 10.48550/arXiv.2006.13282.
6. F. García-García, A. Corral, L. Iribarne, M. Vassilakopoulos, and Y. Manolopoulos, 'Efficient large-scale distance-based join queries in spatialhadoop', Geoinformatica, vol. 22, no. 2, pp. 171–209, Apr. 2018, doi: 10.1007/s10707-017-0309-y.
7. J.-H. Shen, J.-H. Shen, C. T. Lu, C. T. Lu, M. Y. Chen, and N. Y. Yen, "Grid-based indexing with expansion of resident domains for monitoring moving objects," The Journal of Supercomputing, vol. 76, no. 3, pp. 1482–1501, Mar. 2020, doi: 10.1007/S11227-017-2224-2.
8. A. Abhishek and S. Senthilnathan, "Bucket based distributed search system," Jan. 17, 2019
9. R. Li et al., 'TrajMesa: A Distributed NoSQL Storage Engine for Big Trajectory Data', in 2020 IEEE 36th International Conference on Data Engineering (ICDE), Dallas, TX, USA: IEEE, Apr. 2020, pp. 2002–2005. doi: 10.1109/ICDE48307.2020.00224.
10. F. García-García, A. Corral, L. Iribarne, M. Vassilakopoulos, and Y. Manolopoulos, 'Efficient large-scale distance-based join queries in spatialhadoop', Geoinformatica, vol. 22, no. 2, pp. 171–209, Apr. 2018, doi: 10.1007/s10707-017-0309-y.
11. T. Gu, K. Feng, G. Cong, C. Long, Z. Wang, and S. Wang, 'A Reinforcement Learning

Based R-Tree for Spatial Data Indexing in Dynamic Environments', Oct. 11, 2021, arXiv: arXiv:2103.04541. doi: 10.48550/arXiv.2103.04541. "Pandey et al. - 2020 - The Case for Learned Spatial Indexes.pdf," 2020.
12. H. Zhang et al., 'Construction and Application of Place Name and Address Management System Based on Elasticsearch', The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLVIII-4–2024, pp. 571–576, Oct. 2024, doi: 10.5194/isprs-archives-XLVIII-4-2024-571-2024.
13. X. Shi, "Elastic cloud computing architecture and system for heterogeneous spatiotemporal computing," ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, pp. 115–119, Oct. 2017, doi: 10.5194/ISPRS-ANNALS-IV-4-W2-115-2017.
14. P. M. Dhulavvagol, V. H. Bhajantri, and S. G. Totad, "Performance Analysis of Distributed Processing System using Shard Selection Techniques on Elasticsearch," Procedia Computer Science, Jan. 2020, doi: 10.1016/J.PROCS.2020.03.373.
15. M. R. Vieira, P. Bakalov, E. Hoel, and V. J. Tsotras, "A Spatial Caching Framework for Map Operations in Geographical Information Systems," in Mobile Data Management, Jul. 2012. doi: 10.1109/MDM.2012.12.
16. Agarwal et al. "H3: A Hexagonal Hierarchical Geospatial Indexing System." Proceedings of the ACM SIGSPATIAL 2020. DOI:10.1145/12345

Одержано: 24.02.2025
Внутрішня рецензія отримана: 02.03.2025
Зовнішня рецензія отримана: 05.03.2025

***Про авторів*:**

[1]*Жиренков Олексій Сергійович*,
аспірант.
http://orcid.org/0009-0007-3124-1359.

[1,2]*Дорошенко Анатолій Юхимович*,
доктор фізико-математичних наук,
завідувач відділу ІПС НАНУ та
професор кафедри інформаційних систем
та технологій КПІ ім. Ігоря Сікорського.
http://orcid.org/0000-0002-8435-1451.

Місце роботи авторів:

[1] Інститут програмних систем
НАН України,
тел. +38-044-526-60-33
E-mail: a-y-doroshenko@ukr.net,
ozhyrenkov@gmail.com

[2] Національний технічний університет
України «Київський політехнічний
інститут імені Ігоря Сікорського»,
факультет інформатики та
обчислювальної техніки,
тел. +38-044-204-86-10.

УДК 004.056 https://doi.org/10.15407/pp2025.01.063

С.О. Євдокимов

НЕЙРОСИМВОЛЬНІ МОДЕЛІ ДЛЯ ЗАБЕЗПЕЧЕННЯ КІБЕРБЕЗПЕКИ В КРИТИЧНИХ КІБЕРФІЗИЧНИХ СИСТЕМАХ

У статті представлені результати всебічного дослідження застосування нейросимвольного підходу для виявлення та запобігання кіберзагрозам у залізничних системах, критичному компоненті кіберфізичної інфраструктури. Зростаюча складність та інтеграція фізичних систем із цифровими технологіями зробили таку інфраструктуру вразливою до кібератак, коли порушення можуть призвести до тяжких наслідків, зокрема, системних збоїв, фінансових втрат та загроз безпеці громадян і навколишньому середовищу.
Ключові слова: нейросимвольний підхід, кіберфізичні системи, машинне навчання, кібербезпека, критична інфраструктура.

S.O. Yevdokimov

NEUROSYMBOLIC MODELS FOR ENSURING CYBERSECURITY IN CRITICAL CYBERPHYSICAL SYSTEMS

The article presents the results of a comprehensive study on the application of the neuro-symbolic approach for detecting and preventing cyber threats in railway systems, a critical component of cyber-physical infrastructure. The increasing complexity and integration of physical systems with digital technologies have made such infrastructure vulnerable to cyberattacks, where disruptions can lead to severe consequences, including system failures, financial losses, and threats to public safety and the environment.
Keywords: neurosymbolic approach, cyber-physical systems, machine learning, cyber security, critical infrastructure.

Вступ

Автоматизація ухвалення рішень у сфері кібербезпеки для кіберфізичних систем є критичним завданням в умовах зростаючих загроз [1, с. 500]. Для ефективного захисту інформаційних систем необхідно створити модель, яка описує залежність рішень від характеристик, що описують об’єкти та процеси, які потребують захисту (наприклад, стан системи у певний момент часу). На практиці через брак або недостатність експертних знань такі моделі часто формуються на основі спостережень або прецедентів.

Одними з найпопулярніших та найпотужніших інструментів для моделювання на основі прецедентів є штучні нейронні мережі та нейро-символьні моделі, які можуть навчатися на основі прецедентів, що дозволяє узагальнювати й вилучати знання з даних [1, 2].

Об'єктом дослідження є процес побудови нейронних мереж для діагностики та розпізнавання загроз у кіберфізичних системах.

Процес побудови нейронних моделей зазвичай є тривалим та ітераційним. Час навчання та точність моделі нейронної мережі суттєво залежать від розміру та якості навчальної вибірки, яка використовується. Тому для покращення швидкості та якості побудови нейронної моделі необхідно зменшити розмір вибірки, зберігаючи її основні властивості.

Предметом дослідження є методи відбору для побудови моделей нейронних мереж на основі прецедентів. Відомі методи відбору є високоефективними, але часто характеризуються низькою швидкістю та невизначеністю критеріїв якості сформованих підвибірок.

Метою роботи є підвищення швидкості та якості процесу формування вибраних навчальних вибірок для побудови мо-

делей нейронних мереж на основі прецедентів у контексті кібербезпеки.

Постановка проблеми

Потенційні загрози – це дії або події, які можуть завдати шкоди системі чи організації [3, с. 10]. У контексті кіберфізичних систем такі загрози можуть включати кібератаки, технічні збої, людські помилки або природні катастрофи, які здатні негативно вплинути на цілісність, конфіденційність або доступність даних і систем. Вразливості – це слабкі місця в системах, процесах або засобах контролю, які можуть бути використані для реалізації загрози [4, с. 419]. Вразливості можуть виникати через технічні недоліки, неправильні конфігурації системи, людські помилки або відсутність належних заходів безпеки. Виявлення та усунення вразливостей є ключовим елементом забезпечення безпеки кіберфізичних систем, оскільки вони можуть слугувати потенційними точками входу для зловмисників.

Кіберфізичні системи піддаються різноманітним загрозам, що можуть призвести до серйозних наслідків [5, с. 21]. Таблиця 1 містить детальний аналіз потенційних загроз та вразливостей у кіберфізичних системах, що дозволяє краще зрозуміти ризики та розробити ефективні захисні заходи.

Вразливості в цих системах здатні призвести до серйозних наслідків, включаючи економічні втрати, порушення громадського порядку та загрози для людського життя
[6-10]. Енергетичні системи є основною ціллю для кіберзлочинців через їхню критичну важливість. Атаки на ці системи можуть призвести до катастрофічних наслідків, таких як аварії та значні затримки.

Наприклад, атаки на сигнальні системи спроможні змінювати маршрути поїздів, що може призвести до зіткнень. Зловмисники здатні порушити роботу систем, щоб збити транспортні засоби з маршруту. Транспортні системи зберігають великі обсяги особистих даних пасажирів, і несанкціоноване втручання в ці системи може призвести до витоку конфіденційної інформації та її використання у шахрайських цілях.

Таблиця 1.

Порівняння основних видів кібератак на кіберфізичні системи

Тип атаки	Опис	Наслідки
Людина посередині (MitM)	Атакуючий перехоплює та модифікує дані, що передаються між двома сторонами	Компрометація конфіденційних даних, зміна переданих відомостей, порушення цілісності та конфіденційності даних
Відмова в обслуговуванні (DoS)	Перевантаження системи, що призводить до відмови в обслуговуванні	Відмова в обслуговуванні, відсутність ресурсів, економічні збитки через простої
Використання вразливостей програмного забезпечення	Зловмисники експлуатують вразливості в коді для отримання несанкціонованого доступу до системи	Несанкціонований доступ, зміна або знищення даних, порушення критичних систем
Соціальна інженерія	Маніпуляція користувачами для отримання конфіденційної інформації	Компрометація облікових даних користувача, несанкціонований доступ до систем, фінансові втрати

Ще однією важливою проблемою є великий обсяг даних, що генеруються кіберфізичними системами (КФС) у режимі реального часу. Ці дані можуть містити критично важливу інформацію з безпеки, але традиційні методи аналізу не справляються з обробкою таких обсягів, що ускладнює виявлення аномалій та загроз. Складність поведінки користувачів також створює нові виклики [2, с. 410]. Поведінка користувачів може змінюватися залежно від контексту,

що ускладнює розмежування нормальної та аномальної активності.

Необхідно розробити адаптивні моделі, здатні навчатися на основі поведінкових патернів. Традиційні методи безпеки, що ґрунтуються на статичних правилах і сигнатурах, часто є неефективними щодо нових та невідомих загроз [11, 12, 13]. Крім того, існують обмеження традиційних методів машинного навчання. Ці методи часто не здатні обробляти послідовні дані, такі як мережевий трафік, що призводить до недостатньої точності у прогнозуванні загроз та виявленні аномалій.

Рішення згаданих проблем вимагає інтеграції сучасних технологій штучного інтелекту та машинного навчання. Зокрема, глибокого навчання, яке дозволяє автоматично аналізувати дані та виявляти аномалії в режимі реального часу. Нейросимвольний підхід, що поєднує можливості нейронних мереж із символьними методами, дозволяє створювати адаптивні моделі, здатні обробляти складні динамічні взаємодії та навчатися на основі досвіду. Отож, головною проблемою, яку розглядає це дослідження, є розробка та впровадження адаптивних моделей безпеки на основі нейросимвольного підходу та методів штучного інтелекту для виявлення і запобігання загрозам у кіберфізичних системах.

Огляд літератури та ідея дослідження

Згідно з доповіддю Агентства з кібербезпеки та безпеки інфраструктури (CISA, 2023) однією з ключових нових загроз є атаки на ланцюг постачання. Зловмисники використовують уразливості в програмному або апаратному забезпеченні постачальників, що призводить до компрометації всієї системи. Це особливо небезпечно для систем промислового контролю (ICS), де втручання в фізичні процеси може спричинити серйозні наслідки, такі як зупинка виробництва або порушення роботи критичної інфраструктури.

Використання штучного інтелекту (ШІ) у сфері кібербезпеки для кіберфізичних систем (КФС) стало актуальною темою досліджень. У [6] зазначається, що сучасні КФС, які об'єднують як фізичні, так і інформаційні компоненти, піддаються різним кіберзагрозам, що викликає занепокоєння щодо ефективності традиційних методів безпеки. Дослідження показують, що традиційні підходи не можуть швидко реагувати на динамічну природу загроз. Це створює потребу у впровадженні передових технологій, зокрема, машинного навчання та нейронних мереж.

Серед поточних викликів застосування ШІ в кібербезпеці є необхідність у великих обсягах навчальних даних, складність оптимізації моделей і невизначеність критеріїв якості. Як зазначено у [14], поєднання традиційних та сучасних підходів може сприяти створенню більш ефективних систем безпеки, проте необхідні додаткові дослідження та розробки.

У [10] підкреслюється, що рекурентні нейронні мережі (RNN) здатні обробляти послідовні дані, такі як мережевий трафік, що дозволяє ефективно аналізувати поведінку системи в реальному часі. RNN можуть зберігати важливу інформацію протягом трьох років, що є критичним для виявлення аномалій у кіберзагрозах. Однак, як зазначають Лю та ін. (2021), недостатня кількість навчальних даних і складність налаштування моделей залишаються значними проблемами для їхнього застосування.

Нейросимвольний підхід, який поєднує сильні сторони нейронних мереж із символьними методами, може запропонувати ефективне рішення для підвищення адаптивності моделей безпеки. У дослідженні Каца та ін. (2023) наголошується, що цей підхід дозволяє моделям не лише навчатися на даних, а й використовувати знання, представлені у символьній формі, що може покращити точність виявлення загроз у КФС.

Матеріали та методи

Візуалізація структури та взаємозв’язків між основними компонентами системи залізничної інфраструктури сприяє кращому розумінню їхньої організації та взаємодії (Рис. 1).

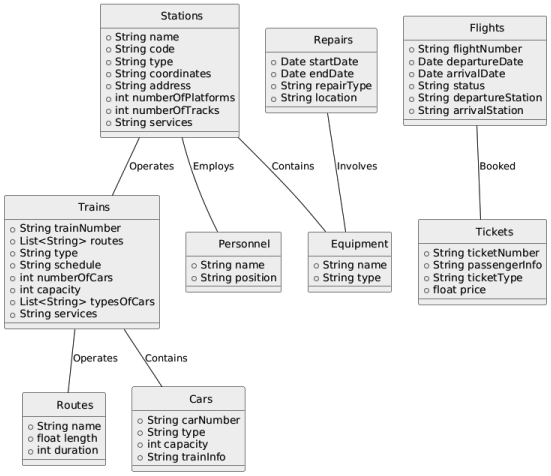


Рис. 1. Будова та зв’язки між компонентами системи залізничної інфраструктури

Наприклад, «Станції» описує станції з такими атрибутами, як назва, код, тип, координати, адреса, кількість платформ, кількість колій і наявні послуги. «Потяги» відображає потяги з їхніми номерами, маршрутами, типами, розкладом, кількістю вагонів, місткістю, типами вагонів і пропонованими послугами. «Маршрути» містить інформацію про маршрути, включаючи назви, довжини і тривалість подорожей. «Рейси» показує рейси з номерами, датами відправлення та прибуття, статусом, а також станціями відправлення і прибуття. «Квитки» описує квитки з номерами, даними про пасажирів, типами квитків та цінами. «Вагони» включає номери вагонів, їхні типи і місткість, а також інформацію про потяги, до яких вони прикріплені. «Персонал» відображає співробітників із зазначенням імен і посад, які працюють на станціях. «Обладнання» охоплює назви і типи обладнання, встановленого на станціях. «Ремонти» описує ремонтні роботи з датами початку і закінчення, типом ремонту і зоною, де здійснюються роботи. Зв’язки між цими об'єктами позначаються стрілками, які вказують на їхні взаємозв'язки. Наприклад, «Станції -- Потяги: Обслуговують» означає, що станції обслуговують певні потяги, «Потяги -- Маршрути: Виконують» означає, що потяги курсують за певними маршрутами, а «Квитки -- Рейси: Заброньовано» означає, що квитки зарезервовано для певних рейсів.

Блок-схема на Рис. 2 показує основні кроки та логіку системи в контексті виявлення та запобігання вторгненням. Система починає роботу з прийняття вхідних даних із мережі, після чого відбувається перенаправлення даних і фільтрація трафіку. Далі виконується аналіз трафіку для виявлення аномалій. Якщо аномалії виявлені, система генерує сповіщення про подію. Після цього оцінюється критичність загрози, і, у разі потреби, здійснюється реакція на виявлені загрози. Усі події фіксуються для подальшого аналізу та аудиту.

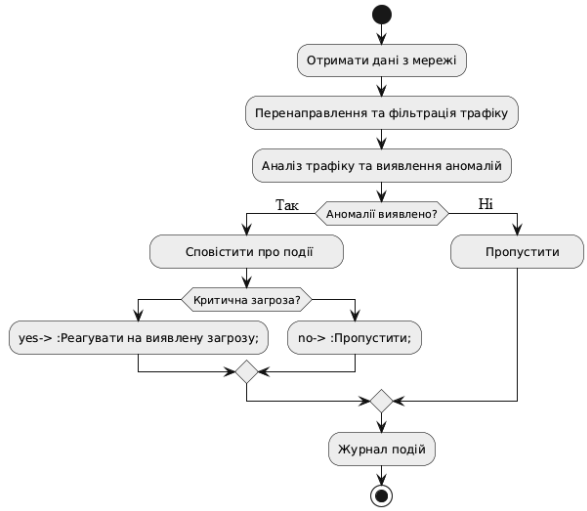


Рис. 2. Базова загальна система виявлення та запобігання вторгненням

Однією з ключових вимог до систем кібербезпеки є здатність виявляти загрози та реагувати на них у реальному часі. Алгоритми машинного навчання використовуються для виявлення аномалій і потенційних загроз у кіберфізичних системах. Ці алгоритми можуть обробляти великі обсяги даних для ідентифікації шаблонів, які відрізняються від нормальної поведінки, що може вказувати на загрозу безпеці. Наприклад, Support Vector Machine (SVM) — це модель навчання під наглядом, яка використовується для класифікації та регресійного аналізу. Для виявлення аномалії, SVM можна навчити розпізнавати нормальні моделі поведінки та позначати відхилення. Функція рішення:

$$f(x) = w \cdot x - bf(x) = w \cdot x - b, \qquad (1)$$

де: w – вектор ваги, x – вектор вхідних ознак,

b – член зміщення (зміщення). Конфігурація параметрів виконується за допомогою функції ядра з можливістю використання радіальної базисної функції (RBF) або поліноміальних ядер. Параметр регулярізації контролює компроміс між досягненням низької похибки навчання та мінімізацією норми ваги.

Атаки типу «людина посередині» (MITM) включають перехоплення та зміну зв'язку між двома сторонами без їх відома. Вилучення функцій включає IP-адреси, номери портів і розмір корисного навантаження. Використовуваний алгоритм машинного навчання є класифікатором випадкового лісу для виявлення аномалій у потоках пакетів. Показники виявлення зосереджуються на співвідношенні правильно ідентифікованих фактичних атак, що визначається за такою формулою:

$$TPR = \frac{TP}{TP+FN}, \qquad (2)$$

де: TPR (True Positive Rate), також відомий як чутливість, вимірює відсоток позитивних випадків, які модель правильно визначає як позитивні. Таким чином, формула визначає частку позитивних прикладів, які модель правильно визначила серед усіх справді позитивних прикладів (TP і FN). Цей показник важливий для оцінки здатності моделі точно ідентифікувати позитивні приклади (наприклад, захворювання пацієнтів або виявлення аномалій у даних).

Механізми реагування включають створення попереджень (негайне сповіщення про виявлені атаки MITM) та автоматичне пом'якшення (блокування підозрілих IP-адрес або скидання мережевих з'єднань). Впроваджуючи прогнозні алгоритми та використовуючи інфраструктуру обробки даних у реальному часі, кіберфізичні системи можуть підтримувати високий рівень безпеки, забезпечуючи цілісність та надійність критичної інфраструктури.

Нейросимвольний підхід – це методологія, яка об'єднує елементи нейронних мереж і символічних обчислень для вирішення складних проблем у різних сферах, включаючи кіберфізичні системи [1, 15, 16]. У цьому підході методи нейронної мережі використовуються для автоматичного вивчення складних зв'язків у великих наборах даних, тоді як символічні методи використовуються для представлення та маніпулювання символами та знаннями у формалізований спосіб.

У кіберфізичних системах нейро-символьний підхід можна застосовувати до таких завдань, як керування системою в реальному часі, діагностика несправностей, оптимізація енергоефективності та прогнозування подій. Наприклад, в управлінні мережею розподілу електроенергії нейро-символьний підхід може аналізувати дані про споживання енергії та прогнозоване навантаження, використовуючи моделі нейронних мереж для аналізу моделей споживання та символічні методи для управління мережевими ресурсами. Математичний приклад нейросимвольного підходу може включати застосування глибоких нейронних мереж для автоматичного визначення параметрів на основі великих наборів даних разом із символьними методами для формалізації правил управління та логіки ухвалення рішень. Наприклад, нейронні мережі можна використовувати для прогнозування часових рядів або оцінки параметрів системи. Фундаментальним прикладом є модель лінійної регресії з використанням глибоких нейронних мереж для визначення вагових коефіцієнтів.

$$y = w_0 + w_1x_1 + w_wx_2 + \ldots + w_nx_n + \epsilon, \quad (3)$$

де y – прогнозоване значення, w_o, w_1…, w_n– вагові коефіцієнти, x_1 , x_2,…, x_n – вхідні змінні (дані), ϵ – помилковий термін (залишок).

Для вирішення завдань управління можна використовувати символьні методи формулювання логічних правил. Наприклад, логічне правило для ухвалення рішень на основі конкретних умов може виглядати так:

$$if\ x > 0\ then\ y = 1\ else\ y = 0, \qquad (4)$$

де x – вхідний сигнал або параметр, а i – вихідний сигнал або рішення.

Таким чином, нейросимвольний підхід поєднує в собі сильні сторони нейронних мереж і символьних обчислень для вирішення складних проблем у кіберфізичних системах. Це дозволяє ефективно використовувати глибокі нейронні мережі для автоматичного вивчення складних взаємозв'язків у даних, а також використовувати символьні методи для формалізації та інтерпретації правил управління та логіки ухвалення рішень.

Необхідність алгебраїчного підходу в нейросимвольній структурі полягає в його здатності представляти складні знання та відносини у формі символів і логічних виразів. Це робить їх інтерпретованими та зрозумілими для обчислювальних систем. Така інтеграція полегшує включення експертних знань і автоматизує ухвалення рішень у кіберфізичних системах на основі проведеного аналізу.

Експерименти

Під час експериментального дослідження кіберфізичної системи залізниці дослідник проаналізував захист даних, їхню цілісність і конфіденційність. Це дослідження дозволило визначити ключові вразливості системи та оцінити необхідність впровадження додаткових механізмів захисту.

На першому етапі експерименту було досліджено механізми забезпечення цілісності даних під час їх передачі та зберігання в системах управління залізничною інфраструктурою. Для перевірки цілісності даних використовувалися хеш-функції (SHA-256), для аутентифікації джерела застосовувалися цифрові підписи (RSA), а алгоритми HMAC використовувалися для захисту під час передачі інформації. Було змодельовано тестові сценарії атак, включно зі спробами перехоплення та зміни даних між диспетчерськими системами та вузлами сигналізації. Тестування показало, що навіть незначні зміни в даних під час атаки можуть бути виявлені хеш-функціями, але для забезпечення автентичності джерела були необхідні цифрові підписи.

Наступним кроком стало створення моделі бази даних залізничної інфраструктури, яка включала ключові компоненти, такі як станції, потяги, маршрути, квитки та обладнання. Для кожного компонента були створені таблиці з атрибутами, а їхні взаємодії були змодельовані. Експеримент проводився на платформі MySQL в середовищі Python для аналізу даних із використанням SQL-запитів. Аналіз показав, що уразливості виникли через недостатньо захищені зв'язки між різними об'єктами, особливо між таблицями «Потяги» та «Маршрути», що дозволило потенційним зловмисникам модифікувати критичні дані.

Під час третього етапу експерименту автор зосередився на моделюванні аномалій, які можуть виникнути через порушення цілісності між таблицями «Потяги» та «Маршрути». Було здійснено цілеспрямовану атаку шляхом модифікації даних про маршрути потягів за допомогою SQL-ін'єкцій у змодельованій системі. В результаті система почала призначати потяги на неправильні маршрути, що призвело до порушень розкладу. Це підкреслило необхідність впровадження додаткових механізмів перевірки даних, таких як перевірка парності або двофакторна аутентифікація, для захисту даних про маршрути.

Четвертий етап експерименту полягав у тестуванні безпеки системи продажу квитків. Було досліджено можливі атаки на зміну даних про рейси та квитки, зокрема, зміну доступності рейсів та інформації про ціни. Експеримент проводився на платформі MongoDB з використанням скриптів Python для моделювання змін у базі даних. Імітація показала, що система продажу квитків вразлива до атак типу «людина посередині», що дозволяє зловмисникам змінювати дані квитків. Це продемонструвало необхідність впровадження сильніших методів шифрування даних та багатофакторної аутентифікації для захисту конфіденційної інформації. Крім того, під час моделювання безпеки системи продажу квитків було виявлено, що зловмисники здатні маніпулювати даними рейсів і квитків, що потенційно може призвести до нераціонального використання ресурсів. Експеримент підкреслив важливість впровадження надійних

методів шифрування та систем аутентифікації для захисту таких чутливих даних.

Останній етап експерименту був зосереджений на впровадженні інноваційного підходу до захисту кіберфізичних систем, з акцентом на інтеграції машинного навчання, штучного інтелекту та нейро-символьних методів для виявлення аномалій у реальному часі. Експерименти проводилися на основі моделі залізничної інфраструктури, створеної на платформі TensorFlow для машинного навчання, з інтеграцією бібліотек Python для моделювання атак. Цей етап дослідження відбувався в умовах змодельованих реальних загроз для критичної інфраструктури на спеціалізованій платформі для кіберфізичних систем, що імітує залізничну мережу. Основною метою було тестування алгоритмів нейросимвольного аналізу для виявлення аномальних поведінкових патернів у даних, що передаються через мережеві вузли системи керування потягами. Особливу увагу було приділено впровадженню криптографічних протоколів для забезпечення цілісності та конфіденційності даних. Для цього використовувалися алгоритми шифрування AES-256 та цифрові підписи RSA. Експеримент продемонстрував, що поєднання криптографічних методів з автоматизованими системами реагування на інциденти значно знижує ризики атак на дані пасажирів та вантажу. Крім того, система показала високу точність у виявленні аномалій, що може запобігти потенційним загрозам для безперервності критичних операцій. У таблиці 2 показано широке використання нейросимвольного підходу, який поєднує здатність нейронних мереж виявляти складні патерни та аномалії з можливостями символьних правил для пояснення і перевірки цих аномалій. Це поєднання підвищує захист критичних систем, таких як залізничні мережі.

Таблиця 2.

Приклад використання RNN/LSTM

Вхідні дані	Процес	Результат	Особливості
Дані про потоки між мережевими вузлами та контролерами залізничної інфраструктури, включаючи часи затримки та маршрути передачі даних	Модель аналізує поведінкові патерни в даних за допомогою нейронної мережі, а потім застосовує логічні правила для перевірки відповідності даних очікуваним параметрам залізничної системи	Виявлення порушень, пов'язаних зі змінами в маршрутизації даних або збоїв, які можуть вказувати на потенційні атаки або помилки системи	Нейронна мережа прогнозує аномалії, а неуросимволічна система надає пояснення цих аномалій через встановлені правила безпеки, такі як криптографічні протоколи
Дані з датчиків та контролерів потягів, що містять швидкість, вагу та інші технічні параметри	Аналіз поведінки системи потягів, зокрема затримок сигналів або збоїв даних датчиків, через нейронні мережі, а також застосування символічних правил для перевірки коректності системи	Виявлення невідповідностей в роботі потягів, які можуть вказувати на технічні несправності або спроби несанкціонованого втручання	Алгоритм використовує комбінацію нейронних моделей та символічних правил для підвищення надійності результатів, що забезпечує високу точність виявлення загроз у реальному часі
Журнали мережевого трафіку з мітками часу та різними атрибутами (IP-адреси, порти, типи трафіку тощо)	Модель LSTM навчається на історичних даних для розпізнавання нормальної поведінки. Під час роботи вона аналізує нові дані в реальному часі та визначає, чи відрізняються вони від вивченого нормального поведінки	Виявлення аномалій, які можуть вказувати на потенційні загрози, такі як мережеві атаки або несанкціонований доступ	Модель поєднує нейронні мережі з логічними правилами для пояснення результатів; наприклад, якщо виявлено певні аномалії, їх логічні причини аналізуються

Результати

Проведене дослідження виявило критичні вразливості, пов'язані з цілісністю даних і недостатньою захищеністю з'єднань між компонентами системи. Зокрема, було виявлено, що таблиці «Потяги» та «Маршрути» мають недоліки, якими зловмисники можуть скористатися для внесення несанкціонованих змін (рис. 3).

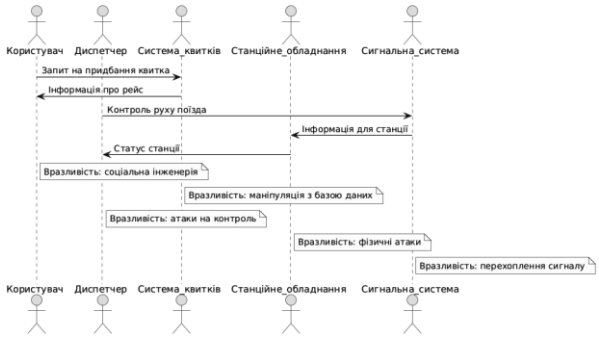


Рис. 3. Квиткова система

Під час тестування безпеки системи продажу квитків було виявлено чотири типи атак на маніпуляції даними квитків, причому в 70% випадків зловмисники змінювали інформацію про доступність рейсів. Час виявлення цих атак становив п'ять секунд, що свідчить про недостатню ефективність системи у виявленні маніпуляцій. Крім того, було проведено десять спроб впровадження SQL, п’ять із яких були успішними. Під час атак система в 60% випадків призначала потяги за неправильними маршрутами, підкреслюючи серйозні недоліки безпеки. На рис. 4 показана діаграма послідовності, яка ілюструє взаємодію між користувачем, веб-інтерфейсом, сервером додатків, базою даних і механізмом маршрутизації під час атаки SQL-ін’єкції.

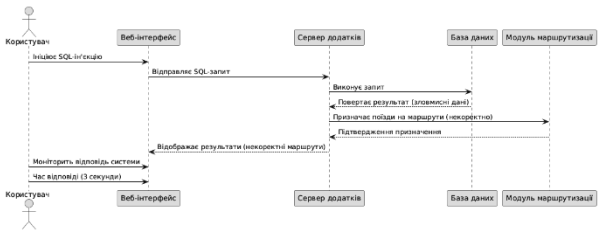


Рис. 4. SQL Injection у кіберфізичних системах

Діаграма детально описує потік подій, починаючи від ініціювання користувачем атаки та завершуючи затримкою відповіді системи. Він виявляє вразливі місця в системі, оскільки сервер додатків виконує зловмисний SQL-запит, що призводить до неправильного призначення поїздів. Трисекундний час відповіді вказує на значну затримку, що відображає неадекватність системи у запобіганні помилкам маршрутизації під час таких атак. Моделювання аномалій за допомогою SQL-ін’єкцій показало, що система може неправильно призначати поїзди за маршрутами, що призводить до збоїв у розкладі. На рис. 5 графік ілюструє кількість спроб ін’єкцій SQL, зроблених в архітектурі безпеки системи продажу квитків протягом десяти випробувань.

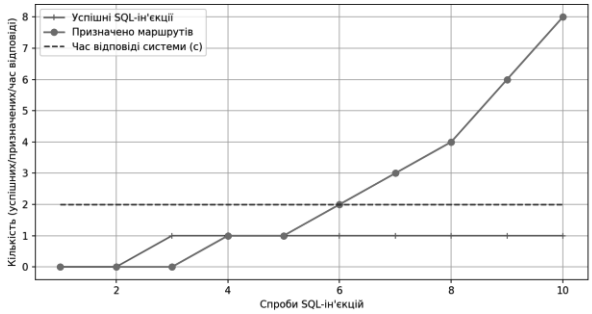


Рис. 5. Результати тесту на SQL-ін’єкції

Згідно з розрахунками автора, половина цих спроб, або 50%, були успішними. Це вказує на серйозні вразливості в системі, які потребують негайного усунення. Серед основних уразливостей, виявлених у схемі, є ненадійні з'єднання між компонентами, недостатня автентифікація користувачів і відсутність шифрування даних.

Поступове збільшення неправильно призначених маршрутів, особливо з п'ятої спроби, демонструє, як внутрішня логіка системи порушується, впливаючи на її здатність правильно обробляти запити під час атаки.

Тож ці результати вказують на те, що, хоча система демонструє частковий опір до атак типу SQL-ін'єкцій, її структура безпеки потребує значного вдосконалення, особливо в таких аспектах, як автентифікація компонентів, шифрування та надійність з'єднань для запобігання більш

серйозним порушенням у майбутніх випробуваннях.

У межах дослідження було здійснено експериментальне впровадження нейросимвольного підходу для виявлення кібератак у системі залізничного транспорту. Використання алгоритмів, які поєднують нейронні мережі та символьну логіку, значно покращило точність і швидкість виявлення загроз порівняно з традиційними методами. Наприклад, модель Long Short-Term Memory (LSTM) досягла точності виявлення 90%, що на 25% вище за результати, отримані з використанням стандартних методів (табл. 3). Моделі на основі нейро-символьного підходу не лише ефективно ідентифікують аномалії в даних, а й дозволяють у режимі реального часу адаптувати стратегії захисту залежно від типу та характеру загроз. Під час тестування було зібрано дані про 1000 спроб кібератак, з яких модель виявила 900 успішних спроб, що демонструє високу чутливість системи до нових загроз.

Таблиця 3.

Оцінка ефективності захисних механізмів від різних видів атак

Тип атаки	Загальна кількість спроб	Успішні спроби	Відсоток успіху	Час (хв)
SQL-ін'єкція	100	50	50%	5
Атака DoS/ DDoS	150	130	86.7%	3
Маніпуляція даними	200	180	90%	4
MiTM-Атака	50	30	60%	6
Невідомі загрози	500	450	90%	2

На основі таблиці 3 видно, що найбільш успішні маніпуляції з даними продемонстрували 90% рівень успіху за час виявлення всього 4 секунди. Це свідчить про ефективність нейросимво-льного підходу у адаптації до нових видів загроз. Крім того, час виявлення атак був значно скорочений, що підвищує загальний рівень безпеки системи.

Результати дослідження підтвердили, що нейросимвольний підхід покращує точність виявлення кібератак і дозволяє автоматично адаптувати стратегії захисту залежно від загрози. Це забезпечує стабільну роботу систем керування поїздами під час атак.

Нейросимвольний підхід значно підвищує ефективність виявлення кібератак, досягаючи до 94% точності під час обробки великих обсягів даних. Крім того, цей підхід адаптує стратегії захисту в реальному часі на основі характеру загрози, забезпечуючи безперебійну роботу систем керування поїздами у разі атак. Зокрема, було продемонстровано, що під час атак, таких як «людина посередині» або DDoS, використання адаптивних моделей захисту зменшує ризик збоїв на 30% порівняно з традиційними методами захисту. Одним з найефективніших інструментів був алгоритм рекурентних нейронних мереж (RNN) з використанням довготривалої пам'яті (LSTM), який був застосований для обробки послідовних даних мережевого трафіку. У дослідженні було використано 10 000 зразків мережевого трафіку, з яких 10% були позначені як аномальні для навчання системи. Модель LSTM досягла 92% точності у прогнозуванні аномалій, аналізуючи часові ряди даних і порівнюючи передбачені значення з реальними.

Тож, результати дослідження показали, що застосування нейросимвольного підходу за допомогою RNN та LSTM може забезпечити високу точність у виявленні кібератак та стабільну роботу кіберфізичних систем.

Обговорення

Отримані результати демонструють, що нейросимвольний підхід значно підвищує ефективність виявлення кібератак у системах залізничного транспорту. Досягнута точність виявлення підтверджує гіпотезу про те, що інтеграція машинного навчання та штучного інтелекту може дати кращі результати порівняно з традиційними методами. Зокрема, здатність адаптуватися до нових типів атак є надзвичайно важливою для забезпечення безпеки критичної інфраструктури, підкреслюючи необхідність впровадження нових технологій у галузі кібербезпеки [8, 17, 18].

Результати показують рівень точності виявлення у 94%, який підтверджує нашу гіпотезу про те, що інтеграція машинного навчання і штучного інтелекту забезпечує вищі результати порівняно з традиційними методами. Здатність системи адаптуватися до нових типів атак є життєво важливою для безпеки критичної інфраструктури [18, с. 309].

Висновки нашого дослідження узгоджуються з попередніми роботами, такими як дослідження Сміта (2022), яке також підкреслює ефективність нейронних мереж у виявленні загроз. Проте, на відміну від цих досліджень, наше дослідження зосереджується на специфіці залізничного транспорту, додаючи нову цінність до розуміння нейросимвольних технологій у цьому контексті.

Попри позитивні результати, існують певні обмеження. По-перше, дослідження базується на моделюванні тестових сценаріїв, які можуть не повністю відображати реальні умови роботи залізничних систем. По-друге, обмежений доступ до даних про реальні атаки може впливати на результати.

Автор статті рекомендує інтеграцію нейросимвольного підходу до вже існуючих систем безпеки залізничного транспорту [19, с. 141]. Крім того, важливо розробити стратегії навчання та збору даних, щоб ці системи могли адаптуватися до нових загроз. Це може включати впровадження механізмів самонавчання, які дозволяють системам не лише виявляти потенційні атаки, а й передбачати їх.

Висновки

Дана робота розглядає актуальну проблему забезпечення кібербезпеки в кіберфізичних системах, зокрема, щодо залізничної інфраструктури. Наукова новизна отриманих результатів полягає в розробці комплексного підходу, який унікально інтегрує машинне навчання, штучний інтелект та нейросимвольні алгоритми для виявлення аномалій в реальному часі. Запропоновані методи ефективно покращують ідентифікацію кіберзагроз, зберігаючи цілісність та конфіденційність даних, що є винятково важливими для захисту інфраструктури, такої як залізничний транспорт.

Практичне значення цих результатів пов’язане із можливістю застосування запропонованих алгоритмів для зміцнення кіберстійкості залізничних систем. Реалізація цих алгоритмів, як очікується, посилить захист чутливих даних пасажирів і вантажів, забезпечуючи безперервність критичних операцій навіть під час кібератак. Крім того, інтеграція криптографічних протоколів та автоматизованої системи реагування на інциденти значно зменшує ризики, пов’язані з витоками даних і порушенням життєво важливих операцій. Цей проактивний підхід до кібербезпеки не лише допомагає в ідентифікації потенційних загроз у реальному часі, а й сприяє швидкому реагуванню на інциденти, тим самим зберігаючи безперервність надання основних послуг.

Перспективи подальших досліджень полягають у вивченні застосування запропонованих методів в інших сферах кіберфізичних систем, таких як енергетика та охорона здоров'я. Крім того, дослідження можуть зосередитися на подальшій розробці нейросимвольних алгоритмів для покращення точності виявлення кібератак у реальному часі та впровадженні нових методів захисту критичної інфраструктури.

Література

1 Taylor, Alex. "Neuro-Symbolic Methods for Cyber Secu-rity." Journal of Artificial Intelligence Research, vol. 12, no. 3, 2021, pp. 500-515.

2 Alashkar H., Ahmad M. A Comprehensive Review on Machine Learning Techniques for Cyber-Physical Sys-tems. Journal of Systems Architecture, 2023, Vol. 129, pp. 102649. DOI: 10.1016/j.sysarc.2022.102649.

3 Mishra A., Gupta R. An Overview of Cyber-Physical Systems: Applications, Challenges, and Future Direc-tions. ACM Computing Surveys, 2022, Vol. 55, № 9, pp. 1–35. DOI: 10.1145/3498707.

4 Vural C., Akbulut U. Cyber-Physical Systems Security: Threats and Machine Learning Countermeasures. IEEE Transactions on Emerging Topics in Computing, 2023, Vol. 11, № 2, pp. 417–426. DOI: 10.1109/TETC.2023.3238515

5 Garfinkel, S., Adams, C., & Warfield, J. (2014). Under-standing cyber-physical attacks and defenses. IEEE Secu-rity & Privacy, 12(1), 20-26.
6 Zhang, Wei, et al. "Adaptive Security Models for Cyber-Physical Systems." In Trends in Cyber-Physical Systems Security, edited by Laura Brown, vol. 5, Cham: Springer, 2022, pp. 200-215.
7 White, Sarah, et al. "Challenges in Protecting Railway Infrastructure." Transport Security Journal, vol. 6, no. 4, 2020, pp. 210-225.
8 Yevdokymov S. O. Modern systems of information pro-tection / Serhiy Oleksandrovich Yevdokymov. - Kyiv: Drukaryk, 2023. - 380 p.
9 Yevdokymov S. O. Applied systems for choosing the optimal route in transport / Serhiy Oleksandrovich Yev-dokimov. - Kyiv: FOP Gu-lyaeva V.M., 2024. - 200 p.
10 Parker, Sam. "The Role of Cryptography in Securing Cyber-Physical Systems." In Cyber Security: Principles and Practices, edited by Alan Richards, New York: Wiley, 2021, pp. 130-150.
11 Khan M. A., Ali S. Machine Learning for Cybersecurity in Cyber-Physical Systems: Recent Advances and Future Directions. Journal of Network and Computer Applica-tions, 2023, Vol. 209, pp. 103531. DOI: 10.1016/j.jnca.2023.103531.
12 Sahu A., Dutta S. Emerging Trends in Cyber-Physical Systems Security: A Review of Machine Learning Solu-tions. Computers & Security, 2022, Vol. 114, pp. 103701. DOI: 10.1016/j.cose.2022.103701.
13 Kumar R., Tripathi A. Anomaly Detection in Cyber-Physical Systems Using Ensemble Learning Techniques. IEEE Transactions on Industrial Informatics, 2023, Vol. 19, № 2, pp. 1253–1263. DOI: 10.1109/TII.2023.3242145.
14 Green, Laura, et al. "Real-Time Threat Detection in Cyber-Physical Systems." Journal of Cyber Security, vol. 15, no. 2, 2021, pp. 200-210.
15 Zhang H., Xu J. Neural-Symbolic Approaches for Cyber-Physical Systems: Enhancing Anomaly Detection. Artifi-cial Intelligence Review, 2023, Vol. 56, № 1, pp. 321–347. DOI: 10.1007/s10462-022-10256-8.
16 Katz, G., et al. "Combining Neural Networks and Sym-bolic Reasoning for Enhanced Security in Cyber-Physical Systems." Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS), 2023, pp. 234-246.
17 Yevdokymov S. O. System programming: Creating appli-cations on Assembler / Serhiy Oleksandrovich Yevdo-kimov. – 2nd ed., supplemented and revised. - London, United Kingdom: LAP LAMBERT Academic Publishing, 2024. - 133 p
18 Johnson, Mark. "Ensuring Data Integrity and Confidenti-ality in Cyber-Physical Systems." International Journal of Cyber Security, vol. 8, no. 3, 2022, pp. 300-315.
19 Rahman M. M., Saha A. Anomaly Detection in Cyber-Physical Systems: A Hybrid Approach. Future Genera-tion Computer Systems, 2022, Vol. 128, pp. 132–146. DOI: 10.1016/j.future.2021.12.035.

Одержано: 08.11.2024
Внутрішня рецензія отримана: 16.11.2024
Зовнішня рецензія отримана: 17.11.2024

Про автора:

Євдокимов Сергій Олександрович,
аспірант кафедри комп'ютерних наук та програмної інженерії,
https://orcid.org/0000-0001-7213-0259

Місце роботи автора:

Херсонський державний університет
email: office@ksu.ks.ua

УДК 004.8 https://doi.org/10.15407/pp2025.01.074

І.П. Сітьков, М.М. Глибовець

МЕТОДИ ОПТИМІЗАЦІЇ АЛГОРИТМІВ РОЗПІЗНАВАННЯ ОБЛИЧЧЯ

У роботі розглянуто основні недоліки сучасних алгоритмів розпізнавання обличчя: низьку швидкість роботи, високу чутливість до якості зображень та розташування обличчя. Запропоновано поділ на три основні підходи до оптимізації алгоритмів розпізнавання обличчя: оптимізація ваги ознак, гіперпараметрів алгоритмів та побудова оптимальної розподіленої системи; наведено приклади застосування алгоритмів Particle Swarm Optimization, Cuckoo Search, методу імітації відпалу, генетичних алгоритмів для усунення згаданих обмежень у наявних алгоритмах. У дослідженні продемонстровано переваги та недоліки вказаних методів оптимізації, визначено перспективні напрями подальших досліджень у галузі оптимізації методів ідентифікації обличчя за допомогою генетичних алгоритмів.
Ключові слова: методи оптимізації, генетичний алгоритм, згорткові нейронні мережі, алгоритми розпізнавання обличчя.

I.P. Sitkov, M.M. Hlybovets

OPTIMIZATION METHODS FOR FACE RECOGNITION ALGORITHMS

The paper examines the main drawbacks of modern face recognition algorithms: low processing speed, high sensitivity to image quality and face positioning. A division into three approaches to face recognition algorithms optimization is proposed: optimization of feature weights, algorithm hyperparameters, and constructing an optimal distributed system architecture. Examples of the application of Particle Swarm Optimization, Cuckoo Search, Simulated Annealing, and genetic algorithms to overcome the mentioned limitations in existing algorithms are provided. The study demonstrates the advantages and disadvantages of these optimization methods and identifies promising directions for further research in face identification methods optimization using genetic algorithms.
Keywords: optimization methods, genetic algorithm, convolutional neural networks, face recognition algorithms.

Вступ

Розпізнавання людини за зображенням обличчя сьогодні є однією з найпоширеніших форм біометричної ідентифікації, яка застосовується як в системах відеонагляду та контролю доступу, так і в роботизованих системах та у взаємодії людини з комп'ютером. Саме тому збільшення попиту на засоби автоматичної детекції породжує потребу в точних та надійних алгоритмах розпізнавання обличчя (далі – РО). Попри досягнення в галузі комп'ютерного зору та широкий спектр застосування методів класифікації зображень, досі існує низка перешкод, які є викликом для сучасних алгоритмів: рівень освітленості обличчя, кут нахилу та повороту, наявність аксесуарів на обличчі, вираз обличчя, вікові зміни, доступність обчислювальних ресурсів на кінцевих пристроях, таких як смартфони або пристрої Інтернету речей тощо. Наслідком чого є суттєве зниження точності та надійності систем.

Зважаючи на перелічені проблеми, метою роботи став аналіз методів оптимізації алгоритмів розпізнавання обличчя, дослідження основних напрямів їхнього застосування та визначення перспектив подальшого розвитку. Окрім цього, результати, наведені у статті, дозволяють зрозуміти переваги та недоліки розглянутих методів оптимізації.

У галузі біометричної ідентифікації на основі обличчя дослідники запропонували низку алгоритмів. Однак оскільки багато з них було розроблено для вирішення досить специфічних проблем комп'ютерного зору та для аналізу використовували набір даних з обмеженого контрольованого

середовища, застосування таких алгоритмів в умовах відкритого світу не є надійним. Наприклад, алгоритм Weber Local Descriptor вирізняється простотою реалізації та ефективністю на тестувальних даних, демонструючи точність розпізнавання вищу за інші поширені алгоритми (Gabor та SIFT) [1]. Утім, він є чутливим до умов освітлення обличчя, що значно звужує сферу його використання. Удосконалений алгоритм WLCGP, представлений у роботі [2], хоч і виправляє попередній недолік, натомість суттєво збільшує обчислювальні витрати, через що також не може широко використовуватись.

Дослідження на тему застосування алгоритмів РО у реальному середовищі [3] також засвідчує обмеження алгоритмів: зниження точності використання у відкритому світі, проблему розпізнавання розфокусованих та розмитих зображень, необхідність великої кількості тренувальних даних та обчислювальних потужностей для навчання моделей. У наведеному в роботі [4] огляді підходів до розпізнавання обличчя та пов'язаних із цим викликів автори фокусують увагу на 7 типах перешкод, викликаних рівнем освітленості, варіативністю пози та виразу обличчя, пластичними операціями, віковими змінами, низькою роздільною здатністю та оклюзіями. Також нетривіальною задачею досі залишається вибір репрезентативних ознак зображення. Відповідно до [4], незважаючи на низку досліджень у галузі комп'ютерного зору, доступні на сьогодні засоби розпізнавання не вирішують жодну з перелічних проблем ефективно через високу різноманітність умов реального світу, які впливають на вигляд обличчя на фото або відео.

Отже, існує необхідність у дослідженні методів оптимізації алгоритмів РО для покращення їхньої стійкості до перешкод та збільшення точності ідентифікації у відкритому середовищі.

На основі проаналізованих методів оптимізації у роботі запропоновано поділ на напрями відповідно до мети застосування та детально розглянуто спосіб використання цих методів для подолання деяких обмежень алгоритмів РО та підвищення точності розпізнавання. Насамкінець вказані шляхи майбутніх досліджень.

Основні напрями оптимізації

Оптимізації алгоритмів РО присвячено велику кількість досліджень. Проаналізувавши деякі з них, нам вдалося виокремити три основні напрями відповідно до мети застосування: оптимізація ваг ознак, гіперпараметрів алгоритму, побудова оптимальної архітектури розподіленої системи.

Оптимізація ваги ознак

Процес розпізнавання обличчя в залежності від алгоритму передбачає низку стадій, серед яких детекція рамок обличчя на зображенні, попередня обробка, нормалізація, узгодження обличчя (зіставлення опорних точок, таких як очі, ніс, рот, з фрагментами зображення), власне класифікація. Однак одним із найважливіших етапів є екстракція ознак обличчя, правильна реалізація якої є передумовою для якісної класифікації. Для цього застосовують як емпіричні методи, які вимагають залучення експерта до формування ознак, так і автоматичну побудову ембедингів за допомогою алгоритмів машинного навчання, методу головних компонент тощо. Проблема полягає в тому, що деякі з ознак є нестійкими до раніше згаданих впливів середовища, або не дозволяють унікально ідентифікувати людину на фото, або не стосуються обличчя, а описують об'єкти на фоні. Як наслідок недосконале числове представлення зображення призводить до значних похибок у передбаченнях алгоритмів. Оптимізація ваги ознак полягає у максимізації впливу репрезентативних ознак та мінімізації значення тих характеристик, які представляють шум або нерелевантну інформацію. Наприклад, відбір може відбуватись шляхом множення значення ознаки на відповідний – більший або менший – коефіцієнт.

Основними інструментами оптимізації ваги ознак в розглянутих роботах є метаевристичні алгоритми Particle Swarm Optimization, метод імітації відпалу, Cuckoo

Search, застосування яких описано в наступних розділах.

Particle Swarm Optimization. Particle Swarm Optimization (PSO) [5] – стохастичний метаевристичний оптимізаційний алгоритм, заснований на популяціях та інспірований колективною поведінкою птахів у зграях. Для пошуку оптимального розв'язку цільової функції ініціалізують популяцію з розв'язків, представлених трьома векторами у D-вимірному просторі, де D – розмірність пошукового простору: x_i – поточна позиція розв'язку, p_i – попередня найкраща позиція та v_i – швидкість зміни позиції по кожній з координат. Еволюція відбувається шляхом оцінки якості поточних розв'язків за допомогою цільової функції та оновлення їхніх позицій відповідно до векторів швидкості. Останні переобчислюються для кожного розв'язку на основі поточної позиції, попереднього вектору швидкості, власної найкращої позиції, глобального найкращого розв'язку та векторів з випадковими значеннями (конкретна реалізація залежить від модифікації алгоритму). Умовою зупинки є досягнення встановленої максимальної кількості ітерацій або отримання розв'язку, який відповідає визначеним критеріям оптимальності.

У роботі [1] Zhang et al. використовують PSO для оптимізації значень компенсувальних коефіцієнтів (КК) $f_1, \ldots, f_8$, що контролюють внесок відповідних 8 ознак у процесі обробки зображення, яке необхідно розпізнати. Ознаки представлені матрицями однакового із зображенням розміру. Їх обчислюють шляхом застосування попіксельних перетворень вхідного зображення (S_1, S_2, S_3, S_4 – матриці лівого, правого, верхнього та нижнього зміщень), комбінування матриць зміщення (S_5 – різницева матриця, S_6 – матриця злиття), перетворення матриці S_6 (S_7 – матриця зі зменшеним шумом), перетворення матриці S_7 (S_8 – матриця підсилення). Утворене в результаті застосування КК зображення Sig (1) розбивають на 36 частин та в кожній з них визначають розподіл значень яскравості пікселів. До об'єднаних розподілів застосовують Principal Component Analysis (PCA) для зменшення розмірності вектора та класифікують утворені ембедінги за допомогою моделі SVM.

$$Sig = A + \sum_{i=0}^{8} f_i * S_i, \qquad (1)$$

де A – вхідне зображення.

Використання алгоритму оптимізації у цьому випадку є обґрунтованим, оскільки виконати повний перебір усіх комбінацій значень (якщо розглядати проміжок [-10, 10] з кроком 0.01) за прийнятний час не можливо. Однак підбір оптимальних значень для КК є критичним, оскільки вони безпосередньо впливають на вагу ознак зображення.

Представлені у роботі [1] експерименти свідчать про ефективність оптимізації. Автори наводять результати тестування власної моделі Image Feature Compenstion (IFC) на трьох датасетах (ORL, YALE, MU_PIE), а також на графіках порівнюють ефективність з іншими алгоритмами. У таблиці 1 наведено узагальнене порівняння на основі даних з дослідження [1].

Таблиця 1

Результати застосування IFC до датасетів

Датасет	*Порівняння з результатами LBP, CLBP, IGP, WLCGP, IGFC*	*Узагальненість КК*
ORL	IFC показує найвищу точність на кількості тренувальних зразків < 7.	Узагальненість висока; вища точність при застосуванні до YALE, на MU_PIE точність не знижується.
YALE	IFC показує загалом вищу точність.	Нижча точність на ORL, висока – на MU_PIE.
MU_PIE	IFC досягає значно вищої точності.	Нижча точність на ORL, висока – на YALE.

Результати дослідження демонструють здатність моделі до узагальнення, вищу точність у порівнянні з відомими алгорит-

мами, а отже, оптимізація за допомогою PSO виявилась ефективною для відбору репрезентативних ознак.

Cuckoo Search Algorithm. Cuckoo Search Algorithm (CSA) [6] – ще один метаевристичний оптимізаційний алгоритм, інспірований природою, а саме паразитичною поведінкою зозулі. Алгоритм спирається на 3 основні правила:

1. Кожна зозуля вікладає одне яйце за раз у випадково обране гніздо.
2. Гнізда з яйцями найкращої якості переходять у наступне покоління.
3. Кількість гнізд фіксована, і власник гнізда може виявити підкинуте яйце з імовірністю $P_a \in [0, 1]$, після чого може або викинути яйце, або залишити гніздо та побудувати нове.

Алгоритм також використовує поняття польоту Леві – випадкового блукання в просторі рішень – для генерування нових розв'язків. Процес оптимізації відбувається ітеративно, де на кожному кроці "зозуля" створює розв'язок та розміщує його у випадковому "гнізді", якщо якість нового розв'язку вища за якість розв'язку, що зберігався там до цього. Тоді частина (P_a) найгірших розв'язків замінюється випадковими новими, і цикл повторюється. Зупинка алгоритму відбувається після досягнення максимальної кількості ітерацій або виконання іншої заданої умови.

У роботі [7] метою застосування CSA також є пошук оптимального набору репрезентативних ознак. На відміну від попереднього дослідження [1], Preeti Malhotra et al. для отримання початкового набору ознак використовують Discrete Cosine Transform (DCT) у поєднанні з PCA. Результати експериментів на датасеті ORL демонструють, що з використанням CSA алгоритм досягає більшої точності (Рис. 1) у порівнянні із застосуванням інших методів оптимізації: PSO, GA, DE (Differential Evolution). Автори також відзначають, що зменшення кількості ознак шляхом оптимізації не призводило до погіршення точності РО.

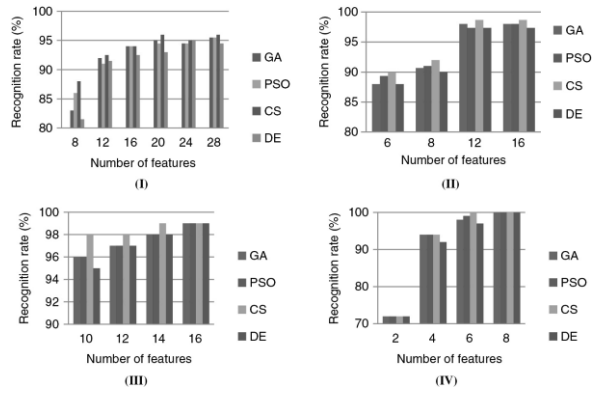


Рис. 1. Порівняння точності розпізнавання між алгоритмами із застосуванням CSA, PSO, GA, DE на датасеті ORL. (I) 40 класів, (II) 30 класів, (III) 20 класів, (IV) 10 класів [7].

Метод імітації відпалу. Метод імітації відпалу – один із найпростіших та найвідоміших метаевристичних оптимізаційних алгоритмів та походить від металургійного процесу термообробки сталі. Оптимізація цільової функції починається з ініціалізації початкової температури та розв'язку випадковими значеннями з простору рішень. На кожному ітерації, поки температура є вищою за встановлений поріг, формують новий розв'язок з околу поточного розв'язку (шляхом невеликого кроку, обміну позицій тощо) та знаходять різницю між значеннями цільової функції на двох розв'язках. Якщо новий розв'язок є кращим (залежно від мети краще мінімізує або максимізує функцію), його приймають як поточний найкращий розв'язок. Інакше його можуть прийняти лише з імовірністю $P = e^{\frac{-\Delta E}{T}}$, де ΔE – різниця значень, T – поточна температура, яка зменшується зі спаданням температури. Наступним кроком зменшують температуру відповідно до встановленої функції (наприклад, $T_{new} = \alpha T$, $\alpha \in (0, 1)$) та повторюють ітерацію.

Дослідження на тему застосування методу імітації відпалу для оптимізації компенсувальних коефіцієнтів $f_1, \ldots, f_8$ продемонстровано на прикладі алгоритму Efficient Face Recognition Algorithm (EFRA) в роботі [8]. Для опису 8 ознак обличчя ($P_1, \ldots, P_8$) автори ввели поняття верхнього, ни-

жнього, лівого, правого, верхнього лівого, правого, нижнього лівого, правого градієнтів, які обчислюються відніманням значень пікселів у сусідніх рядках або стовпцях зображення або по діагоналі. Наступні кроки (розбиття зображення на блоки, підрахунок розподілів яскравості пікселів, зменшення розмірності за допомогою PCA та класифікація з використанням SVM) аналогічні тим, що наведені у розділі про Particle Swarm Optimization.

Обмеження, які намагаються подолати дослідники, – низька точність алгоритмів РО у процесі класифікації зображень, де обличчя частково перекрите іншими об'єктами (масками, окулярами тощо), та низька швидкість класифікації.

Результати експериментів засвідчили вищу точність запропонованого алгоритму на датасетах ORL та YALE у порівнянні з іншими алгоритмами (LBP, CLBP, WLCGP, IGFC). Однак алгоритм поступився у точності LBP на датасеті MU_PIE. За швидкістю розпізнавання алгоритм EFRA виявився гіршим за IGFC, що автори пояснюють витратами на обчислення 8 ознак замість 4, як в алгоритмі IGFC.

Оптимізація архітектури розподіленої системи

У сучасному світі все частіше виникає потреба проводити РО на кінцевих пристроях (edge devices), які мають досить обмежену потужність та обчислювальну здатність (смартфони, камери відеонагляду, системи контролю якості продукції, окуляри віртуальної реальності, розумні біометричні замки, дрони, роботи тощо). Оптимізації алгоритмів для ефективної роботи в середовищі з обмеженими ресурсами присвячено роботи [9], [10]. Іншим підходом до покращення якості ідентифікації на кінцевих пристроях є побудова оптимальної розподіленої системи, яка ефективно використовує можливості всіх учасників взаємодії. Важливу роль у системах сьогодні відіграють хмарні платформи, які дозволяють гнучко використовувати обчислювальні потужності у великих обсягах.

У дослідженні [11] автори пропонують оптимальну архітектуру системи для розпізнавання обличчя у реальному часі, де взаємодіють хмарні сервери та кінцеві пристрої. Основною метою є збільшення швидкості РО. Для цього процес розпізнавання розділяють на два етапи:

1) детекція обличчя на зображенні;
2) власне розпізнавання обличчя.

Перший етап передбачає визначення обмежувальної рамки навколо обличчя на фреймах відео та розмітку обличчя (детекцію очей, носа, рота) за допомогою згорткової нейронної мережі MTCNN з подальшим обтинанням зображення. Цей етап відбувається безпосередньо на кінцевих пристроях. У випадку успішної детекції оброблені фрейми надсилають до сервера. За рахунок зменшення розміру зображень процес обміну даними та розпізнавання значно прискорюється.

Другий етап проходить на строні сервера та включає розпізнавання обличчя за допомогою алгоритма Dlib. Кожне зображення представляють як 128-вимірний ембединг, для якого за Евклідовою відстанню знаходять найкращий відповідник серед зображень у базі даних.

Оцінку ефективності впровадженої системи автори проводили на основі порівняння кількості фреймів, які система могла обробити за секунду. Для експериментів використовувались відео різної довжини (50, 40, 10 секунд), роздільної здатності (HD, FHD) та FPS (30, 60, 90). Результати продемонстрували збільшення швидкості обробки відео у 8.5 разів у порівнянні з системою, яка складалася тільки з кінцевого пристрою, та в 1.91 разів у порівнянні з системою, яка включала тільки хмарні сервери.

Отримані переваги у швидкості свідчать про перспективність напряму з розробки колаборативних систем.

Оптимізація гіперпараметрів алгоритмів

Гіперпараметри визначають ефективність алгоритму. Однак широкий домен кожного з них та кількість шуканих змін-

них унеможливлюють повний перебір варіантів для отримання оптимальних значень під час вирішення конкретної задачі, що викликано високими обчислювальними витратами. У зв'язку з цим розробники встановлюють парметри емпіричним шляхом, спираючись на попередні дослідження та закономірності, що також часто не є ефективним. Для вирішення вказаної проблеми до значень часто застосовують оптимізаційні алгоритми.

Поруч з уже розглянутими алгоритмами розпізнавання обличчя для класифікації зображень використовують згорткові нейронні мережі (ЗНН), які сьогодні є класичним інструментом комп'ютерного зору. Вони демонструють високу точність та узагальненість за умови правильної архітектури, тому оптимізації їхніх гіперпараметрів приділяють значну увагу. Зокрема, для цього використовують генетичні алгоритми.

Генетичні алгоритми. Генетичні алгоритми – сімейство метаевристичниих оптимізаційних алгоритмів, які симулюють генетичні процеси над популяціями розв'язків. Алгоритм починає роботу з ініціалізації популяції хромосом – розв'язків, кожен ген відповідає за шукану змінну або її частину (у випадку бінарного кодування). Після цього на кожній ітерації виконується оцінка пристосованості хромосом за допомогою функції здоров'я. Далі відповідно до методу селекції хромосоми відбирають до батьківського пулу. Популяція для наступного покоління формується шляхом застосування кросинговеру (обміну хромосом ділянками генів між собою) та мутації (випадкової зміни генів) до хромосом у батьківському пулі. Еволюційний процес продовжуються, доки не буде досягнуто умови зупинки або кількість ітерацій не перевищить максимальну.

У роботі [12] автори досліджують ефективність застосування генетичного алгоритму для оптимізації гіперпараметрів ЗНН на прикладі запропонованої моделі CNN-GA. Вони прагнуть збільшити її точність у розпізнаванні облич з різним кутом повороту. Namrata Karlupia et al. для оптимізації обрали основні гіперпараметри – кількість та розмір згорткових фільтрів для 2, 3, та 4-шарових ЗНН; функція здоров'я представлена як *100 - accuracy%*.

За результатами експериметів найвищу точність (94.5%) було досягнуто в процесі використання 4-шарової ЗНН з оптимізованими параметрами. Автори також демонструють перевагу моделі CNN-GA в точності, наводячи порівняння з іншими алгоритмами: AlexNet (78%), VGG-16 (85.002%), ResNet50 (86.066%), InceptionV3 (87.11%) та власною моделлю з емпірично підібраними значеннями параметрів (60%).

Дослідження Sehla Loussaief et al. [13] також було спрямоване на вивчення впливу оптимізації гіперпараметрів ЗНН за допомогою генетичного алгоритму на результати класифікації. Модель, отримана ручним підбором значень, виявилася неточною (30%), тоді як запропонована 5-шарова ЗНН з оптимізованими параметрами досягла показника 98.94%.

Окрім вищої точності оптимізованої моделі, автори Yanan Sun et al. за підсумками роботи [14] відзначають зменшення кількості тренованих параметрів моделі, що тягне за собою зниження обчислювальних витрат на навчання алгоритму та збільшення його швидкодії.

Висновки

У роботі розглянуто основні обмеження сучасних методів розпізнавання обличчя, до яких належить низька швидкість обробки, чутливість до якості зображення та розміщення обличчя. За підсумками аналізу виокремлено 3 напрями застосування оптимізації: оптимізація ваг ознак зображення, гіперпараметрів алгоритму, побудова оптимальної архітектури розподіленої системи. Також наведено метаевристичні оптимізаційні алгоритми PSO, CSA, метод імітації відпалу, генетичні алгоритми, які дозволяють ефективно долати деякі зі згаданих обмежень, зокрема, низьку точність розпізнавання облич під різним кутом повороту або частково перекритих іншими об'єктами. Вказані алгоритми вирішують проблеми відбору репрезентативних ознак та пошуку оптимальних значень для гіперпараметрів. Для ілюстрації продуктивності оптимізованих алгоритмів РО ми навели

результати досліджень, де їх було успішно імплементовано та протестовано. Окрему увагу у роботі приділено важливості організації оптимальної архітектури систем для розпізнавання обличчя, оскільки від цього залежить їхня швидкодія.

Перспективним напрямом сьогодні є дослідження алгоритмів машинного навчання, тому подальші зусилля у сфері застосування методів оптимізації можуть бути спрямовані на вивчення ефективності використання генетичного алгоритму для оптимізації гіперпараметрів згорткових нейронних мереж, які визначають швидкість навчання моделі, кількість епох, зсув згорткового фільтра та dropout rate. Результати стануть корисними для розуміння практичності оптимізації вказаних гіперпараметрів та доцільності її використання в подальших розробках.

Література

1. Zhang, Y. and Yan, L. (2023) 'Face recognition algorithm based on particle swarm optimization and image feature compensation,' *SoftwareX*, 22, p. 101305. https://doi.org/10.1016/j.softx.2023.101305.
2. Fang, S. *et al*. (2017) 'Face recognition using weber local circle gradient pattern method,' *Multimedia Tools and Applications*, 77(2), pp. 2807–2822. https://doi.org/10.1007/s11042-017-4412-8.
3. Zafeiriou, S., Zhang, C. and Zhang, Z. (2015) 'A survey on face detection in the wild: Past, present and future,' *Computer Vision and Image Understanding*, 138, pp. 1–24. https://doi.org/10.1016/j.cviu.2015.03.015.
4. Oloyede, M.O., Hancke, G.P. and Myburgh, H.C. (2020) 'A review on face recognition systems: recent approaches and challenges,' *Multimedia Tools and Applications*, 79(37–38), pp. 27891–27922. https://doi.org/10.1007/s11042-020-09261-2.
5. Poli, R., Kennedy, J. and Blackwell, T. (2007) 'Particle swarm optimization,' *Swarm Intelligence*, 1(1), pp. 33–57. https://doi.org/10.1007/s11721-007-0002-0.
6. Gandomi, A.H., Yang, X.-S. and Alavi, A.H. (2012) 'Erratum to: Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems,' *Engineering With Computers*, 29(2), p. 245. https://doi.org/10.1007/s00366-012-0308-4.
7. Malhotra, P. and Kumar, D. (2017) 'An optimized face recognition system using Cuckoo search,' *Journal of Intelligent Systems*, 28(2), pp. 321–332. https://doi.org/10.1515/jisys-2017-0127.
8. Yan, L., Zhang, Yanhu and Zhang, Yanjun (2023) 'A fast face recognition system based on annealing algorithm to optimize operator parameters,' *The Imaging Science Journal*, 71(4), pp. 323–330. https://doi.org/10.1080/13682199.2023.2182261.
9. Huang, J., Shang, Y. and Chen, H. (2019) 'Improved Viola-Jones face detection algorithm based on HoloLens,' *EURASIP Journal on Image and Video Processing*, 2019(1). https://doi.org/10.1186/s13640-019-0435-6.
10. Li, W. *et al*. (2023) 'Face recognition model optimization research based on embedded platform,' *IEEE Access*, 11, pp. 58634–58643. https://doi.org/10.1109/access.2023.3277495.
11. Oroceo, P.P. *et al*. (2022) 'Optimizing Face Recognition Inference with a Collaborative Edge–Cloud Network,' *Sensors*, 22(21), p. 8371. https://doi.org/10.3390/s22218371.
12. Karlupia, N. *et al*. (2023) 'A genetic algorithm based optimized convolutional neural network for face recognition,' *International Journal of Applied Mathematics and Computer Science*, 33(1). https://doi.org/10.34768/amcs-2023-0002.
13. Loussaief, S. and Abdelkrim, A. (2018) 'Convolutional Neural Network Hyper-Parameters Optimization based on Genetic Algorithms,' *International Journal of Advanced Computer Science and Applications*, 9(10). https://doi.org/10.14569/ijacsa.2018.091031.
14. Sun, Y. *et al*. (2020) 'Automatically designing CNN architectures using the genetic algorithm for image classification,' *IEEE Transactions on Cybernetics*, 50(9), pp. 3840–3854. https://doi.org/10.1109/tcyb.2020.2983860.

Одержано: 18.07.2024

Внутрішня рецензія отримана: 24.07.2024

Зовнішня рецензія отримана: 25.07.2024

Про авторів:

Сітьков Ілля Павлович,
магістр 2 року навчання
спеціальності “Комп’ютерні науки”
https://orcid.org/0009-0004-6311-1442

*Глибовець Микола Миколайович***,**
доктор фізико-математичних
наук, професор.
https://orcid.org/0009-0005-6942-8026

Місце роботи авторів:

Національний університет
«Києво-Могилянська академія»,
04070, м. Київ, вул. Сковороди, 2.
Тел.: (044) 463-69-85
Email: i.sitkov@ukma.edu.ua
Email: glib@ukma.edu.ua

УДК 004.8, 005.94 https://doi.org/10.15407/pp2025.01.082

І.Ю. Гришанова, Ю.В. Рогушина

МЕТОД ПОБУДОВИ ТА ОПТИМІЗАЦІЇ МАРШРУТУ ДЛЯ КОМПОЗИТНОГО ВЕБСЕРВІСУ НА ОСНОВІ Q-LEARNING

Запропоновано метод автоматизованої генерації маршруту для композиції вебсервісів відповідно до поставленої цілі з використанням методів машинного навчання з підкріпленням (reinforcement learning). Агент, використовуючи Q-learning, поступово накопичує знання про середовище та оновлює оцінку корисності своїх дій, яким відповідають наявні сервіси.

Задача поділяється на дві підзадачі: побудову можливих маршрутів – таких послідовностей сервісів, що результати виконання попереднього сервісу змінюють середовище, уможливлюючи виконання наступного; та вибір оптимального маршруту відповідно до критеріїв QoS, що адаптується до змін самого середовища.

Визначено основні складові навчання з підкріпленням, розглянуто їхню специфіку для аналізу сервісів. Розглянуто додаткові підходи, які дозволяють уникнути зациклення та використання непотрібних сервісів. Запропоновано модифікацію методу Q-learning для автоматичної генерації маршрутів на основі вхідних і вихідних даних вебсервісів і для вибору оптимального маршруту на основі аналізу їхніх якісних характеристик з використанням підходу з пам'яттю, де агент із кожним кроком розширює свої знання про середовище. Запропоновано програмну реалізацію розробленого методу, яка дозволяє оцінити його властивості.

Розглянуто можливості запропонованого методу на прикладі генерації оптимальних послідовностей вивчення матеріалів для індивідуальних освітніх траєкторій відповідно до особистих потреб студентів. Кожен навчальний об'єкт (інформаційний об'єкт, що використовується для освітніх потреб та описаний метаданими) розглядається як окремий сервіс, де входи та виходи представлені необхідними та результуючими компетенціями.

Ключові слова: вебсервіс, композиція сервісів, маршрут, машинне навчання.

I. Grishanova, J. Rogushina

FLOW CONSTRUCTING AND OPTIMIZING METHOD FOR COMPOSITE WEB SERVICE BASED ON Q-LEARNING

We propose a method of automated flow generation for the web services composition according to the defined target state based on reinforcement machine learning. An agent that uses Q-learning gradually accumulates knowledge about the environment to updates the evaluations of the usefulness of its actions (these actions correspond to the existing services).

The task is divided into two subtasks: - construction of possible flows represented as sequences of services where the results of the previous service execution change the current environment state and enable the execution of the next service; - choice of the optimal flow according to the history of interactions and to QoS criteria that is adapted to environment changes.

We determine the main components of reinforcement learning and analyze their specifics for service composition task. Additional approaches that allow avoiding looping and the use of unnecessary services are considered. We propose modification of the Q-learning method developed for automatic generation of flows based on input and output data of web services and for selecting the optimal flow based on the analysis of their qualitative characteristics. This modified method uses approach with memory where the agent expands its knowledge about the environment at each step. We consider characteristics of proposed method based on analysis of its software implementation.

Possibilities of proposed method are considered on example of generation an optimal study sequences used for individual educational trajectories in accordance with the personal needs of students. Every learning object (information object used for educational needs described by metadata) is considered as a specific service where inputs and outputs are represented by required and result competencies.

Keywords: web service, composition of services, flow, machine learning.

1. Вступ

У сучасних системах вебсервіси є ключовим компонентом для реалізації інтеграційних рішень. Вебсервіси забезпечують стандартизований інтерфейс для взаємодії між системами, а їх об'єднання дозволяє створювати складні композитні сервіси, що використовують кілька окремих вебсервісів для досягнення певної функціональності [1, 2].

Композиція вебсервісів – це процес вибору набору сервісів та послідовності їх виконання, що дозволяє перейти з наявного стану у бажаний (цільовий) стан. Водночас враховуються як функціональні (входи, виходи, як саме перетворюються дані), так і нефункціональні (QoS – швидкість роботи, вартість, доступність тощо) властивості сервісів [3]. Етапами створення композиції вебсервісів є генерація можливих маршрутів сервісів, що викликаються для досягнення цільового стану, та вибір серед них оптимального (ми використовуємо термін "*маршрут*" (flow) – за аналогією з теорією графів, де так позначається послідовність вершин та орієнтованих ребер – для опису послідовності виконання сервісів). Такий маршрут є основою для подальшої композиції сервісів.

Одним із сучасних підходів до вирішення задачі автоматизованої композиції вебсервісів є використання методів машинного навчання (Machine Learning, ML), які дозволяють ефективно адаптувати алгоритм до змін у середовищі, оптимізувати вибір сервісів на основі параметрів QoS та враховувати складні залежності між даними та процесами.

ML – це галузь штучного інтелекту, що досліджує алгоритми й статистичні моделі, які дозволяють комп'ютерам автоматично поліпшуватися під час виконання завдань, спираючись на накопичені дані та досвід, без явного опису алгоритму [4]. Тобто комп'ютерні програми можуть покращувати свою продуктивність, аналізуючи дані й досвід, а не виконуючи заздалегідь прописані інструкції. Сучасні алгоритми ML оптимізують продуктивність з точки зору певного критерію на основі накопиченого досвіду [5]. *Навчання з підкріпленням (reinforcement learning, RL)* – підклас ML, що використовує зворотний зв'язок у вигляді винагороди або штрафу залежно від корисності певної дії для досягнення обраної цілі [6].

В рамках базової RL-системи виділяють наступні складові:

Агент – це програма (наприклад, Q-learning), що шукає оптимальний шлях для досягнення цільового стану, тобто оцінює, які дії слід виконати у певному стані, щоб максимізувати винагороду і наблизитись до цільового стану. Агент навчається, удосконалюючи свої стратегії для досягнення максимальної винагороди в довгостроковій перспективі.

Середовище – оточення агента, з яким він взаємодіє. Воно змінює стан у відповідь на дії агента й генерує винагороду, яка стимулює або карає агента залежно від того, наскільки його дії сприяють досягненню мети.

Стратегія – правила, за якими агент обирає дії в кожному стані. Ці правила агент змінює в результаті навчання, щоб знайти дії для максимізації сумарної винагороди.

Функція винагороди – правило, яке визначає, яку винагороду отримує агент за дію в конкретному стані та забезпечує зворотний зв'язок, вказуючи агенту, чи його дії наближають до мети, і впливає на оновлення стратегії.

Функція цінності – оцінка впливу стану або пари "стан-дія" на винагороду, яка дозволяє агенту оцінити вигідність своїх дій не лише на короткостроковій, а й на довгостроковій основі.

Правила переходу між станами – визначення того, як середовище змінює свій стан після виконання агентом певної дії. Ці правила можуть бути як детермінованими, так і стохастичними.

Спостереження – інформація про стан середовища, доступна агенту залежно від типу середовища (повна або часткова).

Ці компоненти описують замкнений цикл взаємодії в RL-системі (агент

обирає дію → середовище реагує, змінюючи стан і надаючи винагороду → агент оновлює свою стратегію на основі отриманого досвіду), який повторюється, поки агент не навчиться оптимально виконувати завдання або досягати мети в середовищі.

У методі навчання з підкріпленням Q-learning агент будує таблицю Q-значень для різних станів і дій, поступово знаходячи оптимальну стратегію через спроби й помилки. Таблиця Q-значень (або Q-таблиця) — це матриця, яка використовується в алгоритмі Q-learning для зберігання оцінок корисності виконання певної дії в конкретному стані середовища: рядки відповідають станам, стовпці – можливим діям. Значення Q(s,a) в таблиці показує очікувану винагороду, яку агент отримає, почавши зі стану s, виконавши дію a, і дотримуючись оптимальної політики в подальшому. Завдяки тому, що агент постійно оновлює свої знання на основі отриманих винагород, він здатний адаптуватися до змін у середовищі, що робить його ефективним інструментом для задач, де середовище є динамічним або частково відомим.

2. Постановка задачі

У більшості відомих рішень з композиції вебсервісів побудова маршруту не автоматизована. Ми пропонуємо:

- алгоритм автоматичної побудови маршруту за описами характеристик вебсервісів та вимогами до композитного сервісу;

- метод вибору оптимального маршруту на основі порівняння властивостей за визначеним набором критеріїв QoS .

Для вирішення задачі композиції вебсервісів методом Q-learning ми пропонуємо змоделювати та описати цю задачу через поняття агента і навколишнього середовища, де агент вивчає, які сервіси обирати для досягнення цільового результату (отримати вихідні параметри композитного сервісу).

Побудова композитного сервісу з оптимальними параметрами QoS має враховувати додаткові вимоги:

- уникнення використання зайвих і непотрібних сервісів;

- уникнення зациклювання;

- спрощення введення початкових даних, щоб термінальні стани, які не є цільовими, обчислювалися автоматично. Досягнення таких термінальних станів означає помилку і призводить до закінчення навчання;

- адаптація до можливих змін характеристик сервісів та критеріїв QoS;

- масштабування методу для великої кількості сервісів;

- оптимізація знайденого маршруту.

3. Елементи композитного вебсервісу

Ми пропонуємо наступне визначення вебсервісу: *вебсервіс WS* – це п’ятірка

$$WS = <ID, IN, OUT, QoS, Descr>,$$

де:

ID – ідентифікатор вебсервісу, унікальний атрибут: він дозволяє однозначно ідентифікувати кожен вебсервіс у системі;

IN – множина вхідних параметрів $<inp_1; inp_2; \ldots; inp_n>$, $i = \overline{1,N}$ яка описує передумови для виконання веб-сервісу: чітке визначення вхідних даних дозволяє зрозуміти, які саме параметри має отримати вебсервіс для свого виконання;

OUT – множина вихідних параметрів $<out_1; out_2; \ldots; out_m>$, $m = \overline{1,M}$, яка описує результат роботи вебсервісу та використовується для створення композитного сервісу (виходи одного вебсервісу можуть використовуватися як входи для іншого);

QoS – кортеж нефункціональних характеристик <att_1; att_2; …; att_k>, $k = \overline{1,K}$ (наприклад, доступність, час виконання, пропускна здатність), що містить важливі елементи для задачі адаптивної композиції, які визначають, наскільки вебсервіс відповідає вимогам користувача та середовища, і впливають на вибір сервісів у динамічному середовищі;

Descr – текстовий опис вебсервісу: цей компонент корисний для розуміння призначення вебсервісу, а також при семантичній обробці вебсервісу для певних цілей. На даний час цей аспект ми не розглядаємо, однак вважаємо перспективним дослідження семантичної обробки таких описів.

Композитний вебсервіс (Composite Web Service) – це впорядкована сукупність вебсервісів, об'єднаних для досягнення спільної мети – цільового стану. На початку роботи композитний сервіс задається вхідними і вихідними параметрами (початковим і цільовим станами).

В результаті роботи нам необхідно отримати композитний вебсервіс як множину можливих маршрутів і оптимальний маршрут – послідовність сервісів з оптимальними параметрами *QoS*.

Визначимо специфіку композиції сервісів у термінах базових елементів ML. *Середовище* – це простір станів і дій, які агент може виконувати для досягнення цільового стану. *Станами* є можливі комбінації доступних вхідних і вихідних параметрів, що змінюються залежно від того, які сервіси виконано. Множина всіх можливих станів представляє *простір станів*, де окремо виділяють початковий і цільовий стани. *Поточний стан агента* – це те, які вхідні та вихідні дані відомі агенту після виклику певного вебсервісу. *Початковий стан* визначається вхідними параметрами композитного сервісу. *Цільовий стан* визначається набором вихідних даних (виходів композитного вебсервісу), які агент повинен досягти. Якщо агент досягає цільового стану, епізод завершується, і він отримує винагороду. Досягнення цільового стану вказує на успішну композицію сервісів.

Дія – це операції над станами в результаті виклику агентом доступного вебсервісу, що змінює поточний стан агента, та збільшує його знання про середовище.

Маршрут – це послідовність дій агента (викликів вебсервісів), які переводять його з початкового стану до цільового стану. Кожен крок маршруту відповідає виконанню певної дії (вебсервісу), яка змінює поточний стан агента на новий.

В результаті роботи алгоритму агент формує множину маршрутів – набір усіх припустимих послідовностей дій, які переводять агента з початкового до цільового стану. Кожному маршруту відповідає послідовність вебсервісів, що можуть бути використані для досягнення мети.

4. Q-learning для побудови оптимального композитного сервісу

В загальному вигляді програма навчання агента методом Q-learning має наступний вигляд [7]:

initialize Q(s, a) for all $s \in S$, $a \in A(s)$

for each episode do

s ← s0 // Початковий стан

while s ∉ terminal state do

Choose $a \in A(s)$

using ε – greedy policy

Execute action a, observe reward r and new state s'

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max a`Q(s`,a`) - Q)s,a)]$$

s ← s' // Перехід до нового стану

end while

end for

де:

s – поточний стан агента,

a – обрана дія,

r – винагорода за виконання дії a,

s′ – новий стан після виконання дії,

α – швидкість навчання (learning rate),

γ – фактор дисконтування (discount factor),

ε – параметр для ε-жадібної стратегії, що контролює баланс між дослідженням та експлуатацією.

Загальні характеристики методу Q-learning створюють основу для розробки алгоритму, що дозволяє агенту діяти в середовищі оптимально (саме цей алгоритм реалізовано у нашій програмі). Він забезпечує ефективне навчання, знаходження оптимальної стратегії і, зрештою, вирішення поставленої задачі – побудови оптимального композитного сервісу з урахуванням заданих параметрів QoS.

Розглянемо, як саме ми пропонуємо модифікувати Q-learning для побудови композитного сервісу.

У просторі станів S, що описує можливі комбінації вхідних даних, можна виділити наступні підмножини:

- $S_{ter\,min\,al} \subseteq S$ – множина термінальних станів, де $\forall s_{ter\,min\,al} \in S_{ter\,min\,al}$ – термінальний стан, у якому агент більше не може виконати жодну операцію або досяг завершення своєї мети.

- $\forall S \exists s_{target} \in S_{ter\,min\,al}$ – цільовий стан, в якому досягнуто бажаного результату (отримано необхідні вихідні дані).

- $S_{negative} \subseteq S$ – множина негативних станів, де $\forall s_{negativel} \in S_{negativel}$ – це стан, у якому агент не може знайти жодної дії, яка б наближала його до цільового стану, тобто не може рухатися до мети, і тому епізод має завершитися зі штрафом. .

Стан вважається негативним в наступних ситуаціях:

- для даного стану немає можливих дій (тобто жоден сервіс не може бути виконаний),

- немає доступних дій (сервісів), що переводять агента у новий корисний стан (тобто будь-які дії не наближають агента до цільового стану).

Якщо доступна дія не веде до цільового стану або не додає корисних виходів, можна розглядати її як помилкову і призначати штраф за її виконання.

Функція переходу — механізм, який визначає, як змінюється стан після виконання дії. У запропонованому методі зміна стану залежить від того, які виходи вебсервісу додаються до поточного стану (next_state), та від того, які обмеження враховуються для доступності сервісів. Ми пропонуємо функцію переходу визначати за допомогою таблиці переходів *transition_table*, яка зберігає нові стани для кожної пари (стан, дія). Такий підхід враховує лише доступні сервіси в поточному стані, що запобігає вибору некоректних дій. З кожним кроком агент, переходячи зі стану $s \in S$ в стан $s` \in S$, розширює свої знання про середовище.

Винагорода (Reward) в алгоритмі Q-learning – це кількісна оцінка того, наскільки корисною або шкідливою була ця дія для досягнення кінцевої мети, тобто отримання цільових вихідних даних у композитному сервісі. Агент отримує її після виконання певної дії (виклик/аналіз сервісу) у конкретному стані. Винагорода визначається за формулою:

$$R(s,a) = \sum_{i=1}^{n} w_i * r_i(s,a) - 1.$$

Стан $s \in S$, дія $a \in A$, w_i – вага i-го параметру QoS значення винагороди за стан s та дію a, $r_i(s,a)$ – нормалізована оцінка винагороди для i-го параметру QoS, що визначається за наступним правилом:

$$r_i(s,a) = \frac{att_i(s,a) - att_{min}}{att_{max} - att_{min}},$$ якщо параметр і потрібно максимізувати,

$$r_i(s,a) = 1 - \frac{att_i(s,a) - att_{min}}{att_{max} - att_{min}},$$ якщо параметр і потрібно мінімізувати.

$att_i(s,a)$ – значення i-го параметра QoS для дії a у стані s, а att_{min} та att_{max} – його мінімальне та максимальне значення серед усіх можливих сервісів.

Ми пропонуємо доповнювати визначення винагороди штрафами за непотрібні дії, що не наближають агента до цілі або не дають нових корисних виходів. Додається *функція втрат (Loss Function),* подібно до нейронних мереж, яка визначається через різницю між поточним Q-значенням і цільовим значенням, що використовується для оновлення Q-таблиці. Щокожного оновлення вона визначається як окрема метрика TD-помилки – Temporal Difference error .

Запропонований метод використовує популярну у Q-learning *ε-жадібну (epsilon-greedy) стратегію,* що забезпечує баланс між дослідженням нових дій і використанням вже відомих дій з високими значеннями Q-значень. Спочатку агент активно досліджує середовище, а після накопичення достатньої інформації поступово починає фокусуватися на оптимальних діях, зменшуючи випадкові вибори. Проведені нами експерименти на великих наборах даних виявили потребу в оптимізації процесу обчислення стратегії. Це зумовило необхідність розробки автоматизованого вибору методу розрахунку стратегії, що було реалізовано в нашій роботі.

Алгоритм навчання для побудови оптимального композитного вебсервіса має наступний вигляд:

1. *Ініціалізація середовища*: описуються входи, виходи та QoS-параметри вебсервісів; формується список цільових виходів та початковий стан композитного сервісу; задаються глобальні параметри важливості елементів QoS; генерується таблиця досяжних станів, які можна отримати, виконуючи сервіси послідовно; формується таблиця переходів, можливих за умови виконання наявних сервісів.

2. *Пошук можливих маршрутів* за допомогою модифікованого Q-learning:

- *Ініціалізація Q-таблиці* всіх можливих станів і дій (значення Q-функції ініціалізуються нулями);

- *Навчання.* Початковий стан задається відповідно до вхідних даних композитного сервісу. Агент вибирає дію (вебсервіс), яку можна виконати у цьому стані на основі обраної стратегії. Агент виконує дії та в результаті цього переходить з одного стану в інший. Винагорода нараховується за: *прогрес у досягненні мети* та *якість виконання сервісу* (значення QoS). Якщо досягнуто стан, коли жодна дія не наближає агента до цілі, то за виконання будь-якої можливої дії агент отримує негативну винагороду. Якщо через певну кількість перевірок ситуація не змінюється, епізод завершується невдачею.

3. *Вибір і контроль стратегії* (ε-жадібної стратегії дій агенту). На основі проведених експериментів на даний час ми використовуємо експоненційне та лінійне зниження ε. Найкращий метод зниження ε підбирається автоматично шляхом запуску тестових епізодів з оцінкою кількості знайдених маршрутів та часу виконання.

4. *Збереження досвіду.* Після завершення навчання Q-таблиця зберігається у файлі. Досвід агента (знайдені маршрути) також записується у форматі JSON.

5. *Оцінка ефективності*. Для контролю роботи алгоритму процес знаходження оптимального маршруту відображається наступним чином: виводяться всі маршрути, які досягають мети, із зазначенням їхніх сумарних QoS і наприкінці виводиться оптимальний. За результатами роботи будуються графіки сумарної винагороди, TD-помилки та зміни **ε** за епізодами.

6. *Використання збереженого досвіду*. Отримані результати роботи агента (Q-таблиця та список маршрутів) можливо зберігати і в подальшому **з**авантажувати для виконання цільового завдання без необхідності додаткового навчання.

5. Наукова новизна

Агент, який використовує Q-learning, може поступово вивчати та вдосконалювати стратегію вибору оптимальних вебсервісів, оновлюючи свою Q-таблицю – структуру, в якій зберігаються оцінки вигідності переходу з одного стану в інший через виконання конкретної дії (виклику вебсервісу). Під час цього процесу агент отримує винагороду за правильний вибір дії або штраф за неправильний і таким чином поступово адаптується до змін у середовищі.

Більшість досліджень [8-10], що стосуються композиції вебсервісів із використанням методів навчання з підкріпленням (RL), зосереджуються на оптимізації параметрів якості обслуговування (QoS), тоді, як проблему автоматичної побудови можливих маршрутів часто ігнорують або вирішують вручну чи напів-автоматично. В попередній роботі автори дослідили доцільність використання Q-learning для композиції сервісів за попередньо заданим

маршрутом та визначили потребу в автоматизації побудови таких маршрутів [11].

У роботі запропоновано модифікований алгоритм Q-learning для автоматичної генерації маршрутів на основі вхідних і вихідних даних вебсервісів і вибору оптимального згідно з параметрами QoS і глобальних ваг важливості цих параметрів.

Запропонований підхід побудови множини можливих маршрутів для заданого композитного сервісу і вибір оптимального відрізняється від традиційних тим, що адаптується до змін у середовищі в реальному часі. Агент навчається на основі зворотного зв'язку від середовища і самостійно коригує вибір вебсервісів для досягнення кращої якості. Цей метод не потребує попередньо повної інформації про весь простір станів вебсервісів. Невідомі стани обробляються окремо: якщо для певного стану немає можливих переходів, епізод завершується зі штрафом, що мінімізує ризик неправильних дій та переходів у невизначені стани.

Надана формалізація композиції вебсервісів через базові поняття методу Q-learning: стани, простір станів, дії, переходи таке інше, що дозволяє реалізувати автоматизацію процесу побудови маршрутів методом Q-learning, який раніше виконувався частково вручну.

Запропонований метод базується на даних, що збираються із описів вебсервісів, поданих стандартом OpenAPI [12], що додає гнучкості в адаптації для різних платформ і надає доступ до описів функціональності кожного сервісу.

Оптимізація процесу навчання здійснюється за рахунок виключення з розгляду негативних станів та повторних дій у поточному епізоді, що дозволяє скоротити обчислювальні витрати та зосередити пошук на більш релевантних станах та допомагає запобігти зациклюванню в епізоді. Реалізовано також механізми для запобігання зациклюванню шляхом обмеження на кількість кроків без прогресу, щоб уникати петлевих станів. Такі види оптимізації не входять до класичного Q-learning.

Запропонований метод для побудови композицій вебсервісів реалізує Q-learning алгоритм, що відрізняється від класичного адаптацією до специфічних вимог вибору і комбінації вебсервісів.

6. Основні відмінності запропонованого методу

Запропонований метод побудови множини можливих маршрутів має низку принципових відмінностей від традиційних підходів Q-learning.

Динамічний вибір і корекція дій

У запропонованому методі агент обирає послідовність вебсервісів з урахуванням специфічних параметрів QoS. Класичний Q-learning орієнтується переважно на статичне середовище, де набір дій визначений наперед, а в нашому випадку дії можуть залежати від поточного набору досягнутих станів і комбінації входів-виходів кожного сервісу.

Аналіз QoS.

Даний метод включає оцінку якості сервісів на основі QoS для формування винагороди, що дозволяє точніше оцінювати вигоду від кожного вибору сервісу. Кількість параметрів QoS не є фіксованою. Крім того, на відміну від поширених прикладів з оптимізацією QoS ми вводимо глобальні ваги параметрів QoS для врахування як важливості різних аспектів, так і їхній характер впливу: одні слід максимізувати, а інші — мінімізувати (наприклад, доступність, продуктивність треба максимізувати, а час виконання — мінімізувати) у процесі обчислення винагороди за використання певного вебсервісу. Таке узагальнення дає змогу адаптувати модель до конкретних умов задачі, підвищує її гнучкість і точність у виборі оптимального композитного сервісу. Класичний Q-learning зосереджений лише на кінцевій винагороді, не враховуючи QoS та їхньої модальності.

Використання OpenAPI.

Метод використовує дані, які збираються із специфікацій OpenAPI, що додає гнучкості в адаптації для різних платформ і надає доступ до описів функціональності кожного сервісу. Класичний Q-learning не містить аналогічного етапу інтерпретації структурованих вебданих, тоді як наданий метод можливо використовува-

ти з будь-якими стандартами подання вебсервісів, що містять узгоджені описи вхідних і вихідних даних.

Механізми для запобігання зациклюванню.

Враховуючи складність побудови багатокомпонентних вебсервісів, додатково запроваджені обмеження на кількість кроків без прогресу, щоб уникати петлевих станів. Це розширює можливості класичного Q-learning, де подібні зациклювання не є критичними.

Обробка безвихідних і неперспективних станів.

Запропоновано механізм, який запобігає виконанню дій, що не мають переходів для певного стану, і вводиться штраф за такі дії. Тобто агент перевіряє наявність можливих дій у кожному стані та отримує штраф за неперспективні дії. Це корисне розширення для обробки винятків або "неперспективних" станів, але класичний метод Q-learning цього не має.

Встановлення параметрів навчання залежно від розміру вхідних даних.

Автоматично визначаються:

- *кількість епізодів* – залежно від розміру таблиці станів та складності завдання;

- *обмеження кроків на епізод* – залежно від кількості станів і середньої довжини можливого маршруту;

- *порогова кількість кроків без прогресу* – для уникнення зациклювання залежно від складності середовища;

- гіперпараметри Q-learning: *стратегія навчання epsilon і метод зниження параметра epsilon epsilon_decay* – для забезпечення оптимального балансу між дослідженням та експлуатацією;

- *швидкість навчання alpha* – адаптується під максимальну довжину маршруту і стабільність навчання.

Така модифікація Q-learning дозволяє агенту не просто знаходити можливі маршрути через список доступних вебсервісів, а й адаптивно покращувати стратегію вибору на основі зворотного зв'язку від середовища. Це забезпечує системі можливість автоматично коригувати свої рішення, вибираючи такі вебсервіси, які оптимізують не тільки функціональність композитного сервісу, а й параметри QoS. Цей підхід є особливо корисним у динамічних середовищах, де характеристики вебсервісів, такі як час виконання, доступність або пропускна здатність, з часом можуть змінюватися. Використання Q-learning дозволяє автоматично підбирати найкращі варіанти композиції, враховуючи не лише доступні сервіси, а й їхню поточну продуктивність. Основні компоненти запропонованого методу залишаються в межах класичного Q-learning (формула оновлення, ε-жадібна стратегія тощо), але внесені вдосконалення роблять його реалізацію більш гнучкою і надійною, що робить його ефективнішим у середовищах із високою динамікою і складними вимогами до виконання.

7. Застосування алгоритму композиції сервісів для побудови послідовностей навчальних об’єктів

Розглянемо використання запропонованого методу на прикладі задачі вибору послідовності вивчення *навчальних об'єктів* (Learning Object, LO), що використовуються в індивідуальних освітніх траєкторіях [13], які дозволяють персоніфікувати процес здобуття освіти відповідно до потреб та можливостей конкретного студента.

LO – це сукупність інформації, яку об’єднує певна навчальна ціль [14]. Це автономний інформаційних об’єкт, що може багаторазово використовуватися у навчальному процесі. LO може містити лекційні матеріали, практичні завдання, засоби оцінювання знань, представлені як текст, відео, аудіо, програмний код тощо, а контент та роль LO у навчанні однозначно визначаються в його метаописі.

Використання метаописів є основою для автоматизованого аналізу LO, що забезпечує їхнє багаторазове використання для різних навчальних задач, оптимізує їхній вибір та зменшує час цієї процедури, дозволяє шукати, оцінювати та комбінувати LO без додаткового аналізу їхнього контенту.

Метадані LO дозволяють визначити вимоги до студента, який здатний оволодіти знаннями з LO, – вхідні компетенції, та результати такого вивчення – вихідні компетенції. Для забезпечення можливості співставлення компетенцій LO, викладачем курсу спочатку створюється тезаурус навчального курсу $Th_{course} = C_{in} \cup C_{out}$, що містить усі основні компетенції, що їх має отримати студент, який успішно вивчить цей курс $c_{oit_q} \in C_{out}, q = \overline{1,Q}$, та попередні вимоги до студентів $c_{in_p} \in C_{in}, p = \overline{1,P}$ [16]. Після цього викладач здійснює попередній відбір LO, $\forall l_k \in L \exists c_{in}(l_k) \in C_{in}, \exists c_{out}(l_k) \in C_{out}$ так, щоб у цих LO були надані можливості отримання всіх результуючих компетенцій курсу, тобто $\forall c_{out_p} \in C_{out} \exists l_k : c_{out_p} \in c_{out}(l_k)$.

Це дозволяє визначати семантичні відношення між окремими LO – наприклад, знаходити прийнятний порядок вивчення цих LO або перетин знань в їхньому контенті (Рис.1).

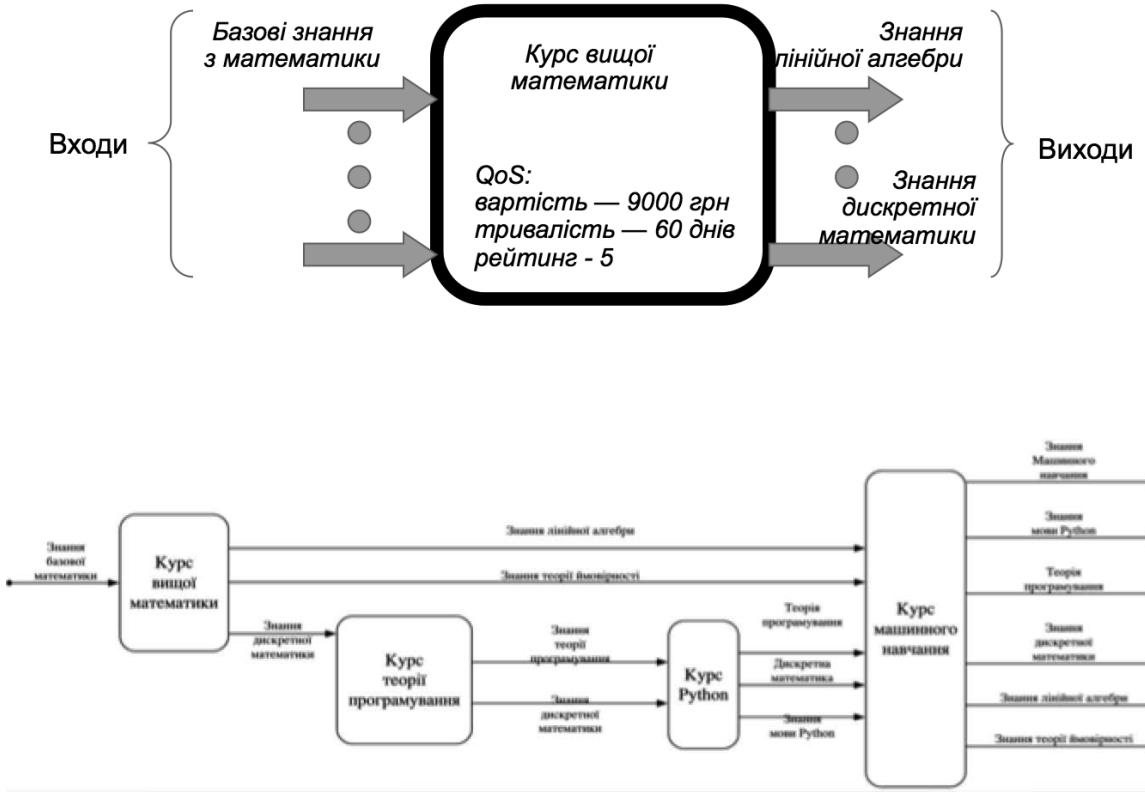


Рис.1. Процес побудови послідовності LO на основі аналізу компетенцій студента

Наступним кроком є оновлення метаданих LO з множини L, щоб пов'язати кожен з них із відповідними вхідними та вихідними наборами компетенцій курсу. Слід звернути увагу, що таке уточнення метаданих LO потрібно виконувати для кожного навчального курсу.

Аналізувати метадані LO можна вручну, але, якщо LO багато та вони досить часто оновлюються, то це забиратиме дуже багато часу, а результати такого аналізу можуть бути неактуальними.

Для автоматизації цього процесу пропонуємо вивчення кожного LO як вебервісу, що має функціональні вимоги і нефункціональні властивості.

Функціональні вимоги сервісу LO – це входи (компетенції студента, які необхідні для вивчення) та виходи (компетенції, які отримуються в результаті вивчен-

ня). Нефункціональні властивості сервісу LO – це характеристики, що можуть впливати на вибір одного із сервісів з однаковими функціональними вимогами – як-от, вартість і час вивчення, які треба мінімізувати, і рейтинг курсу, який треба максимізувати.

Задача полягає в тому, щоб за параметрами наявних LO (вхідні дані – компетенції, що потрібні для вивчення; вихідні дані – компетенції, які можна отримати в результаті навчання [14]) побудувати оптимальну послідовність вивчення LO, що дозволяє отримати певний набір компетенцій – цільовий стан. Цей цільовий стан формалізується як набір C_{out} з Th_{course}.

Сервіси LO функціонують у динамічному інформаційному середовищі, тобто може змінюватися набір доступних сервісів, умови їх використання та контент, а також поточний стан студента. Наприклад, у якийсь момент часу студент розуміє, що для отримання певної компетенції курсу він має доступ лише до таких LO, що подають інформацію природною мовою, якою він не володіє, тобто ситуація може розглядатися як негативний стан. Тоді студент може почекати появи LO тими мовами, котрі він знає, вивчити мову, яку використовують LO з відповідними компетенціями або очікувати на зменшення результуючих компетенцій курсу (у цьому дослідженні динамічні зміни цільового стану не розглядаються).

Крім того, щомиті можуть змінюватися критерії оцінювання та відносні пріоритети параметрів LO. Приміром, іноді час навчання стає важливішим за його вартість. Це викликає потребу у застосуванні методів машинного навчання, які забезпечують можливість автоматизованої побудови маршруту без фіксації його довжини та послідовності елементів.

Запропонований у роботі метод відповідає цим умовам, а тестування його на прикладі з 1000 LO (кількість станів 25913), показало прийнятний час виконання (час формування таблиць: 197.4923 секунд і час виконання: 2.5679 секунд) та високий рівень якості побудованої траєкторії навчання. Тестування виконувалось на Macbook Pro з чіпом Apple M1 Pro з 16Гб пам`яті.

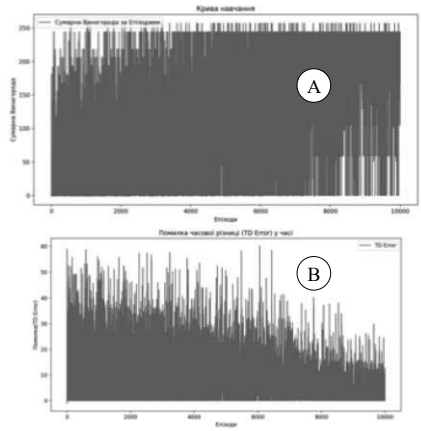


Рис.2. Результати тестування програми на прикладі з 1000 LO

Графік винагороди за епізодами (крива навчання) (Рис.2.A) показує, як змінювалася винагорода під час навчання агента. На початку роботи алгоритму фінальна винагорода за виконання дій була низькою, але поступово її значення збільшувалося, що свідчить про навчання агента і його здатність знаходити кращі маршрути навчання. Позитивна тенденція на графіку свідчить про успішність навчання та покращення алгоритму Q-learning.

Графік TD-помилки (Рис.2.B) відображає помилку часової різниці в процесі навчання. Спочатку помилка була високою, що очікувано, оскільки агент тільки починав вивчати навколишнє середовище. Із часом значення помилки зменшувалося, що свідчить про покращення прогнозів агента щодо майбутніх винагород і стабілізацію процесу навчання.

8. Висновки

Запропоновано новий підхід до побудови композиції вебсервісів, в якому система автоматично генерує можливі маршрути без необхідності попередньо задавати повну інформацію про простір станів і оптимізує їх на основі параметрів QoS. Оптимізація процесу навчання здійснюється шляхом видалення з розгляду неперспективних станів та обмеження повторних дій в межах поточного епізоду, що значно зменшує кількість обчислень та прискорює збіжність алгоритму. Якщо певний стан не

має можливих переходів, то епізод завершується зі штрафом, що запобігає неефективним обчисленням у нерелевантних напрямках. Це дозволяє швидше адаптуватися до змін і динамічно коригувати стратегію вибору сервісів, мінімізуючи час на дослідження та покращуючи ефективність у реальних умовах. Запропоновано засоби запобігання зациклюванню, яке може виникати на великих об'ємах даних.

Надалі для прискорення пошуку оптимальних сервісів рекомендується додати використання евристик для відбору сервісів, алгоритми прискореного навчання та ефективні методи балансу між дослідженням і експлуатацією, а для адаптації до динамічних змін QoS-параметрів – онлайн-навчання, відстеження змін у середовищі та регулярне оновлення стратегії на основі адаптивних алгоритмів. Крім того, для вирішення проблем, пов'язаних з довготривалим пошуком оптимальних сервісів та динамічними змінами параметрів QoS, доцільно застосувати кілька різних стратегій і методів, які дозволять покращувати ефективність навчання та адаптацію агента до змін середовища в реальному часі. Ці рішення допоможуть підвищити ефективність запропонованого методу.

Отож, запропоновано новий підхід до вирішення задачі автоматизації процесу композиції вебсервісів із динамічними характеристиками. Можливі подальші дослідження стосуються підвищення ефективності: покращення швидкості навчання агента, підвищення стабільності роботи, оптимізації таблиці станів і дослідження інших методів навчання з підкріпленням, таких як SARSA та глибокі нейронні мережі.

References

1. *Berners-Lee T.,* Web Services, Program Integration across Application and Organization boundaries, W3C, https://www.w3.org/DesignIssues/WebServices.html
2. Web Services Activity Statement, W3C, https://www.w3.org/2002/ws/Activity
3. *Kashyap V., Bussler C., Moran M.,* Web Service Composition. The Semantic Web: Semantics for Data and Services on the Web. In: The Semantic Web. Data-Centric Systems and Applications. Springer, Berlin, Heidelberg. 2008, pp.233-248 https://doi.org/10.1007/978-3-540-76452-6_10.
4. *Samuel A.,* Some studies in machine learning using the game of checkers. IBM Journal of research and development, 1959, 3(3), pp.210-229.
5. *Mitchell T.,* Machine learning. New York: McGraw-hill. 1997, Vol. 1, No. 9 http://www.pachecoj.com/courses/csc380_fall21/lectures/mlintro.pdf.
6. *Barto A., Sutton R.,* Reinforcement Learning: An Introduction, The MIT Press: Cam-bridge, Massauchsetts, 2018. https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf.
7. *Jang B., Kim M., Harerimana G., Kim J.W.*, Q-learning algorithms: A comprehen-sive classi-fication and applications. IEEE access, 2019.
8. *Lemos A.L., Daniel F., Benatallah B.,* Web service composition: a survey of techniques and tools. ACM Computing Surveys (CSUR), 2015, 48(3), 1-41.
9. *Uc-Cetina V., Moo-Mena F., Hernandez-Ucan R.,* Composition of web services using Mar-kov decision processes and dynamic programming. The Scientific World Journal, 2015 (1).
10. *Kalasapur S., Kumar M., Shirazi B.A.,* Dynamic service composition in pervasive com-puting. IEEE Transactions on parallel and dis-tributed systems, 2007, 18(7), pp.907-918.
11. *Gryshanova I., Rogushyna J.,* Technology of machine learning use for composite web service development. Problems in Programming, 2024, 4, pp.34-43. (in Ukrainian)
12. The OpenAPI Specification, https://swagger.io/specification/.
13. *Rogushina J., Gladun A., Anishchenko O., Pryima S.,* Semantic Support of Personal Learning Trajectory Development, in: UkrPROG 2024, CEUR Vol-3806, 2024, pp.487-505.
14. *Politis D., Tsalighopoulos M., Kyriafinis G.,* Designing Blended Learning Strategies for Rich Content, Handbook of Research on Building, Growing, and Sustaining Quality E-Learning Pro-grams, 2017. DOI: 10.4018/978-1-5225-0877-9.ch017.
15. *Rogushina J., Gladun A., Anishchenko O., Pryima S.,* Semantic analysis of learning objects: thesaurus approach for digital transformation of educational resources,

CEUR Workshop Proceed-ings (2024), CEUR, Vol-3771, 85–99. ceur-ws.org/Vol-3771/paper13.pdf

Одержано: 05.02.25
Внутрішня рецензія отримана: 03.03.25
Зовнішня рецензія отримана: 07.03.25

Про авторів:

Гришанова Ірина Юріївна,
науковий співробітник.
ORCID
http://orcid.org/0000-0003-4999-6294.

Рогушина Юлія Віталіївна,
кандидат фізико-математичних наук, доцент,
старший науковий співробітник.
ORCID
http://orcid.org/0000-0001-7958-2557.

Місце роботи авторів:

Інститут програмних систем
НАН України,
тел. (+38) 066 5501999,
e-mail: ladamandraka2010@gmail.com,

УДК 004.94 https://doi.org/10.15407/pp2025.01.094

В.Д. Міненко

ВІЗУАЛІЗАЦІЯ СЕМАНТИК РІЗНОФОРМАТНИХ ТЕКСТОВИХ ОПИСІВ

Дане дослідження спрямоване на вирішення задачі виявлення семантичного змісту з довільних текстових описів, представлених у різних форматах, з метою подальшої його візуалізації за допомогою сучасних засобів генеративного штучного інтелекту.

Стрімкий розвиток технологій штучного інтелекту надає принципово нові можливості для вирішення як задач аналізу тексту, так і генерації контентів – візуалізацій (у вигляді зображень чи відео). Унаслідок чого можна говорити про інший, сучасний рівень вирішення прикладних задач, що використовують подібну функціональність. Галузь генеративного штучного інтелекту доволі молода і містить чимало не вирішених проблем. Згенерована візуалізація характеризується не лише технічною якістю зображення чи відео, а й адекватністю відображення семантики вхідного текстового опису, яка зазвичай напряму залежить не тільки від можливості обраного застосунку ШІ- генерації, а й від структури та змісту вхідної текстової підказки.

Дана стаття описує алгоритм формування ланцюжка вирішення поставленої задачі від критеріїв вибору засобів реалізації та виокремлення проблем, що потребують вдосконалення та доробки, до визначення схеми композитного рішення. Метод, створенний в рамках запропонованого дослідження, має певні обмеження, а саме: він не підтримує мультимовний контент та не охоплює обробку діалектів, жаргонів, автоматичне визначення мови тексту.

Ключові слова: візуалізація семантики, семантично значущі елементи, генеративний штучний інтелект, обробка тексту природної мови, методи аналізу тексту, токенізація, лемінг, сегментація, модель ШІ-генерації, генеративна змагальна мережа, машинне навчання, моделі з відкритим кодом, метрики оцінювання якості візуалізації, текстова підказка, задача кластерізації, кластерний аналіз.

V. D. Minenko

VISUALIZATION OF THE SEMANTICS OF TEXT DESCRIPTIONS PRESENTED IN VARIOUS FORMATS

This study is aimed at solving the problem of identifying semantics from arbitrary texts presented in various formats and further visualizing it using modern tools of generative artificial intelligence.

The rapid development of artificial intelligence technologies provides fundamentally new opportunities for solving both text analysis tasks and content generation - visualizations (in the form of images or videos). As a result, we can talk about a different, modern level of solving applied problems using similar functionality. The field of generative artificial intelligence is still quite young and contains many unsolved problems. The generated visualization is characterized not only by the technical quality of the image or video, but also by the adequacy of the presentation of the semantics of the input text description, which usually directly depends not only on the possibility of the selected AI tool, but also on the structure and content of the input text prompt.

This article describes the algorithm to form a chain of solving the given task, from the criteria for choosing tools of developments and identifying problems that need improvement or resolving, to determining the scheme of a composite solution. The method created within the framework of the proposed study has certain limitations, namely: it does not support multilingual content and does not cover the processing of dialects, slangs, automatic detection of the language of the text.

Key words: visualization of semantics, semantically meaningful elements, generative artificial intelligence, natural language processing, text analysis methods, tokenization, lemming, segmentation, AI generation model, generative adversarial network, machine learning, open source models, visualization quality evaluation metrics, text prompt, clustering problem, cluster analysis.

Вступ

Стрімке зростання обсягів та різноманітності великих даних і розвиток технологій штучного інтелекту з одного боку висуває нові вимоги до обробки інформації, а з іншого - відкриває принципово інші можливості вирішення задач, що оперують цією інформацією. Метою даного дослідження є визначення методів виявлення семантики неструктурованої текстової інформації та формування алгоритму її візуалізуалізації з максимальним ступенем достовірності.

На сьогодні не існує готового сервісу чи системи, що реалізує поставлену задачу в цілому, охоплюючи весь ланцюжок від аналізу різноформатного (та такого, що походить з різних джерел) тексту природної мови до автоматичної візуалізації виявлених семантичних об'єктів візуалізації. Але існує чимала кількість розробок (методів, бібліотек, моделей), що реалізують окремі функції, а також ті, які можуть і мають бути використані як складові в побудові композитного рішення з певним удосконаленням, розширенням, розвитком і вирішенням інтеграційних питань.

Слід зазначити, що критерії вибору та вага семантичних елементів визначаються, перш за все, предметною областю та цілями класу прикладних задач, на які орієнтований алгоритм.

Якщо результат візуалізації є не лише швидким та яскравим, а передусім інформативним та достовірним, тобто таким, що адекватно відображає саме семантичний вміст вхідного тексту, то реалізація подібного алгоритму та створення методології його побудови уможливлює вирішення цілої низки суттєвих задач на принципово новому інтелектуальному рівні, як-от наприклад:

- візуальний аналіз інформації з метою виявлення суперечливої чи недостовірної інформації;
- візуальний моніторинг змінення в часі об'єктів візуалізації та візуальної сцени в цілому;
- інтелектуалізація систем ухвалення оперативних рішень;
- динамічне формування/коригування стратегії в реальному часі.

Методи аналізу текстів природною мовою

Всю сукупність наявних наразі методів аналізу текстових даних можна поділити на дві великі групи:

- статистичний аналіз,
- лінгвістичний аналіз.

Статистичний аналіз орієнтований на виявлення сенсу тексту за частотним розподіленням слів у ньому. Лінгвістичний аналіз – на виявлення сенсу тексту за його семантичною структурою. Однак казати про належність будь-якого з існуючих підходів до конкретної групи можна лише умовно. Як правило, у реальних задачах обробки тексту доводиться використовувати сучасні похідні з поєднанням методик обох груп з тим чи іншим акцентом.

Більш детальна загальна класифікація наведена на Рис.1.

Далі детальніше розглядаються найбільш значущі методи та методології обробки тексту природної мови.

Морфологічний аналіз. Основні підходи до морфологічного аналізу можна розділити на дві групи [2]: морфологічний аналіз на базі словників та без словників.

Застосування словників забезпечує можливість отримання максимальної інформації за формою відомого слова. Але тут одразу виникає питання повноти словників, що використовуються, та ризик виникнення збоїв на реальних текстах через імовірну наявність помилок.

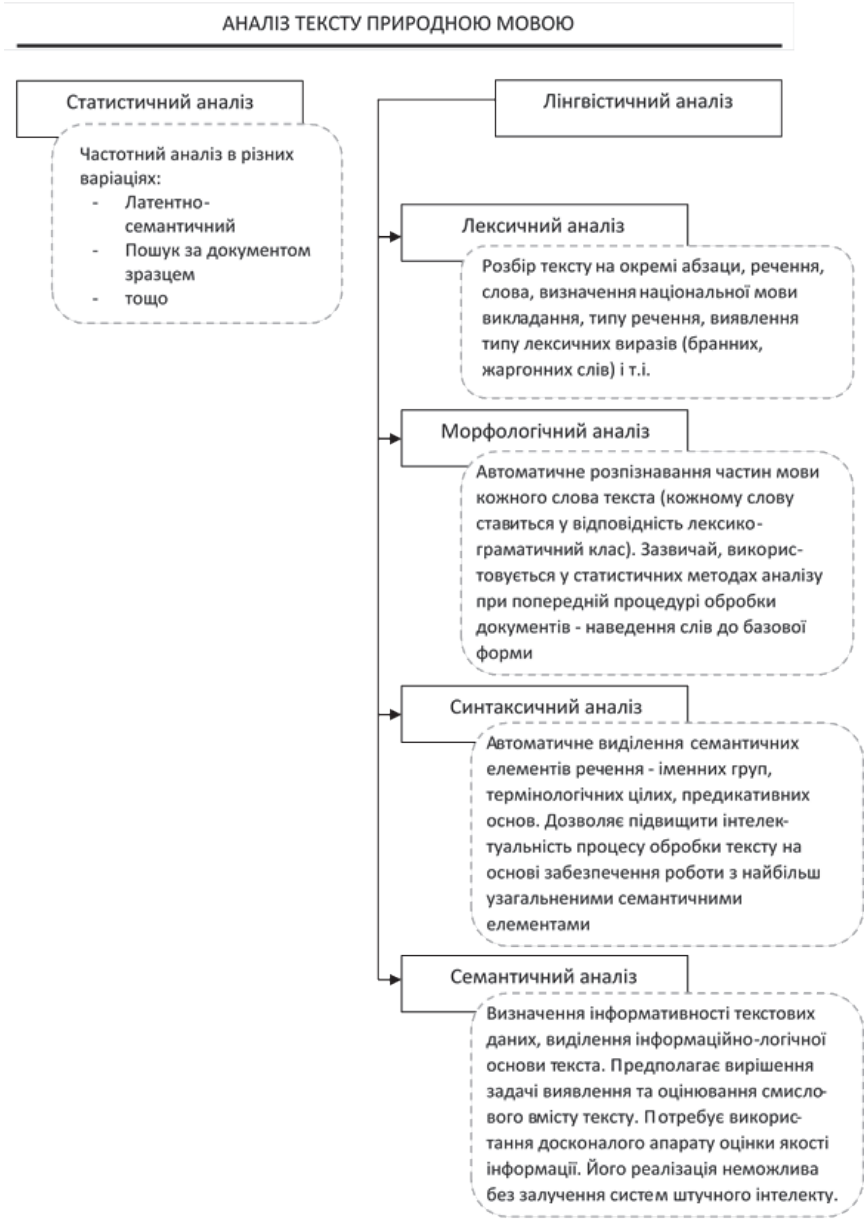


Рис.1. Загальна (умовна) класифікація методів аналізу тексту

Методи без словників для нормалізації слів використовують алгоритми, призначені для перетворення слів у різні граматичні форми. Їх можна поділити на: ймовірно статичні методи та методи лексикону основ і суфіксів.

Слід зазначити, що методи першої групи потребують великої вибірки, а для другої – потрібні великий обсяг лексиконів і методи їх отримання. Ефективність морфологічного аналізу зазвичай намагаються підвищити комбінацією різних підходів.

Морфологічний аналіз може використовувати наступні етапи обробки текстових контентів:

- розбиття тексту на окремі значущі одиниці (абзаци, речення), словоформи, відокремлюючи від тексту знаки, цифри тощо;
- нормалізація словоформ, що має вигляд лематизації або стемінгу;
- морфологічний розбір слова через пошук у лемі суфіксів та закінчень різних частин мов.

Кожний із цих етапів містить цілу низку непростих задач, вирішення яких є не тривіальною задачею.

У [3] пропонується формальний підхід, де для позначення частин мови вводиться множина частин мови:

$$Z = \{z_1, z_2, \dots, z_k\},$$

де z_i - i – та частина мови, k - кількість частин мови в обраній природній мові, а множина слів тексту представляється у вигляді об'єднання k-підмножин різних частин мови, водночас кожне слово тексту може бути віднесене до однієї з цих підмножин:

$$T = \bigcup_{j=1}^{k} W_j,$$

$t_i \in W_j, i = \underline{1,m}, j = 1,\overset{\frown}{k}$, W_j – підмножина слів j-ої частини мови.

Для відображення множини слів T до множини частин речі Z вводиться функція $F(T)$, результатом роботи якої є вектор з показниками приналежності до i-ї частини речі.

$$F: T \to X, F(T) = \{f_1(T), \dots, f_k(T)\} = \{x_1, \dots, x_k\}, i = \overset{\frown}{1,k},$$

де f_i - функція визначення показника приналежності слова до i-ї частини мови, x_i – показник приналежності слова до i-ї частини мови z_i.

Даний алгоритм є словниковим та для проведення аналізу потребує використання таблиць службових слів і таблиць лем та словоформ, з елементами яких здійснює ітераційне співставлення виокремлених із тексту словоформ.

Слід зазначити, що більшість методів даного класу стикаються з проблемою зниження якості аналізу, що обумовлене такими чинниками, як наявність у синтаксичних конструкціях декількох значень, застосування літературного стилю, наявність скорочень у тексті. У [4] автори пропонують вирішення даної проблеми через застосування словника скорочень та виконання попередньої фільтрації слів із низькою частотою появи у тексті. Йомовірно, було б доцільним залучення словників жаргонних, сленгових слів, діалектів, а також здійснювати попередню фільтрацію не лише слів із низькою частотою присутності в текстовому контенті, а й з найвищою, що забезпечить позбавлення «шумових» словоформ (займенників, прийменників тощо).

Також дуже поширеними зараз є методи морфологічного аналізу, засновані на системі машинного навчання [5]. Система машинного навчання здійснює аналіз конкретного тексту (або сукупності текстових контентів) та тренується на ньому, розпізнаючи певні закономірності, і на їхній основі робить деякі узагальнення. Згідно з набутою інформацію про властивості словоформ, що є закономірними у всіх контекстах, які пройшли аналіз, вона може робити прогнози щодо найбільш вірогідної граматичної інтерпретації словоформи у нових текстах. Методи контрольованого машинного навчання роблять прогноз, що є ймовірнісним, після тренування на корпусі текстів, розмічених інформацією про словоформи повністю або частково, а методи машинного самонавчання дозволяють працювати з корпусом, який ще не розмічений. Системи морфологічного аналізу на основі цього методу - це аналізатори, що використовують методику трансформаційних правил, виведених в результаті машинного навчання.

Слід зазначити, що крім загальнолінгвістичних задач, які підлягають вирішенню для досягнення якості аналізу тексту, існують специфічні проблеми кожної конкретної мови, пов’язані із розвитком її морфології, можливим вживанням великої кількості діалектів тощо.

Статистичний аналіз текстового контенту. Найбільш поширеними класами методів даної групи є латентно-семантичний та кластерний аналіз. Мета даних методів полягає у виявленні прихованих закономірностей або очевидних залежностей. Цим і обумовлюється їхнє особливе місце серед величезної кількості алгоритмів пошуку та обробки текстових даних.

Метод латентно-семантичного аналізу. Спершу трохи зупинимось на прак-

тичній значущості методів цього класу. Присутність у текстових контентах полісемії (одне слово має кілька різних значень) та синонімії (кілька слів з однаковим значенням) є стандартною ситуацією для будь-якої природної мови. Одним із можливих шляхів вирішення цієї проблеми – згрупувати слова з однаковими значеннями чи слова із сильною кореляцією. Їх можна представити у вигляді певної прихованої, або «латентної» змінної, яка представлятиме всі ці слова. Звідси й сам термін - «латентно-семантичний аналіз».

Формальніше метод *латентно-семантичного аналізу (LSA)* [6] — це повністю автоматичний метод витягування та виведення взаємозв'язків очікуваного контекстного використання слів в уривках дискурсу. Це нетрадиційний метод обробки природної мови або штучного інтелекту; він не використовує створені людиною словники, бази знань (БЗ), семантичні мережі, граматики, синтаксичні аналізатори, морфології тощо, а за вхідні дані приймає тільки необроблений текст, розібраний на слова, які визначаються як унікальні рядки символів, розділені на значущі фрагменти, як, наприклад, речення або параграфи. LSA є гібридним методом, який використовує комбінацію статистичних та стохастичних технік.

Метод LSA працює на колекції текстових документів. Результатом є матриця «терми-на-текстові документи», її елементи містять частоти використання термів у кожному з документів. Один із найрозповсюдженіших варіантів – LSA, заснований на використанні розкладення вихідної матриці за сингулярними значеннями (SVD). Використовуючи SVD, велика вихідна матриця розкладається на множину з k ортогональних матриць, лінійна комбінація яких є вдалим наближенням вихідної матриці. Згідно з теоремою про сингулярне розкладення, будь-яка дійсна прямокутна матриця X може бути розкладена у добуток трьох матриць:

$$X = U\Sigma V^T,$$

де матриці U та V – ортогональні, а $\mathbf{\Sigma}$ -діагональна матриця, значення на діагоналі якої називаються сингулярними значеннями матриці X. Особливість такого розкладення [7] полягає в тому, що у разі залишення лише k найбільших сингулярних значень, а в матрицях U та V лише відповідні цим значенням стовпці, то добуток отриманих матриць буде найкращим наближенням вихідної матриці X матрицею рангу k ($\dot{X}$):

$$X \cong \dot{X} = U_{lsa}\Sigma_{lsa}V_{lsa}$$

Якщо X - матриця «терми-на-документ», то $\dot{X}$, з одного боку, відображатиме основну структуру асоціативних залежностей, що є в X, а з іншого - не містить зайвої незначущої інформації (шуму).

Таким чином кожний терм та документ представляються за допомогою векторів у загальному просторі розмірності k (так званому просторі гіпотез). Тоді близькість між будь-якою комбінацією термів або документів може бути легко обчислена за допомогою різних метрик відстані (наприклад, скалярний добуток векторів, косинусна, евклідова або манхетенська відстань тощо).

Окреме питання – вибір оптимального значення розмірності k. В ідеалі, k має бути достатньо великим для відображення всієї реально існуючої структури даних. Але в той самий час достатньо малим, щоб не охопити випадкові та маловажливі залежності. Якщо обране k є занадто великим, то метод втрачає свою ефективність та наближається за характеристиками до стандартних векторних методів. Занадто маленьке k не дозволяє вміймати відмінності між схожими словами або документами. Дослідження показують, що зі збільшенням k якість спочатку зростає, а потім починає йти на спад.

На сьогодні відомі щонайменше ймовірнісний, інкрементальний та ієрархічний варіанти методу LSA, що активно застосовуються, зокрема, для автоматичного прогнозування інтересів користувачів у веб, виходячи з накопиченої інформації про їхні уподобання.

Головною перевагою LSA методу є саме його здатність виявляти залежності між словами, коли звичайні статистичні методи безсилі. Також LSA може бути використаний з навчанням (тобто з попередньою тематичною класифікацією текстових контентів) або без навчання (довільне розбиття довільного тексту), що залежить від задачі, яка вирішується.

Одним із головних недоліків латентно-семантичного аналізу є те, що, він розрахований на обробку документів колекції, тож ці документи мають бути доступними. Це обмежує його застосування. До цього можна ще додати значне зниження швидкості обчислень зі збільшенням оьсягу вхідних даних. Як продемонстровано у [8], швидкість обчислень відповідає порядку N^{2k}, де $N = N_{doc} + N_{term}$ – сума числа документів та числа термів, k – розмірність простору факторів.

Методи *Кластерного аналізу* [9] являють собою статистичну процедуру, задача якої полягає в розбитті вибірки інформаційних об'єктів на підмножини, що не перетинаються (жорстка кластеризація) і називаються кластерами. (У випадку м'якої кластеризації один об'єкт може належати кільком кластерам.) Кожен кластер має складатися зі схожих об'єктів, а об'єкти різних кластерів мають істотно відрізнятися один від одного.

Формально, є певна вибірка інформаційних об'єктів $X^l = \{x_1, \dots, x_l\} \subset X$ і функція відстані між об'єктами - $\rho(x, x')$. Треба розбити задану вибірку на кластери, що містять об'єкти, близькі за метрицею ρ. Кожному об'єкту $x_i \in X^l$ призначається мітка (номер) кластера y_i. Алгоритм кластеризації фактично є функцією $a: X \to Y$, яка будь-якому об'єкту $x \in X$ ставить у відповідність мітку кластера $y \in Y$. Множина міток Y в деяких випадках відома заздалегідь, однак частіше завдання полягає у визначенні оптимального числа кластерів з точки зору того чи іншого критерію якості кластеризації.

Методи кластеризації різняться правилами побудови кластерів [10], що є критеріями, за якими визначається «схожість» об'єктів. Кластери можна утворювати, ґрунтуючись на відстані між ними, щільності ділянок у просторі даних, інтервалах або на конкретних статистичних розподілах. Усе залежить від конкретного набору даних та мети використання результатів аналізу.

Методи кластерного аналізу можуть бути застосовані для вирішення цілої низки задач обробки текстових даних, які умовно можна об'єднати в 4 групи:

1) розробка системи класифікації інформаційних об'єктів;

2) дослідження корисних концептуальних схем їх групування;

3) представлення гіпотез на основі дослідження текстових даних;

4) перевірка гіпотез або досліджень для визначення, чи дійсно типи (групи), виділені тим або іншим способом, присутні в наявних текстових даних.

Нескладно помітити, що перелічені групи задач фактично є задачами машинного навчання. Тобто простежується тісний зв'язок між алгоритмами і методами машинного навчання та методами кластерного аналізу.

Одиницею кластерного аналізу є інформаційний об'єкт, заданий вектором ознак. Робота кластерного аналізу спирається на два припущення. Перше – розглянуті ознаки об'єкта в принципі допускають бажане розбиття сукупності об'єктів на кластери. Друге – правильність вибору масштабу або одиниці вимірювання ознак.

Усю сукупність методів кластерного аналізу можна поділити на дві групи: ієрархічні та неієрархічні, кожна з яких включає безліч методів та підходів. Суть ієрархічної кластеризації полягає в послідовному об'єднанні менших кластерів у більші (агломеративні методи) або поділі більших кластерів на менші (дивізимні). Тобто в першому випадку спочатку всі об'єкти є окремими кластерами і послідовно, крок за кроком, схожі об'єкти поєднуються в кластер, кластери збільшуються, а їх кількість зменшується. Друга група – логічна протилежність першій. Спочатку - всі об'єкти

належать одному кластеру, який на наступних кроках алгоритму ділиться на менші кластери, у результаті утворюється послідовність груп, що розщеплюються.

Задача кластеризації є досить суб'єктивною. Для її розв'язання може існувати більше одного правильного алгоритму. Кожен алгоритм дотримується свого набору правил для визначення «подібності» між об'єктами даних. Найбільш відповідний алгоритм кластеризації для конкретної проблеми часто потрібно вибирати експериментально, якщо немає математичної причини віддати перевагу одному алгоритму кластеризації над іншим. Алгоритм може добре працювати на певному наборі даних, але не працюватиме для іншого.

Програмна реалізація алгоритмів кластерного аналізу широко представлена в різних інструментах Data Mining, які дозволяють вирішувати завдання досить великої розмірності.

Взагалі *обробка природної мови* (Natural language processing - NLP) є загальним напрямком штучного інтелекту і математичної лінгвістики. NLP вивчає проблеми комп'ютерного аналізу і синтезу природних мов. Що ж до штучного інтелекту, аналіз означає розуміння мови, а синтез - генерацію грамотного тексту. Більшість NLP-методів є методами машинного навчання.

Методи машинного навчання в загальному процесі семантичного аналізу тексту. Машинне навчання (ML) є одним із визнаних успішних методів обробки даних для отримання з них корисної інформації. Їхньою особливістю є не прямий розв’язок задачі, а навчання на множині подібних прикладів, що дозволяє використовувати ці методи для обробки великих обсягів даних та виявляти в них нові, нетривіальні, корисні та доступні для інтерпретації знання. Існує твердження, що алгоритми ML вчаться витягувати інформацію із даних тим краще, чим більше даних для них доступно [11]. Окрім методів штучного інтелекту, під час розробки моделей машинного навчання як допоміжні використовуються засоби математичної статистики, чисельних методів, методів оптимізації, теорії ймовірностей, теорії графів тощо [11].

Найбільш використовуваними варіантами застосування моделей ML для обробки текстових даних – вирішення задач їх *класифікації* та *кластеризації*.

Класифікація [12] – встановлення функціональної залежності між вхідними і дискретними вихідними змінними. За допомогою класифікації вирішується завдання приналежності об’єктів до одного з відомих класів.

Кластеризація [12] – групування об’єктів на основі їхніх властивостей. Об’єкти в кластері мають бути схожими і відрізнятися від об’єктів інших кластерів. Чим більша схожість об’єктів усередині кластера і чим більше відмінностей між кластерами, тим точніша кластеризація.

Методи машинного навчання поділяють на дві основні категорії [13]: навчання з учителем (supervised) та навчання без учителя (unsupervised). Методи навчання з учителем поділяють вхідні дані на набір наперед заданих класів. Для навчання такого класифікатора потрібна навчальна вибірка, яка містить марковані зразки різних класів. Навчальна вибірка має бути репрезентативною, тобто містити варіативний контент із різними характеристиками (класифікаторами), щоб моделі могли відповідно навчитися їх розрізняти. З використанням цього набору даних можна навчити модель розпізнавати ознаки того чи іншого класу контенту.

Методи навчання без учителя не потребують навчальних даних, проте вони не ставлять у відповідність вхідним даним певний клас, а лише вивчають закономірності у вхідних даних та поділяють вхідні дані на кластери.

У [13] автори систематизували існуючі типи класифікаторів за різними критеріями, результат наведений у таблиці 1 нижче.

Таблиця 1. Різновидності підходів до класифікації залежно від критеріїв

Критерій	**Тип**	**Короткий опис**
Використання/ не використання навчальних даних	Класифікація з учителем	Вхідні дані поділяють, використовуючи набір зразків як навчальні дані
	Класифікація без учителя	Підходи відомі як кластеризація. Не беруть до уваги мітки навчальних даних для класифікації вхідних даних
	Напівавтоматичне навчання	Навчання відбувається з використанням даних як з мітками, так і без
Врахування/ не врахування будь-якого припущення про розподіл вхідних даних	Параметричні класифікатори	Припускається, що функція щільності ймовірності для кожного класу відома
	Непараметричні класифікатори	Класифікатори не обмежуються жодними припущеннями про розподіл вхідних даних
Розгляд одного класифікатора або ансамблю	Один	Використовується єдиний класифікатор для призначення мітки для об’єкта
	Ансамбль	Під час визначення мітки для об’єкта враховуються результати кількох класифікаторів
Використання/не використання технології жорсткого поділу, де кожен об’єкт належить лише одному кластеру	Жорсткий класифікатор	Технології жорсткої класифікації не враховують подальші зміни різних класів
	М’який (нечіткий) класифікатор	Нечіткі класифікатори моделюють поступові граничні зміни, забезпечуючи оцінку ступеня подібності всіх класів
Видача класифікатором розподілу ймовірності належності до всіх класів	Імовірнісний класифікатор	Класифікатор здатен для заданого зразка оцінити розподіл ймовірностей на множині класів
	Неймовірнісний класифікатор	Підхід визначає лише найбільш придатний клас для вхідного образу

Найбільш поширеними методами ML для задач класифікації [14] є штучні нейронні мережі [15], логістична регресія [15], метод опорних векторів [15] та випадковий ліс [16]. Порівняльна характеристика перелічених та інших методів класифікації наведена в [17].

З огляду на тематику даного дослідження, вирішення задачі класифікації для вхідної інформації може дозволити, наприклад, вибрати з множини вхідних повідомлень лише ті, що будуть значущими для візуалізації (стосуватися конкретної предметної області чи події), обрізати фейковий контент [17] чи просто зайвий, класифікувати повідомлення на такі, що відображають статичну та динамічну інформацію, або за призначенням (кому інформація має бути делегована), за темою, відправником, датою, типом повідомлення тощо [1].

Генеративний штучний інтелект

Основною відмінністю технологій генеративного штучного інтелекту (ШІ) є їхня здатність створювати нові дані будь-якого типу. Технології ШІ-генерації використовують штучний інтелект для створення на основі вхідних текстових описів нового контенту, який досить точно (текстово або візуально) відображає зміст і контекст вхідного тексту. Умовно їх можна розділити на три групи за формою результуючого контенту: *text-to-text* (генерується текст), *text-to-image* (генерується зображення) та *text-to-video* (генерується відео).

Технології генерації тексту (або «генерування природної мови») часто базуються на процесах Маркова та на глибоких генеративних моделях. Штучний інтелект або моделі машинного навчання генерують мову згідно правил граматики, синтаксису чи лексики. Модель генерації починається на концептуальному рівні (вибір з контенту даних для перетворення), зменшується до досконалих правил мови (правопис, граматика, вибір слів), та, врешті-решт, створює речення як ланцюжок слів.

Більшість сучасних сервісів *text-to-image* є поєднанням моделі, що обробляє природну мову (NLP), та генеративної змагальної мережі (GAN).

Технології *text-to-image* вміють «прочитати» фрагмент тексту та згенерувати зображення відповідно до його змісту. Як правило, це реалізується трьома кроками. Спочатку виконується аналіз вхідного опису та виявляється значуща інформація (зазвичай для цього використовуються методи аналізу природної мови): ключові слова, сутності, контекст опису. Далі на основі отриманої інформації створюється зображення, використовуючи здебільшого генеративні змагальні мережі (GAN) [31]. Й нарешті вдосконалення результуючого зображення – багатоітераційний процес аналізу та оптимізації версій зображення до досягнення бажаної якості та ступеня відображеня семантичного вмісту.

Технології *text-to-video* уможливлюють створення відеороліків на основі вхідної текстової підказки, та зазвичай охоплюють кілька підгалузей штучного інтелекту: обробку природної мови, комп'ютерний зір та машинне навчання. Стрімкий розвиток технологій генерації відеоконтенту здебільшого пов'язаний із розвитком дифузійних моделей (Stable Diffusion - SD). Вхідні описи можуть мати чималий обсяг. Будь-яке його змінення, навіть додавання/видалення одного слова в/з опис(у), може кардинально впливати на результат генерації. Кожне слово текстової підказки відіграє ключову роль у створенні відео. Приблизний алгоритм генерації можна розглянути на прикладі відомої технології T2V. Алгоритм T2V полягає в наступній послідовності основних кроків:

1) аналіз та інтерпретація вхідного тексту за допомогою методів токенізації з метою визначення його семантики – контексту, значення;

2) вибір відповідних візуальних ефектів та анімації, планування відеоконтенту на основі вхідної текстової підказки;

3) створення візуальних елементів на кшталт 3D-моделей або анімації. Для цього можуть бути використані GAN моделі або ці елементи можуть бути витягнуті просто з наявної бібліотеки відеоматеріалів;

4) побудова з отриманих візуальних об'єктів послідовності, що відповідає вхідному опису, додаючи в цю послідовність переходи та, можливо, синхронізуючі її зі звуком.

Слід зазначити, що розвиток ШІ технологій відеогенерації припадає лише на останні кілька років, тому існує ще багато невирішених проблем та обмежень. Так, наприклад, досі залишається серйозною проблемою генерація руху між відеокадрами.

Аналіз існуючих рішень

Задачі обробки природної мови є досить складними, але більшість із них уже реалізовано в існуючих готових застосунках та бібліотеках аналізу тексту. Генеративні моделі хоча і є досить молодою галуззю, однак вже існує чимала кількість го-

тових рішень. Тому першочергова задача полягає у виборі засобів реалізації з урахуванням вимог до функціональності та можливості подальшої інтеграції в систему, розвитку та вдосконалення.

Серед існуючих інструментів обробки тексту слід виділити:

- Морфологічний аналізатор *pymorphy2* [18] реалізований мовою програмування Python з додатковими розширеннями C ++. Він уміє надавати слову потрібної форми. Наприклад ставити у множину, або змінювати відмінок; повертати граматичну інформацію про слово (число, рід, відмінок, частина мови тощо). Використовує великі ефективно закодовані словники, створені на основі даних OpenCorpora та LanguageTool. Для забезпечення морфологічного аналізу розроблено набір лінгвістично мотивованих правил. Для російської мови pymorphy2 забезпечує ультрасучасний морфологічний аналіз. Але для української мови - вимагає більше специфічних правил для обробки слів позасловникового запасу, а також потребує анотованого корпусу української мови, бо підтримка української мови в цьому аналізаторі поки є експериментальною.

- *Text Analyzer* [19] – безкоштовний інструмент для пошуку та оптимізації ключових слів у тексті. Часто використовуються у вебсередовищі для аналізу сайтів, пошукових запитів, аналізу описів застосунків для Google Play та App Store тощо. Виконує частотний аналіз тексту, дозволяє виділяти ключові слова та спам.

- *SenseClusters* [20] – пакет програм (переважно мовою Perl), що дають можливість користувачу віднести до одного кластеру схожі контенти, використовуючи методи некерованого машинного навчання. Є досвід використання інструментів SenseClusters для розділення слів за їхнім сенсом, категоризації електронної пошти, дискримінації імен. Підтримує кілька різних методів кластеризації тексту. Включають власні методи SenseClusters techniques та LSA метод.

SenseClusters базується тільки на лексичних характеристиках та не використовує ніяких навчальних даних або зовнішніх джерел знань. Це обумовлює його мовну незалежність. Єдиною умовою є можливість токенізації мови за допомогою регулярних виразів Perl, що задаються користувачем. Загалом SenseClusters можна використовувати для вирішення будь-якої задачі, що потребує розпізнавання контекстуально схожих одиниць тексту або слів, які зустрічаються в подібних контекстах.

- Бібліотека для роботи з матрицями *JAMA* [21]. Базовий пакет лінійної алгебри для Java. Надає класи рівня користувача для побудови реальних щільних матриць і керування ними. Підтримує п’ять фундаментальних матричних розкладів, в тому числі сингулярне розкладання прямокутних матриць, що обумовлює широке використання бібліотеки для реалізації LSA методів.

- Електронний словник української мови *ВЕСУМ* [22] - це великий словник словозміни української мови, основними компонентами якого є реєстр лем, коди класів словозміни й правила генерації словоформ на основі цих кодів, а також застосування елементів програмованої логіки. Виконує завдання морфологічного аналізу й синтезу. Перше полягає в лематизації (зведенні окремої словоформи до леми) й присвоєнні цій словоформі відповідних граматичних тегів, а друге передбачає генерування всіх словоформ із певної леми з відповідними граматичними ознаками-тегами. Підтримує «динамічне тегування», а саме обробку утворюваних слів (складних іменників, прикметників, прислівників) шляхом розбиття їх на складові й присвоєння цим складовим відповідних граматичних ознак-тегів [23].

- *NLP UK* [24] - інструмент для аналізу та обробки української мови на основі словника ВЕСУМ (що використовується для тегування лексем) та двигуна LanguageTool (для аналізу текстів). Має підтримку токенізації, лематизації, частиномовного аналізу та базового зняття омонімії. Застосовувався на python3 та java.

- Браунський корпус української мови *БрУК* [25] відкритий, збалансований за жанрами та в майбутньому проанотований корпус сучасної української мови обся-

гом 1 млн слововживань. Корпус побудований на засадах, покладених в основу відомого корпусу англійської мови Brown. Його частина, проанотована за сутностями та готова для автоматичного анотування сутностей (люди, організації, локації та різне); містить векторні представлення слів, простий у використанні токенизатор (на абзаци, речення та слова) тощо.

- *UD Ukrainian* [26] - корпус дерев залежностей для української мови. Містить 122 тисячі токенів у 7000 реченнях художньої літератури, новин, статей, думок, Вікіпедії, юридичних документів, листів, дописів і коментарів за останні 15 років та першу половину 20 століття.

- *Java CoreNLP* [28] – пакет для обробки природної мови на Java. Дозволяє отримувати лінгвістичні анотації для вхідного текстового контенту, включаючи токени, речення, частини мови, іменовані сутності, числові та часові значення, проводити аналіз залежностей, зв'язків, виокремлювати почуття, настрої (тобто емоційні аспекти) та посилання на джерела цитування. Наразі, CoreNLP підтримує 6 мов (арабську, китайську, англійську, французьку, німецьку та іспанську, і, на жаль, не підтримує українську мову).

- *NLTK* (Natural Language Toolkit) [29] — це платформа для розробки програм обробки природної мови мовою програмування Python. Надає зручні інтерфейси для багатьох мовних корпусів, а також бібліотеки для обробки текстових даних, а саме: класифікації, токенізації, стемінгу, розмітки частин мови (POS-тегування), синтаксичного та семантичного аналізу.

- *Stanza* [27] – це пакет засобів на Python для аналізу природної мови (для понад 70 мов). Містить інструменти для декомпозиції тексту у списки речень і слів, для створення базових форм цих слів, визначення частин мови та морфологічних особливостей. Stanza побудовано з високоточними компонентами нейронної мережі, які забезпечують ефективне навчання та оцінку за допомогою власних анотованих даних. Забезпечує надійну текстову аналітику, включаючи токенізацію, розширення багатослівних маркерів (MWT), лематизацію, визначення тегів частин мови (POS) і морфологічних ознак, синтаксичний аналіз залежностей і розпізнавання іменованих об'єктів. Модулі побудовано на основі бібліотеки PyTorch. Крім того, Stanza включає інтерфейс Python до пакета Java CoreNLP і успадковує звідти додаткову функціональність, таку як синтаксичний аналіз, кореферентна роздільна здатність та зіставлення лінгвістичного шаблону.

Проведений аналіз дозволив визначити основні критерії вибору засобів обробки тексту для подальшого використання у вирішенні задачі, а саме - підтримуванні функціональності, відкритості коду та мультимовності.

Інша група – генеративні моделі. Детальний опис проаналізованих генеративних моделей наведений в [30]. Здійснений аналіз дозволив сформувати наступну множину критеріїв вибору такої моделі для реалізації задачі:

- достатня смислова якість візуалізації (тобто можливість отримання адекватного результату);
- відкритість коду, що дає можливість удосконалення, доробки та кастомізації моделі;
- доступна вартість (в ідеалі безкоштовний варіант);
- простота та зручність використання;
- технічна якість візуалізації.

Задача візуалізації семантики текстового контенту

Задача полягає у створенні автоматизованого алгоритму, що реалізує візуалізацію саме семантичного вмісту вхідних текстових даних, отриманих з різних джерел: повідомлення чатів, месенджерів, контент електронних листів, текстові файли тощо. Загальний технологічний ланцюжок вирішення задачі на верхньому рівні наведений на рисунку 2.

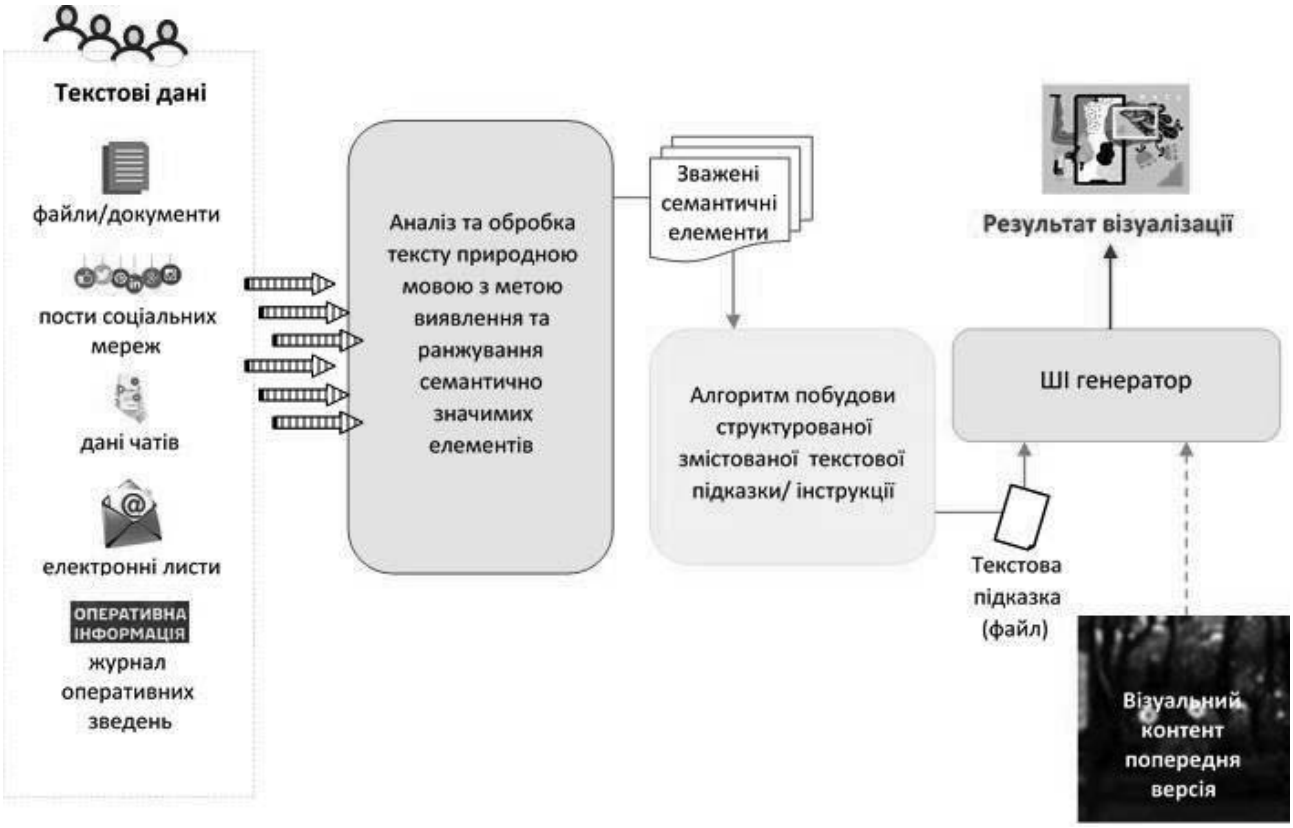


Рис. 2. Алгоритм вирішення задачі візуалізації семантики тексту

Алгоритм має забезпечувати побудову файлу інструкцій (підказок) для візуалізації на базі вибраних в ході аналізу первинних текстових даних ключових семантичних елементів. Файл інструкцій є вхідним для моделі text-to-image та має містити підказку у вигляді структурованих текстових даних, структура та зміст яких забезпечує якісну, достовірну за семантикою візуалізацію. Фактично це є перетворенням тексту в текст визначеної структури зі збереженням ключових семантичних елементів, виявлених під час обробки тексту.

Результат візуалізації є графічним підґрунтям для визначення певних показників, що дозволяють математично оцінити результат вирішення прикладної задачі. Це може бути, приміром, ступінь зміни системи в часі (від попередньої візуалізації) або достовірність вхідної інформації. Такі метрики якості даватимуть можливість оцінити та удосконалити не лише результат візуалізації, а й алгоритм в цілому.

Визначення ключових семантик у вхідному текстовому описі. Ця підзадача в дечому подібна до задач автоматичного реферування, де кінцевою метою є формування стислого та змістовного конспекту, але він формується саме з ключових семантик, виявлених під час аналізу й обробки. Це є однією з найскладніших задач обробки природної мови. Задача даного рівня полягає у виділенні з первинних текстових даних семантичних елементів - ключових фраз (фрагментів, речень), що відображатимуть семантику вхідного тексту.

Обробка тексту природної мови включає наступні основні задачі (див. рис. 3): поділ тексту на токени, переведення всіх літер у нижній регістр, виокремлення основи слова, отримання базової словникової форми слова, видалення стопслів, нормалізацію тексту, розмічування тексту на частини мови, виявлення іменованих сутностей і, нарешті, витягнення з неструктурованого або мало структурованого тексту структурованої інформації, такої, як сутності, зв’язки між ними, їхні атрибути.

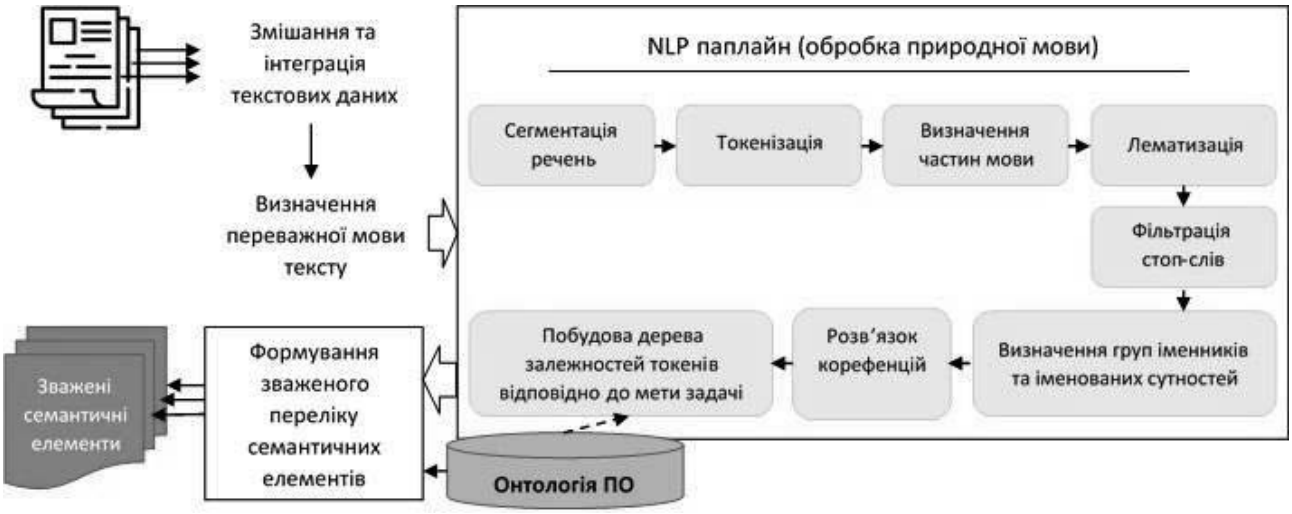


Рис. 3. Задачі обробки текстових даних

Формування файлу підказки. Зрозуміло, що такий алгоритм не може бути повністю універсальним. Досягти високого рівня ефективності (семантичної достовірності зображення відповідно до вхідних даних) можна тільки враховуючи особливості класу прикладних задач, де він буде застосовуватися, та відповідної предметної області. Це дозволяє сформувати адекватну систему характеристик, що уможливлюють певну якість отримання візуалізації. Це можуть бути класи значущих семантик, основні види дій та станів, географічні локації, часові характеристики, використання версійності та номер попередньої версії.

Алгоритм формування файлу підказки наведений на Рис. 4.

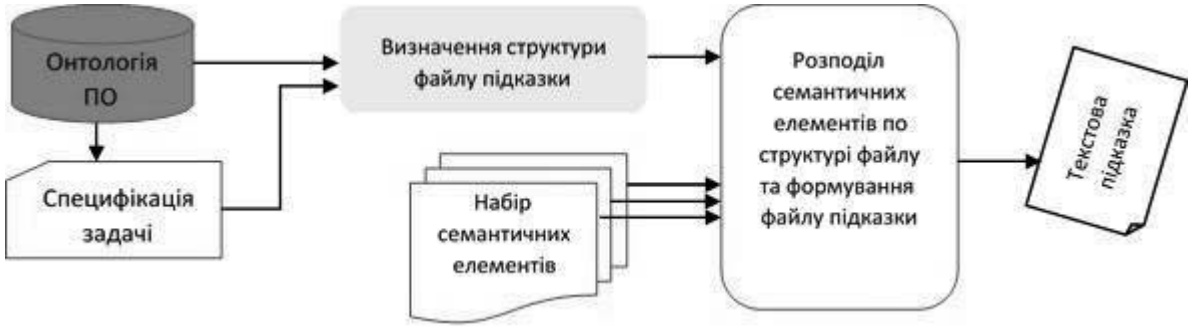


Рис. 4. Формування файлу підказки

Висновки

В роботі проведено аналіз методів обробки тексту природною мовою з метою виявлення його семантичного змісту та технологій генеративного штучного інтелекту, а також найбільш вживаних на сьогодні застосунків, як для аналізу текстових даних, так і ШІ-генерації. Це дозволило виокремити спільні характеристики існуючих систем, їхні недоліки та сформувати основні критерії вибору застосунку ШІ-генерації для його подальшої інтеграції до загальної системи для реалізації функцій створення візуального контенту. На основі проведеного аналізу вироблено основний ланцюжок вирішення задачі, специфіковано складові підзадачі та визначені головні етапи проведення досліджень. Наукова новизна запропонованого алгоритма полягає у:

1) вдосконаленні процесу обробки тексту шляхом зведення та інтеграції текстових даних з різних джерел;

2) створенні технології інтеграції сервісів обробки тексту та візуалізації шляхом алгоритмізації та автоматизації етапу переходу від сформованого списку значущих візуальних елементів тексту до візуалізації цих семантик засобами ШІ генерації;

3) визначенні метрик якості результатів візуалізації.

Подальший розвиток досліджень передбачає, перш за все, вдосконалення аналізу текстових даних з метою отримання більш якісної семантики відповідно до особливостей та цілей вирішення прикладної задачі, а саме охоплює вирішення наступних задач:

1) створення системи критеріїв значущої інформації в аспекті прикладної задачі, що вирішується. Наприклад, як системи з двох онтологій: онтології предметної області, де однією з характеристик сутності чи зв'язку між сутностями є вага (тобто, це саме семантична вага, що враховує інтереси прикладної задачі, а не вага терміну, що визначається частотою його вживання) та онтології задачі з визначенням ролей користувачів, джерел надходження текстової інформації, сутностей, що визначаються практичними цілями реалізації.

2) вирішення задачі класифікації вхідних текстових даних за заданою системою критеріїв. Як класифікатори можуть використовуватись тематика, джерела інформації, ролі кінцевих користувачів, локації, показники часу, яким датується вхідний контент тощо (залежно від практичної задачі, що вирішується);

3) створення технології "очищення" нелітературної мови (обробка суржика, сленгу, діалектів) та нормалізації тексту із залученням словників синонімів;

4) удосконалення процесу аналізу тексту з урахуванням критеріїв значимості у виявленні ключових семантичних фрагментів.

Ще одна задача, яка може бути реалізована на основі класифікації вхідних текстових даних/вихідних візуалізацій за ролями користувачів за попередньо визначеними критеріями/правилами, - це створення системи "маршрутизації завдань", яка пов'язує вхідні дані й, відповідно, отриманий відеоконтент із адресатом (роль користувача).

Література

1. Yakymenko, D. O., Kataieva, Ye. Ye. Methods and Means of Intelligent Analysis of Text Documents. Вісник Черкаського державного технологічного університету, 2022. №2, С. 43-52. https://er.chdtu.edu.ua/handle/ChSTU/4165
2. Bisikalo, O., Vysotska, V., Burov, Y. Conceptual Model of Process Formation for the Semantics of Sentence in Natural Language. Proceedings of the 4th International Conference on Computational Linguistics and Intelligent Systems (COLINS 2020). Volume I: Main Conference Lviv, Ukraine, April 23-24, 2020, 27p. CEUR Workshop Proceedings, available at: http://ceur-ws.org/Vol-2604/paper12.pdf
3. Іващенко О. О., П'ятикоп О. Є. Моделювання методу морфологічного аналізу україномовного тексту. Наукові праці ДонНТУ, Серія "Інформатика, кібернетика та обчислювальна техніка", 2020. № 2(31). С. 65 -72. https://iktv.donntu.edu.ua/wp-content/uploads/2021/04/08_Yvashchenko-Piatykop-1.pdf
4. Singh, J., Singh, G., Singh, R. Morphological evaluation and sentiment analysis of Punjabi text using deep learning classification. Journal of King Saud University: Computer and Information Sciences. 2021, Vol.33, № 5. P. 508 - 517. https://www.sciencedirect.com/science/article/pii/S1319157818300612?via%3Dihub
5. Яровий А., Кудрявцев Д., Крилик Л. Удосконалення методу семантичного аналізу тексту. Інтелектуальні Інформаційні Технології. 2020, С. 34-36. https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/30887/WORK-IES-2020-34-36.pdf?sequence=1
6. Landauer T., Foltz P., Laham D. Introduction to Latent Semantic Analysis. Discourse Processes, 1998. № 25. P. 259–284.
7. Press, W., Teukolsky, S., Vetterling, W. Singular Value Decomposition. Numerical Recipes in C., 2nd edition. Cambridge: Cambridge University Press, 1992. P. 59 -71.

8. Deerwester, S., Dumais, S., Furnas,G. Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science, 1990. Vol. 41 № 6. P. 391–407. URL: http://wordvec.colorado.edu/papers/Deerwester_1990.pdf
9. Основні поняття кластеризації та постановка задачі. https://csc.knu.ua/media/study/asp/mod_probl_inf_tech_sys_analysis_ivohin/lecture/lec11.pdf
10. Методи кластерного аналізу. Ієрархічні методи. https://moodle.znu.edu.ua/pluginfile.php/486140/mod_resource/content/1/Лекція%2010.pdf
11. Grolinger, K., Hayes, M., Higashino, W.A. Challenges for MapReduce in big data in Proc. IEEE World Congr. Services (SERVICES), 2014, pp. 182-189. https://ir.lib.uwo.ca/cgi/viewcontent.cgi?article=1095&context=electricalpub
12. Коновалова К. Машинне навчання: методи та моделі: підручник для бакалаврів, магістрів та докторів філософії спеціальності 051 «Економіка»// Харків: ХНУ імені В. Н. Каразіна, 2020. 280 с. https://www.researchgate.net/publication/345765254_MASINNE_NAVCANNA_METODI_TA_MODELI
13. Новіков О.М., Лавренюк М.С. Огляд методів машинного навчання для класифікації великих обсягів супутникових даних. Системні дослідження та інформаційні технології. 2018. № 1. С. 52-71. http://jnas.nbuv.gov.ua/article/UJRN-0001075162 (дата звернення: 01.06.2024)
14. Maulik, U., Chakraborty, D. Remote Sensing Image Classification: A survey of support-vectormachine-based advanced techniques. IEEE Geoscience and Remote Sensing Magazine. 2017. Vol. 5, № 1. P. 33–52.
15. Bishop C.M. Pattern Recognition and Machine Learning. NY: Springer. 2006. 738 p.
16. Gislason, P.O., Benediktsson, J.A., Sveinsson, J.R. Random forests for land cover classification. Pattern Recognition Letters. 2006. Vol. 27 N 4. P. 294–300.
17. Праздніков В.О., Сугоняк І.І. Моделі та методи машинного навчання для розпізнавання фейкового контенту. Технічна інженерія, 2023. Том 2 №92. С.131-136. https://www.researchgate.net/publication/376878645_Modeli_ta_metodi_masinnogo_navcanna_dla_rozpiznavanna_fejkovogo_kontentu
18. Морфологічний аналізатор Pymorphy2. https://pymorphy2.readthedocs.io/en/stable/
19. Text Analyzer. https://asomobile.net/en/blog/text-analyzer/
20. Sense Clusters. https://metacpan.org/pod/Text::SenseClusters
21. JAMA: A Java Matrix Package. https://math.nist.gov/javanumerics/jama/
22. Рисін, А., Старко, В. Великий електронний словник української мови (ВЕСУМ). Вебверсія 6.1.0. 2005-2023. https://vesum.nlp.net.ua/
23. Старко, В., Рисін., А. Великий електронний словник української мови (ВЕСУМ) як засіб NLP для української мови. Галактика слова. 2020. С.134-141. https://www.researchgate.net/publication/344842033_Velikij_elektronnij_slovnik_ukrainskoi_movi_VESUM_ak_zasib_NLP_dla_ukrainskoi_movi_Galaktika_Slova_Galini_Makarivni_Gnatuk
24. LanguageTool API NLP UK. URL: https://github.com/brown-uk/nlp_uk
25. Браунський корпус української мови. https://github.com/brown-uk/corpus
26. Universal Dependencies corpus for Ukrainian. https://github.com/UniversalDependencies/UD_Ukrainian-IU/tree/master
27. Stanza – A Python NLP Package for Many Human Languages. https://stanfordnlp.github.io/stanza/
28. Manning, C., Surdeanu, M., Bauer, J. The Stanford CoreNLP Natural Language Processing Toolkit. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2014. P. 55-60
29. Natural Language Toolkit: Documentation. 2024. https://www.nltk.org/
30. Міненко В., Аналіз застосування ШІ-генераторів для розв'язання складних бізнес-задач. Системи керування та комп'ютери, 2024, № 4. С. 10 – 18.
31. Іванов А., Онищенко В. Методи генерації зображень з використанням мереж GAN. Адаптивні системи автоматичного управління. 2023. Том 1 №42, С.153-159 https://asac.kpi.ua/article/view/279109

Одержано: 10.01.2025
Внутрішня рецензія отримана: 17.01.2025
Зовнішня рецензія отримана: 19.01.2025

Про автора:

Міненко Валерій Дмитрович,
аспірант
https://orcid.org/0009-0003-5299-6786

Місце роботи автора:

Інститут програмних систем
НАН України.
Засновник, Twigames Inc.
3422 Old Capitol Trail, Suite# 241,
Wilmington, DE 19808, US
Моб. тел.: +380(68) 807 49 48
E-mail: valerii@twigames.net

УДК 004.424, 004.94, 007.52 https://doi.org/10.15407/pp2025.01.110

Р.С. Шевченко, А.Ю. Дорошенко, В.О. Лесик, О.В. Савчук, О.А. Яценко

ІНТЕРФЕЙСНО-ОРІЄНТОВАНИЙ ПІДХІД ДО ЗАСОБІВ МОДЕЛЮВАННЯ МУЛЬТИАГЕНТНИХ СИСТЕМ

Високорівневі системи моделювання мультиагентних систем здатні суттєво прискорити процес розробки та впровадження програмного забезпечення для автономних мультиагентних місій. Оскільки різні задачі вимагають уваги до специфічних аспектів моделювання, інтерфейсно-орієнтований підхід може стати ефективним засобом адаптації різних моделей середовища до заданих інтерфейсів поведінки. Моделювання мультиагентних систем охоплює широкий спектр процесів – від фізичного руху агентів до формування стратегій поведінки та організації їхньої взаємодії. Інтерфейсно-орієнтований підхід дає змогу створювати гнучкі мультиагентні системи моделювання, які можуть виконувати широкий спектр завдань, забезпечуючи швидку розробку моделей поведінки в мультиагентному середовищі та SITL-тестування низькорівневого коду таких складних систем як автопілоти. В рамках підходу проєктування системи розпочинається з визначення інтерфейсної взаємодії між її компонентами, що дає можливість повторного використання коду та створення індивідуальної реалізації компонентів для специфічних експериментальних завдань. Ведуться роботи над прототипом системи мультиагентного моделювання Blefusku, однією з особливостей якої є інтерфейсно-орієнтований підхід і підтримка високорівневих моделей опису поведінки. Основні модулі системи – це середовище моделювання, яке містить модель середовища та формує сигнали сенсорів, і контейнер агентів, що відповідає за поведінку об’єктів і змістовну частину комунікаційного середовища. Інтерфейс описаний як сервіс gRPC, що дозволяє з’єднувати різні компоненти, написані у різних середовищах програмування. Комунікаційний рівень ґрунтується на протоколі MAVLink. Поведінка визначається об’єктом, у якому запрограмована реакція на дані з сенсорів та повідомлення і періодичні активності. Як приклад наведено фрагмент сценарію пошуково-рятувальної місії за участі дронів.

Ключові слова: агенти, інтерфейсно-орієнтований підхід, моделювання, мультиагентна система, робототехніка, сенсори, MAVLink, SITL, ROS.

UDC 004.424, 004.94, 007.52

R.S. Shevchenko, A.Yu. Doroshenko, V.O. Lesyk, O.V. Savchuk, O.A. Yatsenko

INTERFACE-ORIENTED APPROACH TO MODELING TOOLS FOR MULTI-AGENT SYSTEMS

High-level modeling systems for multi-agent systems can significantly accelerate the process of developing and implementing software for autonomous multi-agent missions. Since different tasks require attention to specific aspects of modeling, an interface-oriented approach can be an effective means of adapting different models of the environment to given behavioral interfaces. Modeling of multi-agent systems covers a wide range of processes – from the physical movement of agents to the formation of behavioral strategies and the organization of their interaction. The interface-oriented approach makes it possible to create flexible multi-agent modeling systems that can perform a wide range of tasks, ensuring the rapid development of behavioral models in a multi-agent environment and SITL testing of low-level autopilot code. As part of the approach, the design of the system begins with the definition of interface interaction between its components, which makes it possible to reuse the code and create individual implementations of components for specific experimental tasks. Work is underway on a prototype of the Blefusku multi-agent modeling system, one of the features of which is an interface-oriented approach and support for high-level behavioral description models. The main modules of the system are a modeling environment that contains a model of the environment and generates sensor signals, and an agent container that is responsible for the behavior of objects and the content of the communication environment. The interface is described as a gRPC service that allows connecting different components written in different programming environments. The communication layer is based on the MAVLink protocol. Behavior is determined by an object that is programmed to respond to sensor data and messages and periodic activities. A fragment of a search and rescue mission scenario with drones is given as an example.

Keywords: agents, interface-oriented approach, modeling, multi-agent system, robotics, sensors, MAVLink, SITL, ROS.

Вступ

В Інституті програмних систем НАН України тривалий час проводяться дослідження, повязані із розробкою та застосуванням систем автоматизації програмування та моделювання [1–4].

Моделювання мультиагентних систем є актуальною темою, що охоплює широкий спектр процесів – від фізики переміщення агентів до вироблення високорівневої стратегії поведінки та форми взаємодії. Кожна ситуація моделювання в чомусь унікальна і потребує від програмних засобів моделювання інфраструктури різних властивостей. Водночас, велика кількість елементів інфраструктури не змінюється. Чи можна побудувати архітектуру, що з одного боку дозволяє перевикористання компонентів, а з іншого боку достатньо гнучку для використання у широкому спектрі завдань?

Історично більшість симуляційних моделей були сконцентровані на відтворенні фізичного середовища і передачі значення сенсорів до програмного забезпечення керування (автопілоту) та передачі значення актуаторів назад у систему симуляції. Така організація взаємодії отримала назву SITL (Software in the Loop) і стала однією з найважливіших технологій тестування. Із розвитком стандартизації робототехнічних систем деяку популярність здобув також стандарт ROS2 [5–7] (Robot Operating System), що включає в себе стандартизовані формати повідомлень для даних сенсорів. Тому типова схема симуляції виглядає як на Рис. 1.

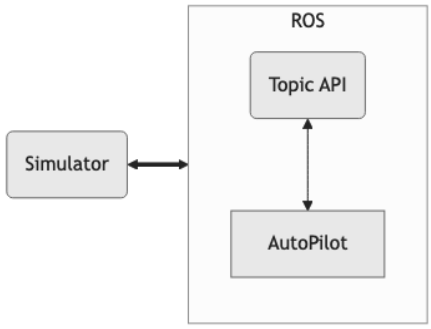


Рис. 1. Типова схема моделювання мультиагентних систем

Як правило, ROS з автопілотом знаходиться в doker-контейнері, запущеному на машині з симуляцією.

Найбільш відомим представником такого типу є Gazebo [8], що де-факто є стандартом у цій сфері, дистрибутиви ROS мають у своєму складі пакети інтеграції з цією системою. Крім того, оскільки для опису динаміки роботів однією з найбільш розповсюджених систем моделювання є Matlab, то MathWorks випустило Matlab ROS Toolbox [9], що дозволяє інтегрувати Matlab моделі з ROS системами.

Подальший розвиток цього напрямку пов'язаний з необхідністю використання більш багатого середовища для поведінки моделей (насамперед пов'язане з розробкою моделей комп'ютерного зору та машинного навчання). В програмуванні є галузь, де однією з головних характеристик розробок є візуалізація віртуальних світів, що містять багато об'єктів, з якими можуть взаємодіяти актори. Це програмування комп'ютерних ігор.

Більше того, в розробці комп'ютерних ігор є такий клас програмного забезпечення, як платформи (або рушії), де однією із задач є моделювання ігрової фізики в режимі м'якого реального часу. Для багатьох задач робототехніки, фізично точне моделювання динаміки може бути замінене приблизною версією, що може виконуватись на ігровій платформі. Знаковим є проєкт AirSim [10] від Microsoft, що використовує для візуалізації Unreal Engine і також підтримував експериментальний плагін для іншої ігрової платформи – Unity. Siemens опублікувала власну бібліотеку інтеграції ROS з Unity [11] та використовувала її у низці індустріальних проєктів. Unity Technology створила Unity Robotic Hub [12] на основі коду Siemens, але згодом припинила її підтримку. Також Unity створила ML-Agent Toolkit [13], що як середовище для тренування нейронних мереж розвивається і зараз.

Якщо переходити до мультиагентних систем, то першою перешкодою, з якою стикаються розробники при еволюції

в цьому напрямку, є швидкодія – паралельне застосування симуляцій не було спроєктоване із самого початку в більшості бібліотек симуляції фізики. LG створили на основі Unity свою версію платформи симуляції – cloisim [14], що використовує той же формат опису світу SDF, що і Gazebo. У Flightmare [15] автори розділили рендеринг та симуляцію і створили API для паралельного генерування даних для сенсорної камери.

Наступна перешкода – складність підтримки. Якщо в нас мультиагентна система, і кожний агент представлений докер-контейнером з автопілотом, як показано на Рис. 2, то конфігурація та підтримка цієї структури стає трудомісткою.

Ще одна річ, про яку слід подбати – це комунікаційний рівень, що відсутній у випадку одного робота. Агенти мають передавати сигнали за допомогою узгоджених протоколів.

В AeroStack [16] пропонується стандартизована ROS-орієнтована архітектура, що включає в себе як SITL симуляцію, так і комунікаційний рівень.

Також уваги потребує опис поведінки агентів, але, якщо система симуляції ґрунтується на моделі SITL, то тоді опис поведінки – це модифікація автопілотів агентів. Однак, автопілоти, як правило, написані низькорівневими мовами програмування, їх модифікувати та дописувати дуже трудомістко. Було б добре мати можливість описувати поведінку за допомогою більш високорівневих моделей.

Мультиагентні системи моделювання, що сфокусовані на абстрактному описі поведінки та не прив'язані до точного моделювання нижнього рівня, представляють собою окремий напрямок. З найбільш відомих систем можна вказати JADE [17, 18] (Java Agent Based Environment), де алгоритм роботи агента представляється Java-класом, що може виконувати запити до середовища та взаємодіяти з іншими такими ж об'єктами і обмінюватись інформацією між ними, використовуючи FIPA-сумісні протоколи. Ще один підхід – це представлення поведінки не як програми, а як функції вибору поведінки з певного набору елементарних елементів поведінки у часі. Такий підхід цікавий тим, що до розробки поведінки, представленої у такому вигляді, можна застосувати методи машинного навчання, такі як навчання з підкріпленням [19]. В CAMAS [20] недетерміністична поведінка сукупності роботів формально описується за допомогою марковських ланцюгів і моделюється за допомогою швидкодіючого симулятора дискретних подій.

1. Інтерфейсно-орієнтований підхід

Чи можна побудувати мультиагентну систему моделювання, призначену для виконання широкого спектру задач, і з одного боку мати можливість швидкої розробки моделей поведінки у мультиагентному середовищі, а з іншого – мати можливість SITL-тестування низькорівневого коду автопілота? Це дасть можливість швидкої розробки і валідації множини моделей поведінки для вибору однієї з них, яку можна буде реалізувати у низькорівне-

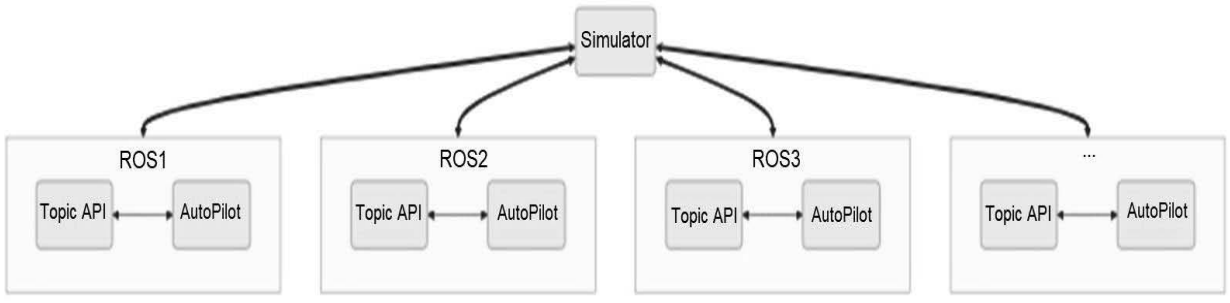


Рис. 2. Мультиагентна система, де кожний агент представлений докер-контейнером з автопілотом

вому автопілоті. Зараз ведуться роботи над прототипом системи мультиагентного моделювання, що має назву Blefusku. Однією з її особливостей є інтерфейсно-орієнтований підхід і підтримка високорівневих моделей опису поведінки.

Інтерфейсно-орієнтований підхід полягає в тому, що опис інтерфейсної взаємодії між різними частинами системи є першим артефактом, з якого починається проєктування системи. Маючи різні реалізації компонент для різних задач, ми можемо водночас і перевикористовувати код, і розробляти індивідуальний варіант функції під експеримент.

В найбільш загальному вигляді спрощену структуру системи можна зобразити, як показано на Рис. 3 – в нас є певне середовище, що включає в себе функціонал фізичних розрахунків, об'єктів та полів, у яких відбувається комунікація.

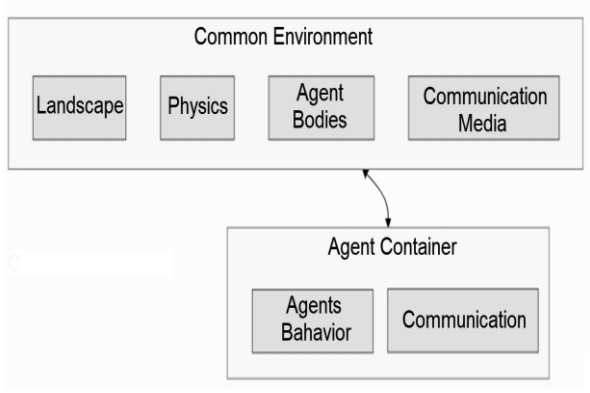


Рис. 3. Основні компоненти симуляції

З іншого боку, є контейнер агентів, що відповідає за поведінку об'єктів та змістовну частину комунікаційного середовища. Основний інтерфейс – це взаємодія між середовищем та контейнером агентів. Інтерфейс описаний як сервіс gRPC, що дозволяє з'єднувати різні компоненти, написані у різних середовищах програмування. Наприклад, для середовища є реалізації API мовами Rust, C# (для Unity), та планується підключення C++ інтерфейсу Unreal Engine.

Функціональність агента можна описати як функцію реакції, що отримує на вхід значення сенсорів та вхідні повідомлення і повертає значення актуаторів та вихідні повідомлення. На protobuf це можна представити як вхідні та вихідні повідомлення:

```
message ActorBehaviorInput {
  map<string, SensorData>
    sensor_data = 1;
  map<string, ActuatorReply>
    actuator_replies = 2;
  Messages input_messages = 3;
}

message ActorBehaviorOutput {
  map<string, ActuatorCommand>
    actuator_commands = 1;
  Messages output_messages = 2;
}
```

Структура SensorData – це дані сенсорів, описані максимально близько до відповідних структур в ROS2:

```
message SensorData {
  oneof data {
    Image image = 1;
    NavSatInfo navsat = 2;
    ImuInfo imu = 3;
    …
  }
}
```

Аналогічно структура ActuatorData – це типова структура вводу актуатора. Репліки з актуаторів надходять до агента у наступному циклі управління. Можна уявити роботу агента як функцію

$$f: ActorBehaviorInput \times State \rightarrow State \times ActorBehaviorOutput,$$

де *State* – внутрішній стан агента, структура якого описана у поведінці.

Комунікаційний рівень ґрунтується на протоколі MAVLink – вважається, що всі агенти об'єднані у спільну MAVLink-мережу. Поряд зі стандартними повідомленнями MAVLink також підтримуються і додаткові повідомлення, визначені користувачем – розроблено транслятор, який приймає на вхід XML-опис MAVLink повідомлення і додає його до protobuf струк-

тури, що об'єднує всі типи можливих повідомлень.

Контейнер агентів містить бібліотеку можливих поведінок. Поведінка визначається об'єктом, у якому запрограмована реакція на дані з сенсорів, та повідомлення і періодичні активності. Ще один компонент, SITL-проксі, може використовуватись для підключення реального автопілоту у режимі SITL-тестування, де ми можемо його запускати в одному середовищі з чисто програмними агентами. Структура такого підключення зображена на Рис. 4.

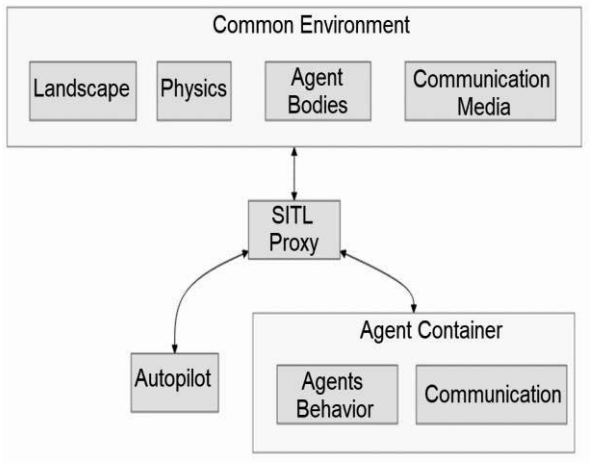


Рис. 4. Симуляція, сумісна з SITL

2. Програмний опис поведінки

Високорівневий опис поведінки задається за допомогою опису класу мовою Python. Екземпляр такого класу створюється для кожного актора і при ініціалізації отримує стан та контекст актора, де визначені методи взаємодії з сенсорами та актуаторами.

Програміст може описати методи реакції на події в об'єкті поведінки та пов'язати їх із повідомленнями за допомогою декораторів.

Як приклад наведемо фрагмент сценарію пошуково-рятувальної місії, у якій беруть участь N дронів. Нехай у нас є певна сфера відповідальності, за сигналом початку місії дрони розподіляють між собою поля пошуку і виконують обліт території за визначеним маршрутом. Водночас камери відслідковують об'єкт пошуку і, знаходячи схоже зображення, надсилають його для підтвердження на наземну станцію. Якщо підтвердження отримане, дрон спускається поряд з об'єктом пошуку та відкриває маніпулятор, а інші дрони завершують місію.

Почнемо з того, що додамо до стандартних повідомлень MAVLink нові повідомлення, що стосуються рятувальної місії. Це може бути п'ять повідомлень:

– RESQUE_MISSION_START – генерується на старті місії;

– RESQUE_TARGET_CANDIDATE (з полями, що включають координати та MFTP URL та унікальним ID) – генеруються, коли камера знайшла кандидата для цілі пошуку;

– RESQUE_TARGET_CANDIDATE_APPROVE – генеруються, коли з наземної станції прийшло підтвердження, що пошук успішний;

– RESQUE_TARGET_CANDIDATE_REJECT – генерується, коли оператор позначив результати пошуку як несправжнє розпізнавання;

– RESQUE_TARGET_REACHED – коли один дрон почав операцію порятунку, а інші можуть закінчити пошук.

Після цього потрібно перегенерувати класи повідомлень та імплементувати поведінку.

Поведінка визначається класом, у якому визначається стан і вказуються реакції на повідомлення (Рис. 5).

Тобто тут ми бачимо, що під час старту пошукової місії, якщо дрон не зайнятий, викликається метод start_search, що ініціює старт місії. Відповідне MAVLink-повідомлення передається на цей же автопілот (і буде відпрацьоване як частина стандартної поведінки).

У режимі пошуку періодично сканується зображення з камери, і якщо знайдений кандидат, то місія призупиняється і на наземну станцію передається зображення.

Отримавши підтвердження того, що пошук вдалий, дрон переходить у режим посадки біля цілі, та посилає броадкаст-сигнал, що пошукову ціль знайдено (Рис. 6).

Усі інші дрони бачать це повідомлення і призупиняють роботу, якщо немає іншої активної пошукової місії.

```
class ResqueMission(BaseBehavior):

   state: 'IDLE' | 'SEARCH' | 'APPROVE' | 'RESCUE' | 'RETURN' | 'LAND'

   @on_message(RESQUE_MISSION_START, when='IDLE')
   async def  start_search(self, message: mavlink_pb2.SEARCH_AREA):
       self.state = 'SEARCH'
       await self.ctx.send_mavlink(my_system_id, 1,
MAV_CMD_DO_SET_MISSION_CURRENT(1,1) )
       self._context.log_info("ResqueMission: start")

   @periodic(duraction = 1, when='SEARCH')
   async def classify_resque_target(self):
       image = await self.ctx.camera_image()
       found = check_for_person(image)
       if found:
           self.state = 'APPROVE'
           await self.ctx.send_mavlink(my_system_id, 1, MAV_CMD_CONDITION_DELAY, 1)
           image_uri = await ctx.upload_image(image)
           self.ctx.send_mavlink(GROUND_STATION, 1, RESQUE_CANDIDATE_FOUND,
...ctx.pos, image_uri)
```

Рис. 5. Клас, що задає поведінку

```
   @on_message(RESQUE_CANDIDATE_APPROVED, when='APPROVE')
   async def  resque_candidate_approved(self, message:
   mavlink_pb2.CAMERA_IMAGE_CAPTURED):
       self._context.log_info("ResqueMission: resque_approved")
       self.state = 'RESCUE'
       await self.ctx.send_mavlink(my_system_id, 1, SET_POSITION_TARGET_LOCAL_NED,
          ...correct_resque_target(ctx.pos))
       await self.ctx.send_mavlink(my_system_id, 1, MAV_CMD_DO_RALLY_LAND)
       self.ctx.send_mavlink(BROADCAST, 1, RESQUE_TARGET_REACHED, ctx.pos)
```

Рис. 6. Після підтвердження успішного пошуку дрон переходить у режим посадки біля цілі та надсилає широкомовний сигнал про її виявлення

Зазначимо, що один агент може описуватись кількома скриптами поведінки, які можуть описувати різні компоненти однієї системи. Так, наприклад, у розробці систем машинного зору, як правило, окремо моделюється камера.

Висновки

Високорівневі системи моделювання мультиагентних систем відіграють ключову роль у прискоренні циклу розробки та впровадження програмного забезпечення для автономних мультиагентних місій. Вони дозволяють створювати складні симуляційні середовища, які можуть точно відтворювати як фізичні умови, так і

поведінкові аспекти агентів, що взаємодіють між собою. Це, у свою чергу, сприяє ефективному тестуванню, налагодженню та вдосконаленню алгоритмів управління автономними системами без необхідності безпосереднього використання реального обладнання на ранніх етапах розробки.

Оскільки різні завдання вимагають акцента на різних аспектах моделювання, гнучкість симуляційних платформ стає критично важливою. Деякі сценарії потребують точного моделювання фізичних законів, таких як аеродинаміка або динаміка руху транспортних засобів, тоді як інші зосереджуються на комунікаційних аспектах або ухваленні рішень агентами в складному середовищі.

У цьому контексті інтерфейсно-орієнтований підхід до розробки мультиагентних систем може значно полегшити адаптацію різних середовищ моделювання до специфічних інтерфейсів поведінки агентів. Такий підхід передбачає чітке розділення між моделями середовища та алгоритмами поведінки, що дозволяє легко змінювати або вдосконалювати окремі компоненти системи без необхідності повного перероблення всієї інфраструктури. Крім того, це сприяє масштабованості та модульності симуляційних платформ, що робить їх придатними для широкого спектру застосувань – від дослідження стратегій кооперації роботів до тестування автономного транспорту та безпілотних літальних апаратів.

Завдяки такій організації можна не лише скоротити час розробки програмного забезпечення, а й покращити його якість, забезпечивши ефективне тестування віртуальних агентів у різних умовах ще до їхнього розгортання у реальному світі.

References

1. P. Andon, A. Doroshenko, V. Akylovsky, P. Ivanenko, O. Yatsenko, Formal and adaptive methods of constructing high-performance parallel programs. Kyiv: Naukova Dumka, 2023. [in Ukrainian]
2. A. Doroshenko, O. Yatsenko, Formal and adaptive methods for automation of parallel programs construction: emerging research and opportunities. Hershey: IGI Global, 2021. doi: 10.4018/978-1-5225-9384-3
3. Rahozin D. V., Doroshenko A. Yu. Modelling videocard memory performance for LLM neural networks, in: Problems in programming 2–3 (2024) 37–44. doi: 10.15407/pp2024.02-03.037 [in Ukrainian]
4. Doroshenko A. Yu., Okonsky I. V., Zhereb K. A., Beketov O. G. Using modeling tools to determine optimal parameters of program execution on video graphics accelerators, in: Problems in programming 2 (2013) 23–31. [in Ukrainian]
5. M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, A. Y. Ng, ROS: an open-source Robot Operating System, in: Open-Source Software workshop of the International Conference on Robotics and Automation (ICRA) (2009) 1–6. Accessed: 04.03.2025. http://www.robotics.stanford.edu/~ang/papers/icraoss09-ROS.pdf
6. S. Macenski, T. Foote, B. Gerkey, C. Lalancette, W. Woodall, Robot Operating System 2: design, architecture, and uses in the wild, in: Science Robotics (2022) 7. doi: 10.1126/scirobotics.abm6074
7. ROS. Accessed: 04.03.2025. https://ros.org/
8. N. Koenig, A. Howard, Design and use paradigms for Gazebo, an open-source multi-robot simulator, in: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566) (2004), vol. 3, 2149–2154, doi: 10.1109/IROS.2004.1389727.
9. Matlab ROS toolbox. Accessed: 04.03.2025. https://www.mathworks.com/products/ros.html
10. S. Shah, D. Dey, C. Lovett, A. Kapoor, AirSim: high-fidelity visual and physical simulation for autonomous vehicles, in: ArXiv (2017) doi: 10.48550/arXiv.1705.05065
11. ros-sharp. Accessed: 04.03.2025. https://github.com/siemens/ros-sharp/
12. Unity-Robotics-Hub. Accessed: 04.03.2025. https://github.com/Unity-Technologies/Unity-Robotics-Hub/
13. J. Arthur, V.–P. Berges, E. Teng, A. Cohen, J. Harper, C. Elion, C. Goy, Y. Gao, H. Henry, M. Mattar, D. Lange, in: arXiv preprint arXiv:1809.02627 (2020) doi: 10.48550/arXiv.1809.02627
14. CLOiSim: Multi-Robot Simulator. Accessed: 04.03.2025. https://github.com/lge-ros2/cloisim

15. Y. Song, S. Naji, E. Kaufmann, A. Loquercio, D. Scaramuzza, Flightmare: a flexible quadrotor simulator, in: Conference on Robot Learning (CoRL) (2020) 1–11. Accessed: 04.03.2025. https://rpg.ifi.uzh.ch/docs/CoRL20_Yunlong.pdf
16. M. Fernandez-Cortizas, M. Molina, P. Arias-Perez, R. Perez-Segui, D. Perez-Saura, P. Campoy, Aerostack2: a software framework for developing multi-robot aerial systems, in: ArXiv (2023). doi: 10.48550/arXiv.2303.18237.
17. F. L. Bellifemine, G. Caire, D. Greenwood, Developing multi-agent systems with JADE (Wiley Series in Agent Technology), John Wiley & Sons, 2007.
18. Jade. Accessed: 04.03.2025. https://jade.tilab.com/
19. P. Pierpaoli, T. T. Doan, J. Romberg, M. Egerstedt, Sequencing of multi-robot behaviors using reinforcement learning, in: Control Theory and Technology 19 (2021) 529–537. doi: 10.1007/s11768-021-00069-5
20. C. Street, B. Lacerda, M. Staniaszek, M. Mühlig, N. Hawes, Context-aware modelling for multi-robot systems under uncertainty, in: 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS'22). International Foundation for Autonomous Agents and Multiagent Systems (2022) 1228–1236. Accessed: 04.03.2025. https://www.ifaamas.org/Proceedings/aamas2022/pdfs/p1228.pdf

Одержано: 05.03.2025
Внутрішня рецензія отримана: 11.03.2025
Зовнішня рецензія отримана: 10.03.2025

Про авторів:

[1]Шевченко Руслан Сергійович,
кандидат технічних наук,
старший науковий співробітник.
https://orcid.org/0000-0002-1554-2019.

[1,2]Дорошенко Анатолій Юхимович,
доктор фізико-математичних наук,
професор, провідний науковий співробітник.
http://orcid.org/0000-0002-8435-1451.

[1,2]Лесик Валентин Олександрович,
інженер,
аспірант КПІ ім. Сікорського.
http://orcid.org/0000-0002-8307-5707.

[2]Савчук Олена Володимирівна,
кандидат технічних наук,
доцент кафедри інформаційних систем та технологій.
http://orcid.org/0000-0003-3176-7952.

[1]Яценко Олена Анатоліївна,
кандидат фізико-математичних наук,
старший науковий співробітник.
http://orcid.org/0000-0002-4700-6704.

Місце роботи авторів:

[1] Інститут програмних систем
НАН України,
тел. +38-067-407-32-33
E-mail: doroshenkoanatoliy2@gmail.com,
ruslan@shevchenko.kiev.ua,
oayat@ukr.net
Сайт: https://iss.nas.gov.ua

[2] Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»,
Сайт: https://ist.kpi.ua

УДК 004.9, 621.3, 539.144, 621.039 https://doi.org/10.15407/pp2025.01.118

Б.В. Лащонов, І.П. Сініцин

ПРОГРАМНО-АПАРАТНА СИСТЕМА БЕЗКОНТАКТНОГО ВИЯВЛЕННЯ МІН НА ОСНОВІ НЕПРУЖНОГО РОЗСІЮВАННЯ НЕЙТРОНІВ ТА МАШИННОЇ ОБРОБКИ СПЕКТРІВ ХАРАКТЕРИСТИЧНОГО γ-ВИПРОМІНЮВАННЯ

Запропоновано програмно-апаратну систему дистанційного виявлення мін з використанням методу нейтронного неруйнівного аналізу. Метод заснований на аналізі взаємодії швидких нейтронів з ядрами азоту, вуглецю та кисню у складі вибухових речовин і машинному обробленні спектрів характеристичного γ-випромінювання, що виникає в результаті непружного розсіювання. Проведено моделювання γ-спектрів для типових компонентів мін, розглянуто можливості реалізації методу в польових умовах, а також надано рекомендації щодо вибору джерел нейтронів і методик розрахунку спектрів з урахуванням спотворювальних факторів.
Ключові слова: нейтронне розсіювання, γ-спектроскопія, розмінування, вибухові речовини, азот, неруйнівний аналіз, дистанційне виявлення мін, машинне навчання, нейронні мережі, штучний інтелект.

B.V. Lashchonov, I.P. Sinitsyn

SOFTWARE-HARDWARE SYSTEM FOR CONTACTLESS MINE DETECTION BASED ON INELIGIBLE NEUTRON SCATTERING AND MACHINE PROCESSING OF CHARACTERISTIC γ-RADIATION SPECTRA

This paper proposes a software-hardware system for remote mine detection using the neutron nondestructive analysis method. The method is based on the analysis of the interaction of fast neutrons with nitrogen, carbon, and oxygen nuclei in explosives and computer processing of the spectra of characteristic γ-radiation resulting from inelastic scattering. The γ-spectra are simulated for typical mine components, the possibilities of implementing the method in field conditions are considered, and recommendations are given for the selection of neutron sources and methods for calculating spectra, taking into account distorting factors.
Keywords: neutron scattering, γ-spectroscopy, demining, explosives, nitrogen, non-destructive analysis, remote mine detection, machine learning, neural networks, artificial intelligence.

Вступ

Розмінування залишається гострою гуманітарною проблемою. Традиційні методи розмінування передбачають контакт із вибухонебезпечними об'єктами, що пов'язано з високим ризиком. Сучасні фізичні методи, включаючи застосування різних типів іонізуючого випромінювання, дозволяють успішно розвивати безконтактні підходи до пошуку мін. Одним із найперспективніших методів є нейтронний аналіз складу речовини з реєстрацією характеристичного γ-випромінювання – метод миттєвого нейтронно-активаційного аналізу (prompt gamma neutron activation analysis, PGNAA) [1–9]. Однак використання цього або інших відомих методів пошуку мін за допомогою дронів, а також застосування нейронних мереж для аналізу спектрів випромінювання на даний момент видається дуже перспективним.

Для кращого розуміння можливості реалізації цього завдання розглянемо всі компоненти, необхідні для його вирішення: джерела нейтронів (табл. 1), хімічний склад вибухових речовин (ВР) (табл. 2), приклади спектрів γ-випромінювання, методики обробки результатів.

Опис запропонованої галузі автоматизації

Спочатку розглянемо існуючі джерела нейтронів та їхні характеристики.

Енергетичний спектр нейтронів від різних джерел зображено на рис.1 [6], де:

1 – ядерний реактор;

2 – циклотрон з пучком дейтронів енергії 40 МеВ;

3 – 241Am/Be-джерело;

4 – 4-14 МеВ нейтронний генератор.

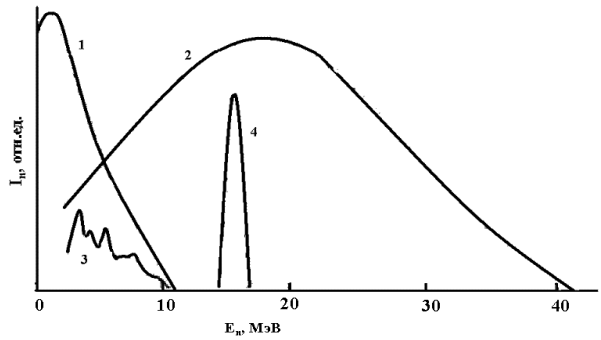


Рис. 1. Енергетичний спектр нейтронів

Таблиця 1

Джерела нейтронів

Тип джерела	Розміри (мм)	Вага (кг)	Вихід нейтронів (нейтронів/сек)	Примітки
Am-Be (AMN.PE4)	Ø30.1 × 60.2	~0.1	~2×10^6 на Ci	Постійне джерело
Thermo Fisher P-320	Ø190 × 440	~9	до 1×10^8	Компактний, портативний
Starfire nGen™-310	Ø70 × 480	~7	до 1×10^8	Ультракомпактний, інтегрований
Коаксіальний генератор	Ø280 × 260	~18	до 1.2×10^{12} (D-D) / 3.5×10^{14} (D-T)	Висока продуктивність

Таблиця 2

Хімічний склад вибухових речовин

Назва	Формула	Хімічна назва
HMX	$C_4H_8N_8O_8$	Octahydro-1,3,5,7-tetranitro-1,3,5,7-tetrazocine
LX-17	92.5% TATB, 7.5% Kel-F 800 $(C_8H_2Cl_3F_{11})_n$	LX-17-0
TNT	$C_7H_5N_3O_6$	2-methyl-1,3,5-trinitrobenzene
Composition B	63% RDX ($C_3H_6N_6O_6$), 36% TNT, 1% wax	--
TATB	$C_6H_6N_6O_6$	2,4,6-trinitro-1,3,5-benzenetriamine
PBX-9501	95% HMX, 2.5% Estane $(C_{5.14}H_{7.50}N_{0.19}O_{1.76})_n$	
2.5% BDNPA-F	--	
PBX-9502	95% TATB, 5% Kel-F 800	--
NM	CH_3NO_2	Nitromethane
ANFO	95% Ammonium Nitrate ($H_4N_2O_3$), 5% fuel oil	Ammonium nitrate-fuel oil mixture
Black Powder	75% KNO_3, 15% charcoal, 10% sulfur	--
TATP	$C_9H_{18}O_6$	Triacetonetriperoxide

Процеси, що відбуваються під час взаємодії потоку нейтронів з вибуховою речовиною

Процеси, що відбуваються у разі опромінення будь-якого матеріалу потоком нейтронів високих енергій, а також ядерні реакції (n, n), (n, p), (n, α) або (n, 2n) досить добре вивчені в широкому діапазоні енергій нейтронів і досліджуваних матеріалів – N (азот), O (кисень). Для вирішення вказаної задачі мають значення γ-спектри. Зі збільшенням енергії налітаючих нейтронів (рис. 2) на кілька порядків буде менше переріз реакції (n, γ) у діапазоні енергії доступних нам джерел нейтронів.

Тому зупинимося на розгляді реакції непружного розсіювання (n, nγ) на хімічних елементах, що входять до складу ВР. Ця реакція дає спектри, котрі мають декілька характерних піків (рис.3):

1. $n + {}^{14}N \rightarrow {}^{14}N^{*} \rightarrow {}^{14}N + \gamma$ (2.31 МеВ)

Енергія збудження: 2.31 МеВ.

Рівень добре видно у процесі опромінення швидкими нейтронами.

Є ключовим маркером наявності азоту – основний елемент RDX, TNT, ANFO та ін.

2. $n + {}^{12}C \rightarrow {}^{12}C^{*} \rightarrow {}^{12}C + \gamma$ (4.44 МеВ)

Дуже характерна γ-лінія.

Вуглець – основа всіх органічних речовин, включаючи корпус міни та ВР.

3. $n + {}^{16}O \rightarrow {}^{16}O^{*} \rightarrow {}^{16}O + \gamma$ (6.13 МеВ)

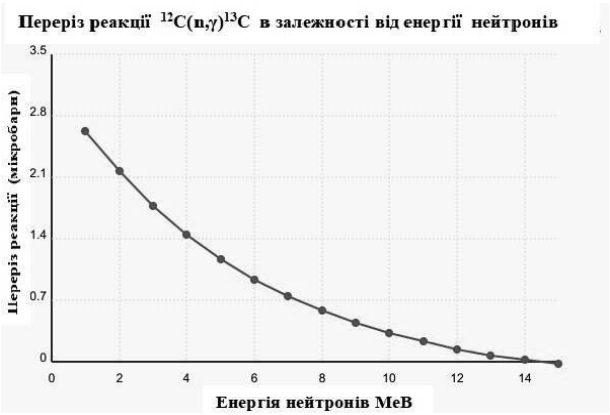


Рис. 2. Характерний переріз реакції (n,γ) [6]

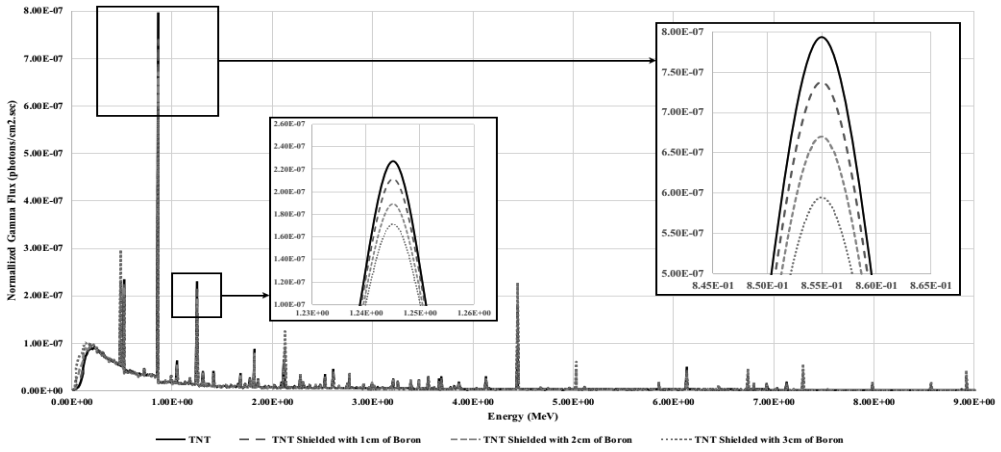


Figure 1. Sample of PGNAA obtained gamma spectra for TNT when shielded with boron.

Рис. 3. Характерний спектр реакції (n, nγ), отриманий на зразку ТНТ

Алгоритм та обчислювальні методи

Як видно з розглянутих раніше матеріалів, в ідеальному випадку сама фізика процесу взаємодії нейтронів з ВР дає можливість виявляти ці речовини за характерними піками, але у реальних польових умовах виникає ціла низка труднощів у вирішенні цього завдання.

Оскільки міни можуть мати зовнішню оболонку, а також знаходитися в шарі ґрунту, то реальні спектри будуть спотворені γ-випромінюванням від сторонніх хімічних речовин. Тому останнім часом для аналізу таких спектрів почали використо-

вувати можливості штучного інтелекту (AI) [9], зокрема глибоку нейронну мережу (deep neural network), яку тренували на спектрах, змодельованих з використанням методу Monte Carlo (MCNP Code) [7]. Вдалося отримати 95%-у точність визначення відомої ВР при попередньому навчанні моделі AI на наборі відомих зразків ВР, а також 80%-у точність визначення невідомої ВР.

У роботі [8] розраховували співвідношення N/C і N/O у спектрах γ-випромінювання, отриманих методом MCNP. Ці співвідношення є хорошими індикаторами наявності ВР. Подальше порівняння цих розрахунків із вимірюваннями, проведеними на реальних зразках, підтвердило хороший збіг (рис. 4).

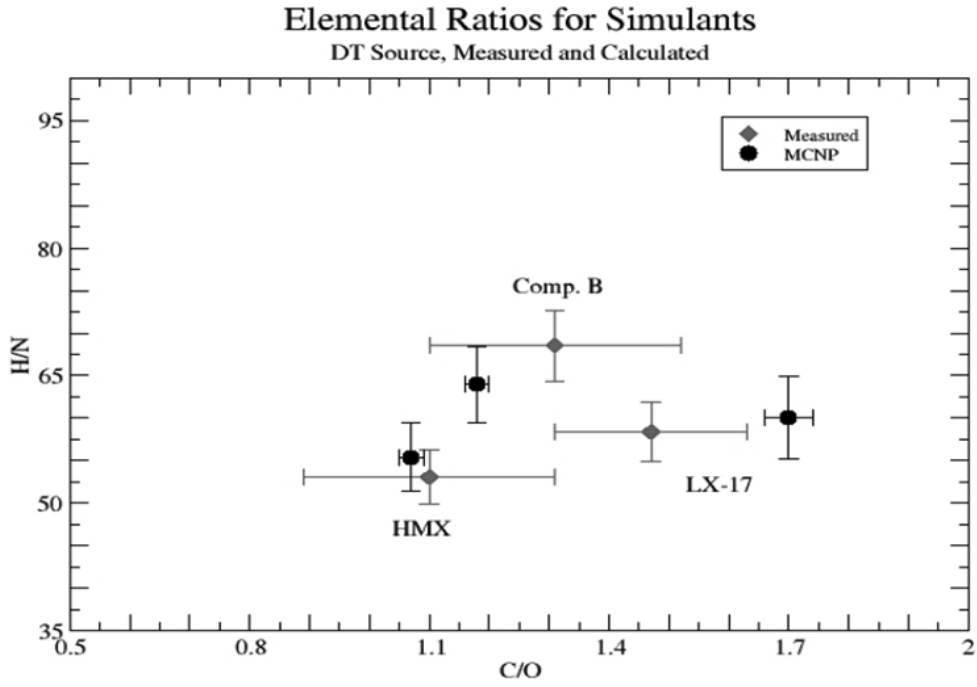


Рис. 4. Порівняння розрахованого та виміряного співвідношень N/C та N/O у ВР

Алгоритм обліку втрат у повітрі

Для створення програмно-апаратної системи (ПАС) виявлення ВР з використанням дронів, додатковим ускладнюючим фактором є наявність шару повітря між джерелом нейтронів і землею. Оскільки цей фактор у випадку з дроном буде присутнім завжди, то є сенс його оцінити, розрахувати і розрахункові значення (як функцію відстані від джерела до землі) вводити як поправку ще на етапі реєстрації спектрів.

Розглянемо основні види взаємодії швидких нейтронів у повітрі:

- Пружне розсіювання (в основному на N та O);
- Непружне розсіювання (генерація γ-квантів від повітря);
- Захоплення нейтронів (незначно для швидких нейтронів);
- Незначне уповільнення (в тому числі внаслідок розсіювання).

Для оцінки ослаблення потоку нейтронів при проходженні через 100 см повітря, використовуємо приблизно експоненційне згасання:

$$\Phi = \Phi_0 \times e^{-\Sigma_t x},$$

де Φ_0 – 1×10^8 n/сек початковий потік,

Φ – потік після проходження шару повітря;

Σ_t – макроскопічний коефіцієнт ослаблення (total macroscopic cross-section);

x = 100 см =1 м – товщина шару повітря.

Для швидких нейтронів (14 МеВ) у повітрі значення повного перерізу взаємодії σ_t буде невеликим — близько 0.6 барн

на атом у середньому за складом повітря ($N_2 \approx 78\%$, $O_2 \approx 21\%$).

Макроскопічний переріз:

$$\Sigma_t = N * \sigma_t ,$$

де N – кількість ядер в 1 см³ повітря, тобто

$$N = \rho * N_A \approx 2.7 * 10^{19} \text{ ядер/см}^3.$$

Для повітря $\Sigma_t \approx 0.003–0.005$ см$^{-1}$ при 14 МеВ. Візьмемо $\Sigma_t \approx 0.004$ см$^{-1}$.

$$\Phi = 1x\ 10^8 * e^{-\Sigma_t x} = 1x\ 10^8 * e^{-0.4} = 6.7*10^7 \text{ n/сек.}$$

Разом:
Початковий потік нейтронів 10^8 n/сек після проходження 1 метра повітря зменшиться до $6.7*10^7$ нейтронів/сек, що видається цілком прийнятним для використання методу (PGNAA) у вирішенні поставленого завдання.

Труднощі реалізації програмно-апаратної системи

Дана робота присвячена не тільки огляду існуючих методів пошуку та виявлення ВР, але й опису етапів реалізації ПАС, що включає вже наявні напрацювання і передбачувані програмні рішення для їх комплексного використання в реальних польових умовах із застосуванням дронів або, як мінімум, роботизованих компактних наземних пристроїв.

У зв'язку з цим слід зазначити низку додаткових труднощів та передбачувані шляхи їх вирішення задля створення цієї ПАС.

Труднощі	Рішення
Високий γ-фон	Комплексний аналіз (N/C/O співвідношення, форма сигналів, часові залежності, колімація) Застосування нейромереж в аналітиці спектрів
Невелика кількість речовини	Навчання моделей на симульованих даних (MCNP)
Радіаційна безпека	Використання нейтронних генераторів, імпульсний режим

Результати

Проведений аналіз робіт [1–9] і деякі попередні розрахунки показують, що існуючі методики розрахунків і сучасна апаратура дозволяють:

- з 95%-ю точністю знаходити відомі за складом ВР (див. табл. 1).
- з 80%-ю точністю знаходити невідомі за складом ВР за допомогою виділення характерних піків ^{14}N, ^{12}C та ^{16}O.

Здійснено оцінювання та представлено алгоритм розрахунку додаткових факторів, що впливають на отримання кінцевого результату розв'язання задачі пошуку мін та інших ВР.

Огляд вже існуючих методів дистанційного виявлення мін і наявного на даний момент обладнання дає змогу сподіватися на реалізацію програмно-апаратної системи, яка забезпечить вирішення цього завдання із застосуванням дронів або, як мінімум, роботизованих наземних комплексів. На підставі зазначеного раніше можна запропонувати такі етапи реалізації ПАС.

Етапи реалізації ПАС

1. Використання компактних нейтронних генераторів (див. табл. 1).
2. Розрахунок (за допомогою MCNP) найбільш ефективної геометрії установки.
3. Симулювання рядів даних за допомогою MCNP для різних конфігурацій та ефективностей детекторів, складів ВР, фонового γ-випромінювання.
4. Навчання моделей на симульованих даних.
5. Розшифровування отриманих спектрів за допомогою навчених моделей.
6. Визначення місця знаходження мін та інших ВР.

Література

1. Csikai J. Handbook of Fast Neutron Generators and Applications. Boca Raton: CRC Press, 2021. 300 p.
2. Chichester D. L., McMurdo J., Priestley K. Fast neutron interrogation of explosives. Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms. 2007. Vol. 261. P. 838–841.

3. IAEA. Humanitarian Demining Using Nuclear Techniques. Vienna: International Atomic Energy Agency, 2002. 150 p.
4. Eleon C., Perot B., Carasco C. Gamma ray spectroscopy with LaBr3 detectors for explosive detection. *IEEE Transactions on Nuclear Science*. 2011. Vol. 58, no. 5. P. 2310–2316.
5. Majewski S., Gozani T., Gottesman S. Neutron-based detection of landmines. *Applied Radiation and Isotopes*. 2005. Vol. 63, no. 5-6. P. 669–674.
6. Азаренков Н. А., Кириченко В. Г., Левенець В. В., Неклюдов І.М. Ядерно-фізичні методи в матеріалознавстві: навч. посіб. Харків: ХНУ імені В.Н. Каразіна, 2011. 300 с.
7. X-5 Monte Carlo Team. MCNP-Version 5, Vol. I. Overview and Theory. Los Alamos: Los Alamos National Laboratory, 2003. (LA-UR-03-1987).
8. Hossny K., Hany A. Explosives Detection and Identification by PGNAA. Idaho Falls: Idaho National Laboratory, 2006. (INL/EXT-06-01210).
9. Hossny K., Magdi S., Soliman A. Y., Hossny M. Detecting shielded explosives by coupling prompt gamma neutron activation analysis and deep neural networks. Scientific Reports. 2020. Vol. 10, article 13467.

References

1. Csikai J. Handbook of Fast Neutron Generators and Applications. Boca Raton: CRC Press, 2021. 300 p.
2. Chichester D. L., McMurdo J., Priestley K. Fast neutron interrogation of explosives. Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms. 2007. Vol. 261. P. 838–841.
3. IAEA. Humanitarian Demining Using Nuclear Techniques. Vienna: International Atomic Energy Agency, 2002. 150 p.
4. Eleon C., Perot B., Carasco C. Gamma ray spectroscopy with LaBr3 detectors for explosive detection. *IEEE Transactions on Nuclear Science*. 2011. Vol. 58, no. 5. P. 2310–2316.
5. Majewski S., Gozani T., Gottesman S. Neutron-based detection of landmines. *Applied Radiation and Isotopes*. 2005. Vol. 63, no. 5-6. P. 669–674.
6. Azarenkov N. A., Kirichenko V. G., Levenets V. V., Neklyudov I. M. Nuclear physics methods in materials science: a textbook. Kharkiv: V. N. Karazin Kharkiv National University, 2011. 300 p. [in Ukrainian]
7. X-5 Monte Carlo Team. MCNP-Version 5, Vol. I. Overview and Theory. Los Alamos: Los Alamos National Laboratory, 2003. (LA-UR-03-1987).
8. Hossny K., Hany A. Explosives Detection and Identification by PGNAA. Idaho Falls: Idaho National Laboratory, 2006. (INL/EXT-06-01210).
9. Hossny K., Magdi S., Soliman A. Y., Hossny M. Detecting shielded explosives by coupling prompt gamma neutron activation analysis and deep neural networks. Scientific Reports. 2020. Vol. 10, article 13467.

Одержано: 12.02.2025
Внутрішня рецензія отримана: 19.02.2025
Зовнішня рецензія отримана: 08.03.2025

Про авторів:

Лащонов Борис Вікторович,
кандидат фізико-математичних наук,
https://orcid.org/0009-0006-2055-2058.

Сініцин Ігор Петрович,
доктор технічних наук, професор,
член-кореспондент НАН України,
https://orcid.org/0000-0002-4120-0784

Місце роботи авторів:

Інститут програмних систем НАН України,
03187, м. Київ-187,
проспект Академіка Глушкова, 40.
Тел.: (044) 526 3559.
E-mail: ips2014@ukr.net

УДК 004.9:378.4:37.018.43 https://doi.org/10.15407/pp2025/01/124

П.І. Біда, С.В. Надозірний, В.В. Кот

ПІДВИЩЕННЯ ЯКОСТІ УПРАВЛІННЯ ОСВІТНІМ ПРОЦЕСОМ ЧЕРЕЗ ІНТЕГРАЦІЮ МОДУЛЯ НА ОСНОВІ ERP ODOO В КОНТЕКСТІ ХМАРНИХ ТЕХНОЛОГІЙ

У статті розглядається інтеграція модуля управління освітнім процесом на основі ERP-системи Odoo у контексті використання хмарних технологій. Описується важливість автоматизації та оптимізації управлінських процесів в освітніх установах. Система ERP Odoo представлена як універсальний інструмент для централізованого управління розкладами, обліком успішності та комунікацією між усіма учасниками освітнього процесу. Окрему увагу приділено адаптації системи до потреб українських навчальних закладів, створенню індивідуальних рішень, а також тестуванню модуля на базі реального освітнього закладу. Результати дослідження демонструють ефективність інтеграції системи, зокрема автоматизацію рутинних завдань, підвищення прозорості та продуктивності роботи закладів освіти.
Ключові слова: ERP Odoo, освітній процес, хмарні технології, автоматизація, цифровізація освіти, оптимізація управління.

P.I. Bida, S.V. Nadozirnyi, V.V. Kot

IMPROVING THE QUALITY OF EDUCATIONAL PROCESS MANAGEMENT THROUGH THE INTEGRATION OF A MODULE BASED ON ERP ODOO IN THE CONTEXT OF CLOUD TECHNOLOGIES

The article examines the integration of an education management module based on the Odoo ERP system in the context of cloud technology usage. It highlights the importance of automation and optimization of management processes in educational institutions. The Odoo ERP system is presented as a universal tool for centralized management of schedules, academic performance tracking, and communication among all participants in the educational process. Special attention is given to adapting the system to the needs of Ukrainian educational institutions, developing customized solutions, and testing the module in a real educational environment. The research results demonstrate the effectiveness of system integration, particularly in automating routine tasks, enhancing transparency, and improving the productivity of educational institutions.

Вступ

Актуальність даного дослідження зумовлена необхідністю впровадження інноваційних рішень для оптимізації управлінських процесів в умовах сучасного розвитку технологій. Освітня сфера, яка завжди була важливою складовою суспільного розвитку, нині потребує переосмислення традиційних підходів до управління та організації. Впровадження цифрових рішень у навчальні заклади є не лише вимогою часу, а й відповіддю на постійно зростаючі обсяги інформації, яку необхідно обробляти, аналізувати та використовувати для ухвалення рішень. Удосконалення управління навчальними процесами, підвищення ефективності адміністративних завдань та оптимізація ресурсів є ключовими викликами для закладів освіти. Зважаючи на те, що обсяг завдань і кількість інформації в навчальних закладах значно зростає, використання сучасних технологій стає незамінним для забезпечення якісного функціонування освітніх установ.

ERP-системи, такі як Odoo, стають важливим інструментом у процесі діджиталізації освітнього сектору завдяки своїй здатності об'єднувати різноманітні процеси в одну інтегровану платформу. Ця система дозволяє не лише автоматизувати рутинні завдання, а й створювати єдиний інформаційний простір, який забезпечує доступ до даних у реальному часі, сприяє покра-

щенню комунікації між усіма учасниками освітнього процесу та мінімізує ймовірність помилок у роботі. В умовах дистанційного навчання, коли потреба в швидкому доступі до інформації, гнучкості та доступності є критично важливою, ERP-системи набувають особливого значення. Інтеграція ERP Odoo дозволяє забезпечити централізоване управління розкладами, обліком успішності студентів, фінансовим плануванням, а також комунікацією між викладачами, студентами та адміністрацією.

Інтеграція ERP систем в управління освітнім процесом

Унікальність Odoo полягає в тому, що це платформа з відкритим кодом, яка дозволяє адаптувати її функціонал відповідно до специфічних потреб конкретного навчального закладу. Вона може бути налаштована під особливості роботи як великих університетів, так і малих коледжів чи навіть шкіл. Завдяки своїй модульній структурі система дає можливість створювати рішення, що відповідають вимогам конкретних освітніх установ, включаючи індивідуальні налаштування розкладів, облік академічної успішності, контроль відвідуваності, а також забезпечення прозорої та ефективної взаємодії між усіма учасниками освітнього процесу. Особливої уваги заслуговує можливість розгортання Odoo у хмарному середовищі, що гарантує доступність системи з будь-якого місця та на будь-якому пристрої, забезпечуючи її масштабованість і знижуючи витрати на інфраструктуру.

Сучасний розвиток технологій потребує від навчальних закладів не лише адаптації до нових реалій, а й постійного вдосконалення внутрішніх процесів, спрямованих на ефективне управління та організацію освітньої діяльності. У цьому контексті ERP Odoo виступає як універсальний інструмент, здатний змінити підхід до управління освітнім процесом. Вона дозволяє забезпечити автоматизацію багатьох завдань, які раніше виконувалися вручну, зокрема, складання розкладів, ведення обліку успішності, моніторинг відвідуваності, управління кадрами та планування фінансів. Це не лише знижує навантаження на адміністрацію, а й дозволяє зосередитися на стратегічних аспектах розвитку навчального закладу.

Впровадження ERP-систем у сфері освіти вже продемонструвало свою ефективність у багатьох країнах світу. Водночас адаптація таких систем до потреб українських освітніх установ потребує додаткових досліджень, оскільки кожен навчальний заклад має свої унікальні вимоги та особливості. У цьому контексті важливим є не лише технічний аспект інтеграції системи, а й врахування педагогічних, організаційних та культурних факторів, які впливають на її успішне впровадження. Odoo дозволяє створювати індивідуальні рішення, які враховують ці фактори, забезпечуючи максимальну ефективність та зручність використання.

Розгортання ERP Odoo в хмарному середовищі є однією з ключових переваг цієї системи. Використання хмарних технологій забезпечує не лише гнучкість і доступність, але й дозволяє мінімізувати витрати на інфраструктуру, що є особливо важливим для навчальних закладів з обмеженим бюджетом. Крім того, це створює передумови для безперебійного функціонування системи, навіть у випадку технічних проблем або форс-мажорних обставин, таких як пандемії чи інші кризові ситуації, які можуть ускладнити традиційну організацію освітнього процесу.

Автоматизація рутинних адміністративних завдань, яка стає можливою завдяки впровадженню ERP Odoo, дозволяє значно скоротити час, необхідний для їх виконання. Наприклад, автоматичне складання розкладів враховує всі можливі обмеження та конфлікти, забезпечуючи оптимальне планування навчальних занять. Так само автоматизований облік успішності дозволяє не лише спростити процес внесення оцінок, а й забезпечити прозорість цього процесу для студентів, викладачів та адміністрації. Це сприяє підвищенню довіри між усіма учасниками навчального процесу та забезпечує доступ до актуальної інформації в режимі реального часу.

Ще одним важливим аспектом впровадження ERP Odoo є підвищення якості управлінських рішень. Завдяки інтеграції всіх процесів у єдину інформаційну систему адміністрація навчального закладу

отримує доступ до повного спектра даних, необхідних для аналізу та прийняття обґрунтованих рішень. Це дозволяє не лише оперативно реагувати на поточні виклики, але й прогнозувати розвиток ситуації та планувати подальші дії. Крім того, централізоване зберігання даних забезпечує їх захищеність і цілісність, що є важливим у сучасних умовах цифровізації.

В умовах зростаючого попиту на дистанційне навчання, ERP Odoo стає незамінним інструментом для організації освітнього процесу. Вона забезпечує не лише доступ до навчальних матеріалів, але й можливість інтерактивної взаємодії між студентами та викладачами, що є важливим для збереження якості освіти навіть за умов віддаленого навчання. Інтерактивні функції, такі як форуми, обговорення, електронні журнали та тестування, дозволяють створювати ефективний навчальний процес, який відповідає сучасним вимогам і очікуванням студентів.

Впровадження ERP Odoo у навчальні заклади є важливим кроком до інтеграції України в світовий освітній простір. Використання сучасних технологій не лише підвищує ефективність управління, а й сприяє формуванню позитивного іміджу освітніх установ, які прагнуть відповідати найкращим світовим стандартам. Це створює передумови для підвищення конкурентоспроможності українських навчальних закладів, залучення іноземних студентів та інтеграції до міжнародних освітніх програм.

Дослідження впровадження ERP-систем у сфері освіти активно розвивається. Зокрема, у роботі [1] Л. В. Ноздріної та О. І. Критовича описано підходи до діджиталізації фандрейзингової діяльності вищих шкіл за допомогою ERP-систем, що доводить важливість таких інструментів для підвищення конкурентоспроможності освітніх закладів. Однак, адаптація цих систем до специфічних потреб українських навчальних закладів досліджена недостатньо. Особливий інтерес становить можливість створення індивідуальних рішень для фахових коледжів і університетів на основі Odoo.

Результати цього дослідження можуть стати основою для подальшого впровадження цифрових технологій в освітній сектор України, сприяючи інтеграції до світового освітнього простору та підвищенню ефективності управлінських процесів у навчальних закладах.

Дослідження та оцінка ERP ODOO

Для визначення основних завдань дослідження необхідно оцінити роль ERP-систем як ефективного інструменту автоматизації управлінських процесів у закладах освіти. Головна мета впровадження таких систем полягає в інтеграції адміністративних і навчальних функцій в єдину інформаційну платформу, яка відповідала б сучасним вимогам організації освітнього процесу.

ERP Odoo дозволяє автоматизувати низку ключових операцій, таких як управління навчальними планами, облік академічної успішності студентів, складання розкладу занять, моніторинг відвідуваності, кадрове управління та організація взаємодії між учасниками освітнього процесу. Завдяки інтеграції цих процесів в єдину систему забезпечується централізоване зберігання даних, доступ до яких можливий у реальному часі. Це сприяє значному покращенню якості управлінських рішень, оперативності їх ухвалення та зниженню ймовірності помилок у роботі.

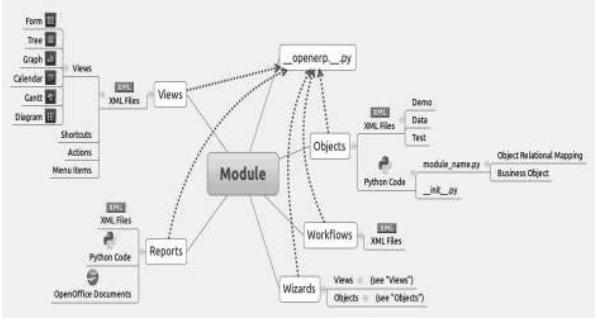


Рис. 1. Архітектура ERP Odoo

Архітектура ERP Odoo (Рис. 1.), яка застосовується в навчальних закладах, базується на модульній структурі. Це дозволяє налаштовувати систему відповідно до потреб конкретної організації. У контексті розробки модуля для управління освітнім процесом вона забезпечує інтеграцію таких функцій (рис. 2.), як планування навчального процесу, контроль за виконанням навчальних планів та створення звітності. Розроб-

лювана система має інструменти для забезпечення доступу користувачів різних рівнів (адміністрації, викладачів, студентів) до необхідної інформації, що створює умови для прозорості та взаємодії між усіма учасниками освітньої діяльності.

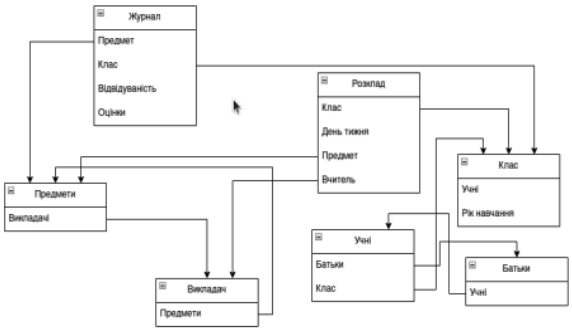


Рис. 2. Архітектура розроблюваної системи

Окрім забезпечення автоматизації, ERP-система сприяє оптимізації роботи закладу освіти. Вона значно скорочує час, необхідний для виконання рутинних адміністративних завдань, мінімізує дублювання інформації та забезпечує зручність підготовки звітності. Ці переваги роблять систему важливим інструментом для підвищення продуктивності освітньої установи, дозволяючи адміністрації зосередитися на стратегічних аспектах управління, а викладачам – на навчальній діяльності.

Реалізація дослідження передбачала розробку модуля управління освітнім процесом відповідно до діаграми зображеної на Рис. 2. Основним інструментом для написання бекенд-частини модуля виступила мова програмування Python, яка забезпечує гнучкість у реалізації функціональних можливостей системи. Для створення інтерфейсу користувача застосовувалися технології HTML, CSS та JavaScript, що дозволяють реалізувати сучасний, зручний і функціональний фронтенд.

Модуль інтегрує ключові освітні компоненти, дозволяючи автоматизувати такі процеси, як складання розкладу, управління навчальними планами, облік успішності та відвідування, а також налаштування роботи студентів і викладачів. Архітектура системи побудована так, щоб забезпечити централізоване зберігання даних та їхню доступність для всіх учасників освітнього процесу.

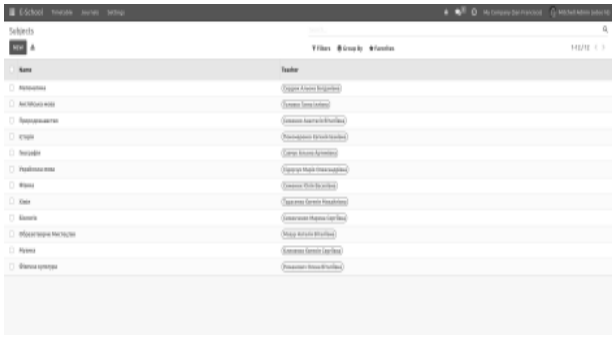


Рис. 3. Освітні компоненти

У модулі реалізовано функції для управління викладачами, що дозволяє враховувати їхнє навантаження, вести облік особистих даних та взаємодіяти з іншими компонентами системи.

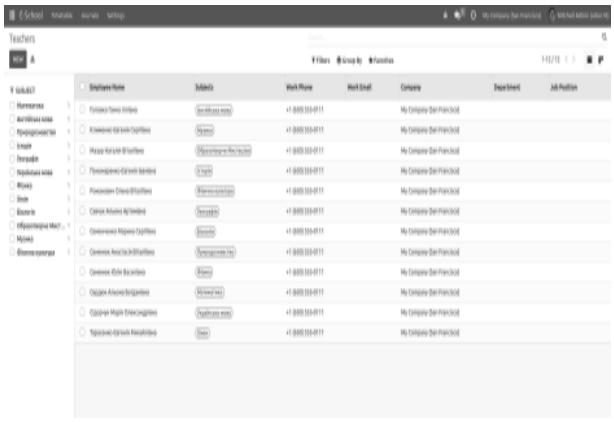


Рис. 4. Викладачі

Особливу увагу приділено створенню зручного інтерфейсу для роботи з розкладом, який автоматично виявляє неточності та пропонує оптимальні варіанти для формування графіка навчальних занять.

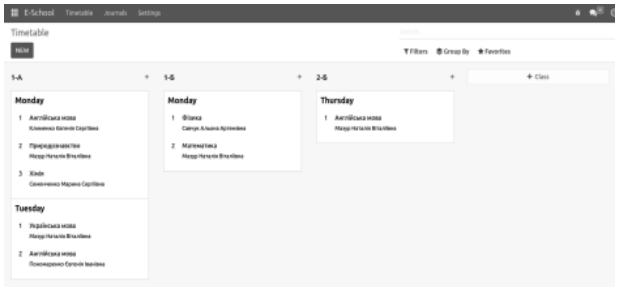


Рис. 5. Розклад

За допомогою системи для студентів реалізовані інструменти, що надають доступ до інформації про розклад, успішність та на-

вчальні матеріали, забезпечуючи інтерактивність і прозорість освітнього процесу.

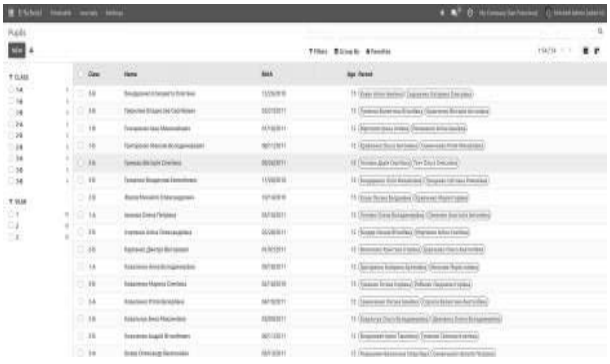


Рис. 6. Студенти

Система дозволяє адмініструвати налаштування інформації про студентів у групах, забезпечуючи можливість управління навчальними планами, груповими завданнями та іншими параметрами.

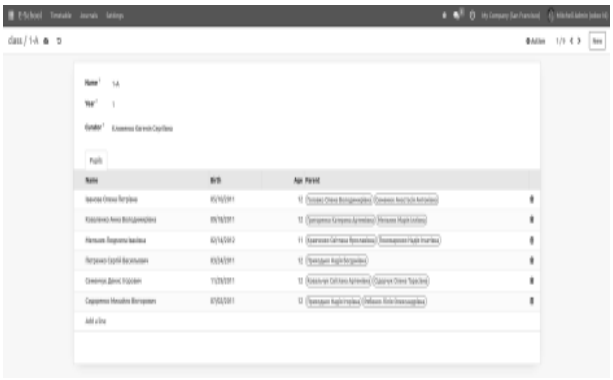


Рис. 7. Налаштування інформації про студентів у групі

Wizard (майстер) відкриття журналу створений для зручного доступу до інформації про відвідуваність і успішність за певний день та предмет.

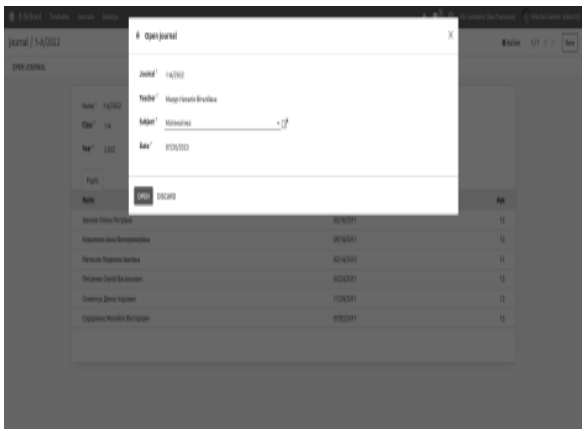


Рис. 8. Wizard відкриття журналу на певну дату з предметом

Окрім цього, побудовано функціонал для роботи з журналом відвідуваності та оцінювання, який дозволяє викладачам оперативно вносити дані та переглядати звіти.

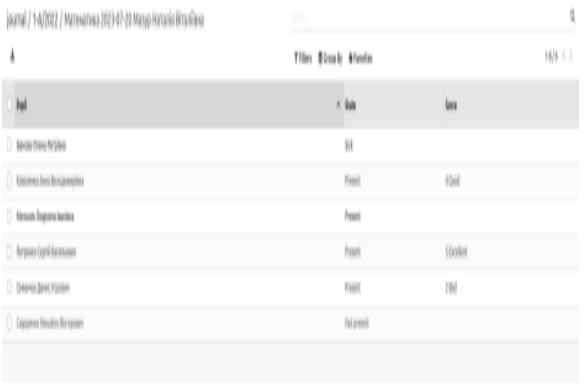


Рис. 9. Робота з журналом відвідування і оцінювання

Тестування

Тестування розробленого модуля проводилося на базі коледжу ВСП «РФК НУБіП України» м. Рівне, де була створена унікальна можливість оцінити його функціональні можливості у реальних умовах експлуатації. Цей етап дослідження відіграв важливу роль у визначенні ефективності впровадженої системи, її здатності вирішувати нагальні завдання управління освітнім процесом і покращувати внутрішні процеси навчального закладу. Особливу увагу під час тестування було приділено автоматизації рутинних завдань, таких як ведення обліку успішності студентів, складання розкладу занять, моніторинг відвідуваності студентів і адміністрування роботи викладачів. Ці процеси зазвичай вимагають значних зусиль і часу з боку адміністрації та викладачів, а їхня автоматизація дозволила суттєво підвищити ефективність роботи.

У ході тестування модуль продемонстрував свою здатність значно скоротити витрати часу на обробку великих обсягів даних. Наприклад, автоматизоване ведення журналу успішності дало змогу викладачам зосередитися на навчальній діяльності, мінімізувавши час, необхідний для внесення оцінок та підготовки звітів. Система автоматично обробляла дані про академічні досягнення студентів, генеруючи структуро-

вані звіти, які відповідали потребам адміністрації та вимогам зовнішніх інстанцій. Наприклад, звіти містили детальну статистику про середній бал студентів за семестр, кількість перездач та динаміку успішності, що спрощувало аналіз результатів. Завдяки цьому було зменшено ймовірність людських помилок, уникнуто плутанини у розрахунках середнього балу та забезпечено збереження всіх даних у захищеному електронному форматі.

Функціонал модуля також охоплював автоматизоване складання розкладу занять, що виявилося особливо корисним для забезпечення безперебійного навчального процесу. Ця функція враховувала численні параметри, такі як доступність викладачів, завантаженість аудиторій та уникнення конфліктів у розкладі, що раніше вимагало ручної перевірки та значних зусиль з боку персоналу. Наприклад, система автоматично розподіляла лекції так, щоб у викладачів не виникало накладок між різними групами, а студенти не отримували пари в нерівномірному порядку, з великими проміжками або пізно ввечері без потреби. Крім того, вона оперативно перерозподіляла заняття у разі змін, таких як відрядження викладача або тимчасова недоступність аудиторії, що дозволяло уникати збоїв у навчальному процесі.

Моніторинг відвідуваності студентів також став простішим і ефективнішим завдяки використанню розробленого модуля. Замість ручного заповнення паперових журналів викладачі могли відзначати присутність студентів у цифровій формі одним кліком, а система автоматично підраховувала пропуски та формувала звіти. Крім того, ця інформація була доступна у режимі реального часу, що дозволяло викладачам і адміністрації оперативно реагувати на випадки невідвідуваності та вживати відповідних заходів. Студенти та їхні батьки також отримали можливість переглядати ці дані через інтерактивний інтерфейс, що підвищувало прозорість навчального процесу та сприяло мотивації до відвідування занять.

Система також продемонструвала свою гнучкість у налаштуванні, що є важливим для навчальних закладів із різними потребами. Наприклад, у процесі тестування вдалося налаштувати функціонал модуля для вирішення унікальних завдань коледжу ВСП «РФК НУБіП України», таких як управління спеціалізованими навчальними програмами чи забезпечення підтримки для викладачів у їхній повсякденній роботі. Це дозволяє адаптувати модуль для використання в інших освітніх установах з урахуванням їхніх особливостей, що робить систему універсальним рішенням для автоматизації управління.

Завдяки інтерактивному інтерфейсу всі учасники освітнього процесу отримали зручний доступ до необхідної інформації. Викладачі могли переглядати розклад, вносити оцінки та коментарі, створювати індивідуальні навчальні плани для студентів і навіть спілкуватися з ними через систему. Для студентів був доступний персоналізований кабінет, у якому вони могли переглядати свої успіхи, дізнаватися розклад занять, отримувати доступ до навчальних матеріалів та взаємодіяти з викладачами. Це створило умови для прозорого та зручного навчального процесу, підвищуючи довіру між усіма учасниками освітнього середовища.

Важливим аспектом тестування було оцінювання зручності використання системи. Інтуїтивно зрозумілий інтерфейс дозволив користувачам швидко освоїти роботу з модулем, що стало одним із визначальних факторів його успішного тестового використання. Навчальні заходи, організовані для викладачів і адміністративного персоналу, показали, що система не вимагає значних зусиль для опанування її функціоналом, а підтримка з боку розробників забезпечила швидке вирішення будь-яких технічних питань.

Окрім цього, під час тестування була відзначена висока надійність і безпека роботи системи. Завдяки розгортанню у хмарному середовищі забезпечувалася стабільна робота модуля навіть за умов високих навантажень. Захищеність даних гарантувала, що конфіденційна інформація про студентів, викладачів та навчальні процеси залишалася недоступною для сторонніх осіб. Це стало важливим аргументом на користь впровадження системи, оскільки за-

безпечення інформаційної безпеки є пріоритетом для сучасних навчальних закладів.

Результати тестування показали, що впровадження розробленого модуля сприяє значному покращенню ефективності управління навчальним процесом. Адміністрація закладу отримала інструмент для ухвалення обґрунтованих рішень на основі актуальних даних, викладачі змогли зосередитися на викладанні замість виконання рутинних завдань, а студенти – отримати доступ до всіх необхідних для навчання ресурсів. У перспективі така система має потенціал для подальшого розвитку й адаптації, відкриваючи нові можливості для цифровізації освітньої діяльності.

Висновки

Розроблений модуль управління освітнім процесом на основі ERP Odoo підтвердив свою ефективність у реальних умовах навчального закладу. Його впровадження може забезпечити:

- автоматизацію ключових освітніх і адміністративних процесів;
- зменшення витрат часу на виконання рутинних завдань;
- підвищення точності та доступності інформації для ухвалення управлінських рішень;
- покращення комунікації між адміністрацією, викладачами та студентами.

Впровадження такого рішення створює передумови для подальшого вдосконалення процесів управління в закладах освіти, сприяє інтеграції сучасних інформаційних технологій у навчальну діяльність і підвищує загальну ефективність функціонування освітніх установ.

Отримані результати можуть бути використані для розробки рекомендацій щодо адаптації ERP Odoo у закладах освіти різних рівнів, що робить це дослідження актуальним і корисним для подальшого розвитку цифровізації освіти в Україні.

Література

1. Ноздріна, Л. В., Критович, О. І. (2022). Застосування ERP-систем у фандрейзингу вищої школи. Науковий економічний вісник, 31 березня URL: https://ser.net.ua/index.php/SER/article/download/432/446 (дата звернення: 26.01.2025).
2. Посібник користувача odoo, Режим доступу https://www.odoo.com/documentation
3. Посібник розробника odoo, режим доступу https://www.odoo.com/documentation/17.0/developer.html
4. Brunn H. Odoo Development Cookbook / H. Brunn, A. Fayolle, D. Reis. - Birmingham: Packt Publishing Ltd, 2016. - 377 с.
5. Безус П.І., Серебряков Р.А. Застосування автоматизованих інформаційних систем управління у вищих навчальних закладах/ Науковий вісник Академії муніципального управління. – Серія Економіка. – Вип.6. – С. 12-17.

Одержано: 24.02.2025
Внутрішня рецензія отримана: 28.02.2025
Зовнішня рецензія отримана: 04.03.2025

Про авторів:

Біда Петро Іванович,
кандидат технічних наук,
https://orcid.org/0000-0003-0266-9974

Надозірний Святослав Вікторович,
викладач, спеціаліст вищої категорії

Кот Василь Васильович,
кандидат технічних наук
https://orcid.org/0000-0002-5139-4391

Місце роботи авторів:

Відокремлений структурний підрозділ
«Рівненський фаховий коледж
Національного університету біоресурсів
та природокористування України»,
33001, м. Рівне, вул. Коперніка, 44

P.I.Bida1976@gmail.com
nadozirny.s@gmail.com
kotpm04@ukr.net

ВИМОГИ ДО ОФОРМЛЕННЯ СТАТЕЙ

У журналі "Проблеми програмування" публікуються наукові матеріали, які раніше не публікувалися в інших виданнях.

Мова статті: українська, англійська. Обсяг статті - від 6 до 16 сторінок формату А4. Документ зберігається у форматі doc або docx. Назва файлу включає транслітерацію прізвища автора (авторів), наприклад, “Petrenko.doc”.

Стаття надається без нумерації сторінок.

Автори можуть користуватися електронною поштою для передачі до редакції тексту статті, ділової переписки та правки при коректурі. E-mail редакції: alengoro@isofts.kiev.ua, alengoro2022@gmail.com. Телефон: +380 (96) 418 3082. Для надійності інформаційного обміну в умовах можливих відключень електрики – прохання надсилати матеріали на обидві електронні пошти одночасно.

Оформлення файлу з текстом статті

При підготовці файлу використовуються: стиль нормальний (звичайний) або normal; шрифт Times New Roman, розмір шрифту 12 пт.; міжрядковий інтервал – 1,0; абзацний відступ -1,25 см; вирівнювання – по ширині. У тексті не допускається вирівнювання пропусками; розстановка переносів – автоматична. Формат паперу А4, розміри полів документа – 20 мм. Текст статті після зазначення авторів, назви і анотацій (двома мовами) має бути оформлений **у 2 колонки,** ширина яких – 7,86 см, а пробіл між ними – 1,27 см.

Послідовність розміщення та оформлення матеріалу статті

Верхній колонтитул: назва рубрики відповідно до переліку, прийнятому редакцією журналу *(пропонується авторами, остаточно уточняється редакцією).*

УДК *(зліва під рискою верхнього колонтитулу)**:*** індекс за універсальною десятковою класифікацією; ***DOI:*** в тому ж рядку правіше *(заповнюється редакцією)*.

Автори: ініціали та прізвища авторів, курсив *(світлий)*.

Заголовок 1 ***(назва статті)***: не містить абревіатур та строго відповідає змісту статті. Шрифт 15 пт, напівжирний, регістр верхній, вирівнювання по центру.

Анотація: 1800-2100 знаків враховуючи пробіли, не має бути абревіатур. Шрифт 10 пт, звичайний.

Ключові слова: не більше 10 слів, не містить абревіатур, подаються в називному відмінку, розділені комами. Шрифт 10 пт, звичайний.

УВАГА!

Автори, заголовок статті, анотація і ключові слова зазначаються **ДВІЧІ:** українською і англійською мовами. Спочатку мовою статті, потім іншою мовою.

Нижній колонтитул (тільки для першої сторінки) включає стандартну інформацію Copyright: перший рядок – прізвища авторів, рік; другий рядок – номер ISSN, назва журналу, рік, номер випуску.

Заголовок 2 ***(назва розділу)***: шрифт 14 пт, напівжирний; абзац із центральним вирівнюванням, без переносів. Заголовки нижчого рівня *(пункти і т.п.)* у самостійний абзац не виділяються і проходять першим реченням текстового абзацу, шрифт 12 пт, напівжирний.

Формули створюються в редакторі Microsoft Equation 3.0 або MathType. Формули, на які є посилання в тексті, повинні мати наскрізну нумерацію. Номер формули друкується в круглих дужках біля краю правого поля. Розмір основного шрифту редактора формул – 12 пт. Розміри символів у формулах: звичайний – 12 пт, великий індекс – 9 пт, дрібний індекс

– 7 пт, великий символ – 18 пт, дрібний символ – 11 пт. Не допускається масштабування формульних об’єктів. Великі формули мають бути розбиті на декілька рядків.

Наприклад:

$$\frac{\partial T}{\partial t}+\frac{u}{a\cos\varphi}\frac{\partial T}{\partial \lambda}+\frac{v}{a}\frac{\partial T}{\partial \varphi}+w\frac{\partial T}{\partial z}=\frac{\delta}{c_V\rho}, \quad (1)$$

де λ – довгота, φ – широта, z – висота над рівнем моря, $V=(u,v,w)$, a – радіус Землі, ω – швидкість добового обертання Землі, $F_f=(F_\lambda, F_\varphi, F_z)$.

Рисунки мають бути створені вбудованим редактором Word Picture або експортовані з прикладних програм Windows у графічних форматах (bmp, pcx, gif, jpg або tif). Рисунки розташовуються по центру. Нумерація рисунків здійснюється відповідно до порядку згадування у тексті. Нумеровані підписи розміщуються під рисунком з позначенням "Рис. ", далі вказується номер рисунка і текст підпису.

Таблиці мають бути підготовлені стандартним вбудованим в Word інструментарієм "Таблиця". Таблиці нумеруються за порядком згадування. На номер таблиці мають бути посилання в тексті. Номер таблиці вказується в окремому рядку з вирівнюванням по правій стороні (наприклад: "Таблиця 1"). Назви таблиць розміщуються над таблицею з вирівнюванням по центру. Мінімальний розмір шрифту в таблицях – 11 пт.

При посиланні на формулу, рисунок, таблицю або літературне джерело, використовуйте наступні позначення відповідно: (1), (1, 2); Рис.1, Рис.1, 2; Табл.1., Табл.1, 2; [1], [1, 2].

Основний текст статті має такі необхідні елементи:

– постановка проблеми в загальному вигляді і її зв'язок з важливими науковими або практичними завданнями;

– аналіз останніх досліджень і публікацій, у яких розпочато рішення даної проблеми і на які спирається автор, виділення невирішених раніше частин загальної проблеми, яким присвячується дана стаття;

– формулювання цілей статті (постановка задачі);

– виклад основного матеріалу дослідження з повним обґрунтуванням отриманих наукових результатів;

– висновки з даного дослідження і перспективи подальших розробок у даному напрямку;

– подяка (за наявності такої).

Наведений вище маркований список має наступні параметри: маркер має відступ 1,25 см, текст для першого рядка – 1,9 см. Аналогічні відступи слід підтримувати і для нумерованого списку.

Література: єдиний нумерований список джерел згідно ДСТУ 8302:2015 від 01.07.2016 р., шрифт 11 пт, відступ: спеціальний, навислий, 0,63 см.

References: література англійською мовою подається як список використовуваних джерел згідно *Harvard Style*. Джерела з заголовками на латиниці наводяться без перекладу. Для літератури джерел на мовах, що не використовують латинський алфавіт, необхідно забезпечити переведення назв джерел і вказати після них у дужках мову оригіналу. Прізвища та ініціали авторів, слід транслітерувати за правилами як для закордонного паспорта.

Література, що надана другою мовою не враховується при підрахунку кількості сторінок статті.

Дата надходження статті позначається редакцією цифрами окремим рядком після слова «Одержано:»/”Received:”.

Дата надходження внутрішньої рецензії позначається редакцією цифрами окремим рядком після слів «Внутрішня рецензія отримана»/”Internal review received:”.

Дата надходження зовнішньої рецензії позначається редакцією цифрами окремим рядком після слів «Зовнішня рецензія отримана:»/” External review received:”.

Відомості про рецензентів конкретної статті не розголошуються для підвищення об’єктивності рецензування.

***Дані про авторів*:** мають починатися рядком “*Про авторів:*”, напівжирний курсив. Далі вказуються для кожного з авторів ПІБ повністю, вчений ступінь, наукове звання, посада, обов’язково номер ORCID (сайт ORCID http://orcid.org/). Особисті телефони та електронні пошти авторів вказуються тут тільки, якщо автор хоче, щоб вони були опубліковані в журналі.

Перелік авторів подається під номерами (надстроковим шрифтом), що відповідають нумерації місць роботи, де вони працюють (надстроковим шрифтом).

***Дані про місце роботи авторів*:** починаються рядком “*Місце роботи авторів:*”, напівжирний курсив. Далі вказуються місце роботи, телефон, електронна пошта, веб-сайт.

Обов’язково вказати мобільний телефон і e-mail відповідального виконавця для роботи з редактором при підготовці статті до друку. Ця інформація не публікується і призначена виключно для контакту редактора з авторами.

16.07.2020 р. набули чинності положення Закону України «Про забезпечення функціонування української мови як державної». Відповідно до статті 22 «Державна мова у сфері науки» у наукових виданнях не повинно бути вміщено матеріалів іншими мовами, окрім державної, англійської та мов ЄС.

Підписний індекс журналу "Проблеми програмування" – **90853**.