НАЦІОНАЛЬНА
АКАДЕМІЯ НАУК УКРАЇНИ
ІНСТИТУТ ПРОГРАМНИХ СИСТЕМ

ПРОБЛЕМИ ПРОГРАМУВАННЯ

науковий журнал

№ 2 ***квітень - червень*** ***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 IN PROGRAMMING

scientific journal

№ 2 ***April – June*** ***2025***

Founded in March, 1999

CONTENT

УДК 681.3 http://doi.org/10.15407/pp2025.02.003

П.І. Андон, С.В. Суботін, П.І. Перконос, Є.М. Твердохліб

СУЧАСНИЙ СТАН ТА ПРОБЛЕМИ РОЗВИТКУ СЕРВІС-ОРІЄНТОВАНОЇ ПАРАДИГМИ

На основі аналізу розвитку програмної інженерії визначені основні принципи та джерела сервіс-орієнтованої парадигми. Розглянуті архітектурні особливості та основні шаблони сервіс-орієнтованих застосувань та стандарти, на яких вони ґрунтуються. Надана характеристика програмної інженерії розробки сервіс-орієнтованих застосувань та сучасного стану засобів її підтримки. Виявлені основні проблеми побудови сервіс-орієнтованих застосувань та запропоновані шляхи їх подолання.

Ключові слова: автоматизація наукових досліджень, інтегрована інформаційна інфраструктура, корпоративні інформаційні системи, сервіс-орієнтована архітектура, Semantic Web, семантичний веб-сервіс, онтологія.

Ph.I. Andon, S.V. Subotin, P.I. Perkonos, Ie.M. Tverdokhlib

CURRENT STATE AND EXTENTION PROBLEMS OF THE SERVICE-ORIENTED PARADIGM

Based on the analysis of the development of software engineering, the main principles and sources of the service-oriented paradigm are identified. The architectural features and basic templates of service-oriented applications and the standards on which they are based are considered. The software engineering of service-oriented application development and the current state of its support tools are described. The main problems of building service-oriented applications are identified and ways to overcome them are proposed.

Keywords: automation of scientific research, integrated information infrastructure, corporate information systems, service-oriented architecture, Semantic Web, semantic web service, ontology.

Вступ

В умовах всеосяжної інформатизації суспільства та насиченості його обчислювальними ресурсами, поєднаними відомчими та глобальними мережами, домінуючою парадигмою побудови та використання програмних засобів стає сервіс-орієнтована парадигма. Тому розуміння основних принципів та джерел становлення цієї парадигми, архітектурних особливостей, стандартів, що лежать в її основі, сучасного стану засобів програмної інженерії, котрі забезпечують створення та використання розподілених сервіс-орієнтованих застосувань, є запорукою ефективного створення та використання прикладних програмних засобів.

Джерела та еволюція сервіс-орієнтованої парадигми

Сервіс-орієнована парадигма сформувалась в результаті еволюції програмної інженерії відповідно до розвитку обчислювальної інфраструктури й потреб суспільства та успадкувала принципи й методи, відпрацьовані в попередніх парадигмах, що домінували в різні часи (рис.1).

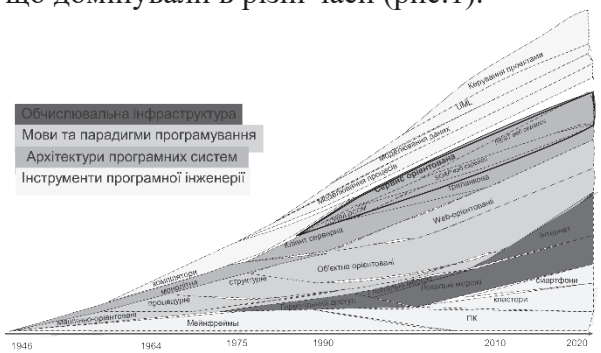


Рис. 1. Еволюція програмної інженерії

Один з основних принципів - принцип повторного використання коду набував різних форм у різних парадигмах по-

чинаючи з машинно-орієнтованого програмування. Поява універсальних мов програмування (Алгол, Фортран) дозволила повторно використовувати програми на комп'ютерах із різною системою команд, а також компонувати складні алгоритми з типових частин завдяки процедурній парадигмі програмування.

Складність процесів, що автоматизувались, швидко зростала, і логіка їх виконання, заснована на операторах прямих посилань, ставала неконтрольованою та важкою для супроводження. Для забезпечення прозорості алгоритмів були запропоновані типові принципи організації логіки виконання, засновані на прозорих конструкціях послідовного, умовного та циклічного виконання, відомі як структурне програмування . Мови програмування, засновані на цій парадигмі (Pl/1, Pascal, C), швидко витіснили застарілі Fortran та Algol. Структурна парадигма започаткувала модульний принцип побудови програм, згідно з якою складна система декомпозується на ряд модулів, які могли розроблятись незалежно різними виконавцями і потім збиратись у готову систему.

Але для цього потрібно було здійснювати попереднє проєктування системи, виходячи з вимог до її функціональності. Для забезпечення проєктування програмних систем виникла методологія моделювання різних аспектів систем за допомогою графічних нотацій SADT (structured analysis and design technique) [1].

Централізована обробка даних різними користувачами на одній «потужній» ЕОМ забезпечила взаємодію користувачів та використання напрацювань за рахунок їх розміщення в спільній файловій системі мейнфрейму та обміну даними й програмами на магнітних носіях.

Наступним кроком організації прикладних застосувань була розробка баз даних (БД) - пойменованих сукупностей даних, організованих за певними правилами, що передбачають загальні принципи опису, зберігання і маніпулювання даними, незалежно від прикладних програм. Перші БД, запропоновані для використання на мейнфреймах, використовували ієрархічну або мережеву (графову) модель взаємозв'язків між записами БД та відповідні мови опису та доступу до даних, швидко витіснених більш гнучкою реляційною моделлю даних, яка стала однією із складових методології SADT та відтворена в CASE середовищах. Зберігання даних у централізованому структурованому сховищі зі стандартизованими методами доступу до них спростило користувачам обмін даними та забезпечило поєднання програм, що автоматизують окремі операції технологічного ланцюга в єдину систему з автоматичною передачею даних між ними.

З появою мікропроцесорів обчислювальна інфраструктура підприємств почала стрімко змінюватись. Поряд із мейнфреймами з'явились міні та персональні комп'ютери, вартість придбання та утримання яких була значно меншою, а за продуктивністю вони практично не поступалися мейнфреймам. Тому підприємства швидко насичувались персональними комп'ютерами, і обробка даних почала виконуватись розподілено на багатьох робочих місцях, розосереджених по всьому підприємству. Однак водночас виникла потреба інформаційної взаємодії та спільного використання даних і програмних напрацювань, яка на мейнфреймах вирішувалася природним чином за рахунок централізованої обробки на одній ЕОМ і розміщення даних в її файловій системі та базах даних.

Ця проблема була швидко вирішена шляхом поєднання окремих розподілених на підприємстві комп'ютерів у мережу завдяки досвіду забезпечення доступу до мейнфреймів з терміналів, віддалених на багато сотень, а то й тисяч кілометрів через канали зв'язку. Локальні мережі забезпечили сумісне використання розподілених обчислювальних ресурсів підприємства і доступ до спільної файлової системи та баз даних. Монолітна архітектура прикладних систем, притаманна мейнфреймам, практично була витіснена дволанковою клієнт-серверною архітектурою, за якою функціонал структурованого зберігання та доступу до даних виконувався окремим застосуванням (СУБД) на окремому сервері. Їх обробка відбувалася на інших застосунках, які взаємодіяли з сервером за стандар-

тними мережевими протоколами та стандартною мовою запитів до даних.

Подальшим розвитком структурного програмування стала об'єктно-орієнтована парадигма (ООП), яка поєднала в одній конструкції (класі) структуру опису об'єкта автоматизації та його поведінку (процедуру обробки його даних) і розглядала прикладну систему як динамічну сукупність взаємодіючих об'єктів. Механізми інкапсуляції структури опису об'єктів та їхньої поведінки значно спростили логіку прикладної системи, а механізми наслідування та поліморфізму гнучкість повторного використання готових рішень. Тому в широко доступні структурні мови програмування додавалась підтримка об'єктно-орієнтованої парадигми (C->C++, Pascal->Object Pascal), створювались нові мови підтримки ООП (JAVA, Python).

Також, як і для методології структурного програмування, були запропоновані методології IDEF та відповідні CASE засоби, для моделювання систем в ООП була запропонована методологія UML [2].

Повторне використання реалізованих алгоритмів у новій прикладній системі в процедурній або в об'єктно-орієнтованій парадигмі потребувала їхньої компіляції та зв'язування із власним кодом у монолітну, виконувану в одному адресному просторі програму. Водночас розробник був обмежений у використанні готових рішень рамками платформи розробки своєї системи. Тому фірма Microsoft запропонувала технологію COM (Component object model) - виклику методів об'єктів, засновану на стандартизації опису інтерфейсу методів. Водночас код виконується паралельно на тій же ЕОМ в окремому адресному просторі як готовий компонент і не потребує компіляції та вбудовування в прикладну систему. За приклад використання цієї технології можна навести механізми OLE (Object linking and embeding) вбудовування в документи офісних систем частин, обробка яких здійснюється іншими застосуваннями. Наступним кроком розвитку COM технології стала специфікація DCOM (distributed COM), яка дозволяла здійснювати віддалений виклик процедур об'єктів, що створені та існують іншою програмою на іншій ЕОМ з передачею параметрів та отриманням результату по мережі. Нажаль, ці специфікації були реалізовані тільки в рамках операційної системи Microsoft Windows. Тому групою OMG була розроблена альтернативна специфікація CORBA цього механізму інтеграції ізольованих систем, розроблених різними мовами програмування, працюючих в різних вузлах мережі на різних платформах та взаємодіючих одна з одною так само просто, наче вони знаходились в адресному просторі одного процесу, що можна вважати початком сервіс-орієнтованої парадигми.

Перша глобальна мережа ARPANET створювалась як закрита мережа військового призначення під контролем міністерства оборони США. Розроблені в її рамках протоколи TCP/IP [3] адресації вузлів та передачі пакетів даних між ними були закритими та заборонені для використання у відкритих проєктах. Після закриття 1989 року проєкту ARPANET ця заборона була знята, що сприяло стрімкому нарощуванню поєднання вже існуючих локальних мереж в єдину глобальну мережу і використання служб передачі файлів та електронної пошти, створених в рамках проєкту ARPANET.

Для забезпечення сумісності апаратури та програмного забезпечення вузлів глобальної мережі 1994 року було створено консорціум W3C, покликаний розробляти єдині принципи та стандарти (рекомендації) Інтернету. На основі одного з перших стандартів – мови розмітки текстових документів із гіперпосиланнями HTML та протоколу прикладного рівня передачі даних у мережі інтернет http [4] були створені програми браузерів, що підтримують взаємодію користувача через мережу з програмою, що виконується на віддаленому сервері. Це здійснюється через відображення сформованого програмою документа мовою HTML та відправки підготовлених користувачем даних для виконання програми. Такі триланкові WEB застосування на сьогодні практично витіснили традиційні дволанкові клієнт-серверні застосування, оскільки не потребують розгортання на машині користувача,

доступні з будь-якого робочого місця, під'єднаного до Інтернету, в тому числі з мобільного смартфону, та можуть надаватись у користування як сервіс і не потребують від користувача підтримки та обслуговування.

Протокол HTTP не накладає ніяких обмежень на структуру та формат даних, що передаються, тому за його допомогою можна розповсюдити технологію CORBA та DCOM віддаленого виклику процедур та забезпечити взаємодію не тільки користувача з прикладними застосуваннями через браузер, а й між будь-якими застосуваннями, виконуваними на вузлах Internet мережі. Для цього консорціум W3C, враховуючи досвід використання HTML для кодування інтерфейсу користувача, адаптував його для представлення будь-яких структурованих даних XML (extended markup language) та на його основі визначив стандарт міжпрограмної взаємодії через інтернет SOAP (simple object access protocol) [5], який був значно спрощений порівнянно з CORBA. Підтримка SOAP була реалізована в різних платформах програмування та швидко витіснила CORBA для організації взаємодії прикладних застосувань як в рамках одного підприємства ESB (Enterprise Service Bus), так і між підприємствами (B2B).

Завдяки широкому розповсюдженню WEB-застосувань за технологією AJAX, що використовує взаємодією користувачів через браузер із вбудованим інтерпретатором мови JAVAscript, більш популярним для організації програмного інтерфейсу програм через інтернет став архітектурний стиль REST, запропонований Роєм Філдінгом 2000 року [6].

Цей підхід почав витісняти протокол SOAP завдяки тому, що він передбачає віддалений виклик процедур та отримання результату напряму за протоколом HTTP, підтримка якого вбудована в JAVAscript. А формат обміну не обмежується і може бути використаний прийнятий в JAVAscript формат JSON, більш компактний, ніж XML, і тому організація взаємодії здійснюється простіше та ефективніше.

Віддалений виклик процедур мережею на основі стандарту SOAP або REST технології для виконання певної обробки або отримання потрібних даних забезпечує використання готових програмних рішень без необхідності їхньої компіляції та зв'язування в єдиний модуль, скориставшись вже розгорнутим ПЗ на власній інфраструктурі або на інфраструктурі постачальника ПЗ як сервісу. Це в свою чергу дозволяє компонувати прикладні системи автоматизації бізнес-процесів на основі автономних програмних компонент – сервісів, кожний з яких виконує чітко визначену бізнес логіку у власному операційному оточенні, отримує запит на виконання та повертає результат, використовуючи стандартні протоколи та чітко визначений інтерфейс.

Така розподілена сервіс-орієнтована архітектура прикладних систем в умовах розвинутого Інтернету стала домінуючою та поступово витісняє традиційні локальні та клієнт-серверні застосування завдяки:

- скороченню витрат та термінів розробки потрібного програмного забезпечення за рахунок декомпозиції складної системи на більш керовані автономні компоненти (мікросервіси), використання готових сервісів, доступних в Інтернеті;
- забезпеченню використання програмних систем та окремих компонент без необхідності їхнього розміщення та обслуговування на власних технічних ресурсах;
- колективному використанню програмних систем та окремих її компонент;
- масштабуванню та модифікації системи відповідно до потреб шляхом додавання нових компонент або заміщення продуктивнішими.

Сервіс-орієнтовані системи суттєво відрізняються від традиційних десктоп та клієнт-серверних застосувань своєю розподіленою природою та асинхронним виконанням обробки даних на різних вузлах.

Тому технологія та засоби розробки програмного забезпечення на всіх етапах починають розвиватись з урахуванням цих особливостей та підтримувати подієорієнтовану парадигму програмування, яка

забезпечує синхронізацію одночасного виконання окремих процедур процесу.

Архітектурні особливості та шаблони сервіс-орієнтованої парадигми

Згідно із базовою моделлю [7] сервіс-орієнтована архітектура - це парадигма побудови прикладних систем на основі використання взаємодіючих розподілених програмних компонент, які виконуються в різних операційних оточеннях, що можуть належати і бути під контролем різних власників.

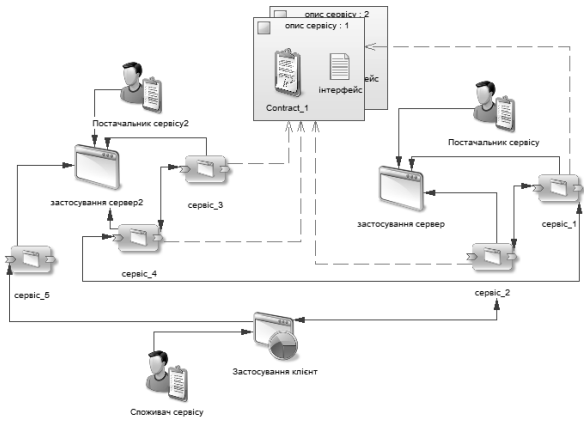


Рис. 2. СОА

У сервіс-орієнтованому застосуванні частина функціональності не реалізується, а її виконання делегується іншому застосуванню-серверу через надсилання повідомлення-запиту на виконання потрібної дії та отримання результату за узгодженим протоколом. Застосування-сервер забезпечує виконання чітко визначеної функціональності (сервісу) у відповідь на отримане повідомлення та відправку результату виконання в чітко визначеному форматі. Зазвичай, це отримання інформації, яку підтримує провайдер, або зміна стану об'єкта в зоні відповідальності провайдера. Водночас клієнт для виконання потрібної йому функціональності може використовувати різні сервіси від різних постачальників, організовуючи логіку свого процесу відповідно до отриманих результатів виконання. В свою чергу сервіс для виконання своєї функціональності також може використовувати інші сервіси, як свої, так і від інших постачальників.

Застосування-сервер розробляється, розгортається та підтримується постачальником сервісу (провайдером), який гарантує виконання сервісу за запитами потенційних користувачів за визначеним протоколом відповідно до умов постачання на ресурсах провайдера. Для того, щоб користувач зміг скористатися сервісом, постачальник має забезпечити наступні властивості та складові сервісу:

Виявлення сервісу потенційними користувачами повинно забезпечуватись постачальником через публікацію опису його сервісу на сайтах, загальнодоступних реєстрах, адресного інформування або іншим чином.

Зацікавленість передбачає, що сервіс виконує потрібну функціональність потенційним користувачам на прийнятних умовах, які викладаються в його описі.

Доступність передбачає, що сервіс забезпечує взаємодію через спільне з потенційними користувачами комунікаційне середовище (як правило, інтернет) за узгодженими протоколами (зазвичай, розповсюджені стандарти) а також адресою, які визначені в описі, та готовий виконувати запити на прийнятних умовах, які викладаються в описі (як правило, 24/7).

Інтерфейс визначає формат і структуру даних вхідного та вихідного повідомлень, на основі яких сервіс виконує визначену функціональність, а застосування-клієнт отримує результати виконання, на основі яких будує логіку своїх процесів. Для цього крім формату та структури даних, що передаються, обидві сторони повинні однаково інтерпретувати інформацію, якою обмінюються. Тобто використовувати однакову семантику (онтологію) даних, наданих у повідомленнях і яка має бути надана в описі сервісу.

Функціональність провайдер реалізує сервіс на власний розсуд, враховуючи потреби потенційних користувачів і надає відомості про результати виконання сервісу, не вдаючись у деталі «як саме він це робить». Користувач зі свого боку використовує сервіс як «чорну скриньку» для задоволення власних потреб, надаючи дані для виконання завдання та отримуючи потрібний результат через опублікований ін-

терфейс. Водночас сервіс може здійснювати додаткову обробку, яка не опубліковується і не є суттєвою для використання сервісу потенційним користувачем, але необхідна для підтримки стану об'єкта та роботи сервісу, якими він опікується. Для правильного використання сервісу крім структури та семантики повідомлень клієнт має бути обізнаний про дії, які виконує сервіс та їх вплив на стан спільних об'єктів споживача та постачальника.

Умови використання публікуються в описі сервісу та визначають підстави надання доступу (комерційна основа, виробничі взаємовідносини, вільний доступ).

QoS визначає якість надання послуг (швидкість відклику, обсяги, надійність, вартість тощо), які повинні додаватися до опису сервісу оптимального вибору потрібного користувачу сервісу серед наявних пропозицій.

Базова модель СОА визначає загальні принципи побудови сервіс-орієнтованих систем, конкретні реалізації котрих можуть відрізнятися стандартами взаємодії із сервісами, які використовуються, топологією взаємозв'язків, типами інтерфейсів тощо.

Зокрема, розрізняють дві топологічні архітектурні моделі композиції сервісів для реалізації бізнес-процесів:

Оркестровка використовує центральний керуючий сервіс, який, подібно до диригента в оркестрі, координує виконання окремих сервісів, що реалізують частину складного процесу, викликаючи їх у потрібній черзі. Центральний процес організує логіку виконання складного процесу, контролюючи результати виконання окремих кроків та формуючи послідовність викликів потрібних сервісів. Остання дія залежить від результатів виконання попередніх для забезпечення виконання композитного сервісу.

Хореографія не має централізованого контролю над процесом подібно до команди танцюристів, кожний з яких знає своє завдання та взаємодіє з партнерами на основі визначених правил.

Складність процесів, які реалізуються прикладним застосуванням в монолітній архітектурі обмежено можливостями комп'ютера, на якому воно виконується і, як правило, не виходить за межі процесів одного або декількох підрозділів підприємства. Тому для забезпечення складних процесів у межах всього підприємства потрібно інтегрувати окремі монолітні застосування, що автоматизують окремі ланки таких процесів, з виконанням їх у відповідності з встановленими бізнес-правилами. Також має забезпечуватися автоматична передача даних між ними на засадах розподіленої сервіс-орієнтованої архітектури.

Для інтеграції окремих компонент у межах однієї організації застосовуються різні методи (архітектурні шаблони) та відповідні інструменти. Одним із перших таких шаблонів можна визначити корпоративну сервісну шину (ESB), яка підтримує обмін даних між різнорідними застосуваннями в режимі реального часу. Організації можуть підключати застосування до централізованої шини, досягаючи мінімальних витрат на реалізацію інтерфейсу, використовуючи протоколи та компоненти платформи-застосування. ESB у свою чергу пропонує централізовану платформу, яка стандартизує та надає сервіси для перетворення різних форматів даних, протоколів передачі даних, маршрутизації повідомлень на основі онтологічного метаопису застосувань та їхньої взаємодії . Це спрощує організаціям обслуговування та масштабування своїх застосувань та керування ними.

З комерційної точки зору розрізняють різні типи сервісів:

Внутрішні сервіси - такі, що підтримуються та використовуються організацією для виконання власних процесів.

B2B (buisines to buisines) – сервіси, що підтримуються та надаються партнерами для забезпечення взаємодії в спільних процесах.

B2C (buisines to consumer) – сервіси, що підтримуються організацією та надаються клієнтам для взаємодії з бізнесом, в тому числі для надання послуг, зокрема і на комерційній основі.

G2B (government to buisines) – сервіси, що підтримуються державою та нада-

ються суб'єктам господарювання для надання адміністративних послуг.

На сьогодні використання корпоративних сервісних шин (ESB) обмежено в основному застарілими системами, що потребують взаємодії компонент за різними протоколами. Розвиток Інтернету та всеосяжне впровадження WEB-протоколів передачі даних практично витіснило інші протоколи. Тому складна архітектура з громіздкою централізованою шиною, через яку взаємодіють монолітні прикладні застосування, повсюдно замінюється архітектурами на основі WEB сервісів, а монолітні застосування декомпозуються на ряд слабко зв'язаних легких застосувань – мікросервісів, які взаємодіють один з одним через API (програмний інтерфейс). Клієнт-серверні десктоп-застосування, які поєднують бізнес логіку та користувацький інтерфейс витісняються триланковими застосуваннями з організацією взаємодії з користувачем через WEB інтерфейс за допомогою браузера. Це спрощує доступ до прикладних систем та дозволяє розповсюджувати та надавати програмне забезпечення як послугу (сервіс) не тільки через програмний інтерфейс, а й через інтерфейс користувача.

Завдяки такій архітектурі значно спрощується масштабування системи та її гнучкість. Автономні сервіси можна легко переміщувати з одного сервера на інший, поєднувати сервіси в єдину систему еволюційним шляхом у процесі експлуатації без необхідності перезбирати монолітне застосування під час зміни або додавання функціональності.

Впровадження не потребує повного переналаштування процесів та надає можливість використовувати звичні, добре зарекомендовані застосування. Достатньо лише додати відповідний інтерфейс для вже готової функціональності.

Мікросервісна архітектура дає підприємству можливість швидше адаптуватись до змін бізнесу, замінюючи реалізацію сервісу, залишаючи незмінним його інтерфейс. Використання стандартних протоколів взаємодії дозволяє поєднувати компоненти, розроблені на різних платформах та різними мовами.

Сервіс-орієнтована архітектура пропонує розробнику новий підхід до використання готових рішень без необхідності вбудови його коду в монолітне застосування. Замість цього надається композиція складних сервісів із готових розгорнутих сервісів, що реалізують частини процесу. Одночасно сервіси можуть бути розподілені в мережі і навіть належати різним компаніям та надаватись як послуги, не потребуючи розгортання та підтримки.

Розподілена сервісна архітектура прикладних застосувань суттєво відрізняється від монолітних застосувань, які виконуються в межах адресного простору одного процесора та взаємодіють через спільну оперативну пам'ять та базу даних.

Забезпечення взаємодії окремих компонент сервіс-орієнтованої системи (сервісів), що виконуються на різних процесорах та використовують власні бази даних для зберігання своїх даних, потребує інших архітектурних підходів, заснованих на обміні повідомленнями через мережу. Залежно від потреб організація обміну повідомленнями може здійснюватися синхронно або асинхронно.

Синхронний запит/відповідь передбачає відправку клієнтом запиту видавцю сервісу та очікування відповіді від видавця. Виконання клієнта блокується на час очікування та його виконання продовжується після отримання відповіді

Асинхронний запит/відповідь передбачає відправку клієнтом запиту без очікування відповіді. Клієнт не блокується на час очікування, а обробка відповіді здійснюється обробником події після надходження відповіді.

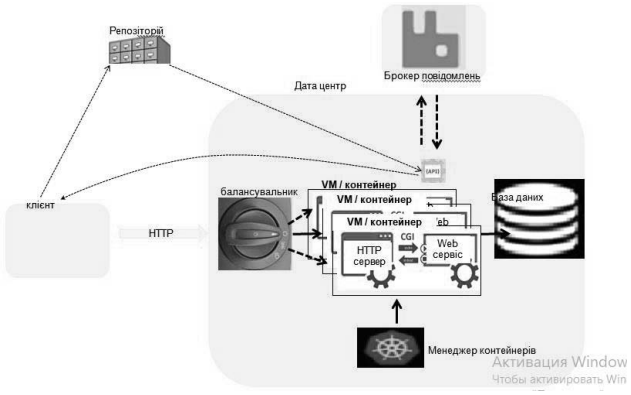


Рис. 3. Основні компоненти СОА

Взаємодія здійснюється через віддалений виклик (RPI- remoute procedure invocartion) сервісу через транспортний протокол мережі, переважно через посередництво HTTP сервера, який бере на себе багатопоточне прослуховування мережевого каналу приймання вхідного повідомлення та передачу його на виконання сервісу та відправлення результату виконання. Крім того, асинхронна взаємодія може здійснюватись через посередництво так званих брокерів.

Взаємодія через віддалений доступ обумовлює жорстку залежність від сервісу видавника та вимагає його постійної готовності, що не завжди можливо, оскільки завжди існує ризик часткової відмови на запит, який може бути обумовлений збоями обладнання та мережевого каналу або недоступністю сервісу у зв'язку з технічним обслуговуванням. Крім того, сервіс може бути перевантаженим та не відповідати на запити вчасно.

Тому СОА-система повинна передбачати відповідні заходи для запобігання таких ситуацій. Для цього можуть бути використані наступні шаблони:

Мережевий час очікування. Повторювання запиту у разі перевищеня певного часу очікування, що може запобігти впливу збоїв.

Обмеження кількості запитів. Після обумовленої кількості невдалих запитів, блокує ймовірно недоступний сервіс.

Запобіжник перевіряє доступність сервісу контрольним викликом через певний час та за потреби знімає блокування.

Виклик сервісу здійснюється за його мережевою адресою, яка може змінюватись у випадку фізичного переміщення сервісу, що призводить до відмови його клієнтів та потребує відповідного корегування адреси в коді клієнта, його перезбирання та перевстановлення. В разі використання сервісів на власних ресурсах ситуація може бути полегшена використанням конфігураційних файлів, які корегуються під час переміщення сервісів, що допомгає уникнути необхідності перезбирання клієнтів. Однак адреси екземплярів зовнішніх сервісів під контролем інших організацій а також сервісів, розміщених у хмарних середовищах, можуть мінятися динамічно. Більше того, набір цих екземплярів може постійно мінятися через постійне масштабування, відмови та оновлення, що потребує засобів динамічного виявлення актуальних адрес сервісів. Механізм динамічного виявлення сервісів заснований на використанні реєстру, здійснюється адміністратором або автоматично самим сервісом вбудованими засобами. Функції такого реєстру може виконувати, зокрема, звичайний DNS сервер.

Такий механізм виявлення дозволяє легко масштабувати систему у разі перевантаження певних сервісів та динамічно балансувати навантаження. Якщо кількість запитів до сервісу починає перевищувати його можливості, запускається ще один або кілька екземплярів, які реєструються в реєстрі, й запит спрямовується до якогось із них, використовуючи одну із стратегій балансування:

Round Robin - обслуговування по колу, за якого кожний наступний запит передається наступному зареєстрованому в реєстрі екземпляру;

Weighted Round Robin - враховує продуктивність екземпляру, визначену ваговим коефіцієнтом;

Least Connections - враховує кількість підключень до екземплярів на даний момент;

Destination Hash Scheduling і Source Hash Scheduling Sticky Sessions - враховує зони мережевого розташування клієнта на основі IP-адреси.

Механізм взаємодії сервісів через RPI потребує активності обох компонент, чого не завжди можна досягти. Цього недоліку можна уникнути в архітектурі на основі обміну повідомленнями через посередництво брокерів подібно архітектурі ESB. Згідно із цим механізмом сервіс - відправник записує повідомлення в канал, а отримувач зчитує його з цього каналу. Під час відправки повідомлення активність отримувача не обов'язкова, оскільки повідомлення зберігається в каналі доки отримувач його не прочитає та не опрацює.

Повідомлення складається із заголовку, який містить метадані щодо повідомлення, включаючи ідентифікатор пові-

домлення та зворотну адресу, що визначає канал, куди потрібно спрямувати повідомлення-відповідь, та тіло повідомлення, яке містить власне дані в узгодженому форматі (текстовому або двійковому).

За специфікою обробки повідомлень розглядають два типи каналів:

- «крапка — крапка» - використовуються для доставки повідомлення тільки одному отримувачу, який зчитує їх із цього каналу та обробляє його. Як правило, через такі канали передаються повідомлення типу «команда»;
- «видавець — підписник» доставляє кожне повідомлення всім підключеним споживачам. Зазвичай через такі канали передаються повідомлення типу «подія».

Взаємодія за допомогою повідомлень за своєю природою асинхронна, оскільки після відправлення повідомлення сервіс не очікує відповіді, а сам ініціює його отримання з відповідного каналу, використовуючи ідентифікатор повідомлення наданого в повідомленні запиту разом з каналом відправника. Крім того, на відміну від віддаленого виклику, відповіді можуть оброблятися будь - яким екземпляром сервісу, незалежно від того, який екземпляр здійснив запит.

API асинхронного сервісу, який використовує обмін повідомленнями крім операцій включає опис подій, які він публікує в каналах типу видавець-підписник.

Також, як і для віддаленого виклику сервісу взаємодія через повідомлення потребує знання дислокації «каналів» в мережі та протоколів доставки повідомлень. Тому повинен використовуватись механізм виявлення, але не для адреси сервісу, а для його каналів. Також, як і під час віддаленого виклику, потрібно гарантувати доступність «каналу», що може бути проблематичним у разі великого його завантаження. З цими проблемами можна впоратися успішніше, використовуючи не пряму відсилку повідомлень в канал, а через посередництво спеціального сервісу-брокера. Він не аналізує тіло запитів, а лише виконує приймання повідомлень та їх збереження в своїй базі даних і видачу повідомлень на запити споживачів.

Для виконання запиту клієнта достатньо взаємодіяти лише з брокером та відправляти повідомлення у відповідний канал. Клієнту не потрібно використовувати механізм виявлення сервісу та турбуватись про масштабування.

Брокер буферизує повідомлення доти, доки вони не зможуть бути оброблені, і взаємодія сервісів в системі буде здійснюватись навіть у разі тимчасової недоступності сервісу-виконавця.

Потенційно вузьке місце продуктивності централізованого обробника повідомлень долається завдяки відсутності прикладної обробки повідомлення, а лише його збереженням в БД а також через балансування навантаження сервісу-брокера.

Брокер може протоколювати процес обміну повідомленнями, авторизації під час доступу до каналів та виконувати інші функції, пов’язані з обміном.

Залежно від організації зберігання повідомлень, розрізняють два типи брокерів:

Брокер повідомлень - сервіс, який приймає повідомлення від клієнта та зберігає його в одній або декількох чергах видавництв сервісів доти, доки сервіс видавця не забере повідомлення зі своєї черги та не виконає відповідні дії, результатом яких може бути розміщення відповіді в черзі клієнтського сервісу.

Брокер подій - використовує парадигму видавця-підписника. При цьому в брокері повідомлень кожний сервіс не має власної черги. Замість цього клієнти розміщують повідомлення в базі брокера в чергах відповідно до типу повідомлення, на яке підписуються сервіси-обробники, які отримують ці повідомлення від брокера подій на основі адреси, наданої під час підписки або шляхом запитів до брокера.

В сервіс-орієнтованому застосуванні окремі сервіси мають «власну модель» БД, тому потребують підтримки розподілених транзакцій. Механізм двофазної фіксації (2PC), яку використовують в монолітних застосуваннях під час роботи з декількома базами даних передбачає жорстку взаємозалежність та доступність сервісів у процесі виконання розподіленої тран-

закції, що суперечить принципу слабкої зв'язаності сервісів в СОА.

Згідно з відомою CAP теоремою Брюєра [8] неможливо одночасно забезпечити: узгодженість даних в усіх вузлах (Consistency), доступність вузлів(Availability), стійкість до розділення (Partition tolerance). Оскільки в сервіс-орієнтованому застосуванні апріорі підтримується Partition tolerance, неможливо забезпечити Consistency або Availability, які потрібні для 2PC, тому для підтримки розподілених транзакцій в СОА застосовується інший архітектурний шаблон «оповідання» «SAGA». Цей шаблон забезпечує узгодженість даних, користуючись послідовністю прямих та компенсуючих транзакцій, які координуються асинхронними повідомленнями.

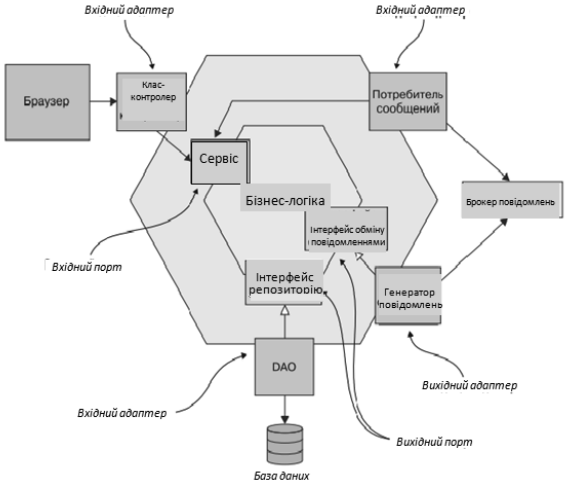


Рис. 4. Архітектура сервісу в СОА

Архітектура сервісів відрізняється від пошарової архітектури (MVC, MVP, MVPP) монолітних застосувань. Сервіси в СОА зазвичай реалізують бізнес логіку операцій пов'язаних з одним суб'єктом, відстежують його стан у своїй автономній БД та виконують свій функціонал за запитами через API. Тому архітектурно сервіс має не пошарову архітектуру, а так звану шестигранну [9] з ядром (рис 2), яке виконує бізнес-логіку, та адаптерами, які забезпечують взаємодію. Механізм віддаленого виклику сервісу заснований на використанні програмного інтерфейсу (API) сервісу, який визначає перелік операцій, що можуть викликатись, та подій, що можуть генеруватись під час виконання. Для кожної операції визначаються формат та семантика даних, необхідних для її виконання, а також формат та семантика результату виконання операції. Для події визначається її тип, формат та семантика пов'язаних даних, що публікуються в каналі взаємодії. Клієнтська бізнес-логіка для віддаленого виконання операції зовнішнім сервісом звертається до проксі-інтерфейсу – класу адаптера, який інкапсулює узгоджений API. RPI-проксі формує та відправляє через мережевий протокол (як правило HTTP) повідомлення-запит сервісу. Запит приймається класом-адаптером RPI-сервер, який витягує з повідомлення дані відповідно до API, та викликає бізнес-логіку сервісу . Після відпрацювання бізнес-логіка сервісу викликає метод RPI-серверу, який запаковує результат виконання відповідно до API, та відправляє відповідь через мережевий протокол, який приймається методом RPI-проксі. А той витягує дані результату та викликає обробку результату клієнтською бізнес-логікою. Водночас надсилання HTTP запиту клієнтом може здійснюватися синхронно так, що процес клієнта очікує відповідь сервісу, або асинхронно таким чином, що процес клієнта продовжується, а обробка результату здійснюється методом–обробником події приходу результату HTTP-запиту, який вказується у надсиланні запиту.

Завдяки цьому внутрішні зміни бізнес-логіки операцій не впливають на клієнтів, якщо не змінюється API.

Стандарти сервіс-орієнтованої парадигми

Сервіс-орієнтована архітектура прикладних систем покладається на взаємодію її окремих компонент, що функціонують на різних платформах в різних обчислювальних середовищах, підпорядкованих різним організаціям через повідомлення по мережі. Це в свою чергу потребує узгоджених правил (протоколів) прийому/передачі даних, їхніх форматів як на фізичному, так і на логічному рівні, які підтримуються різними платформами, а також метаопису компонент та їхніх інтерфейсів.

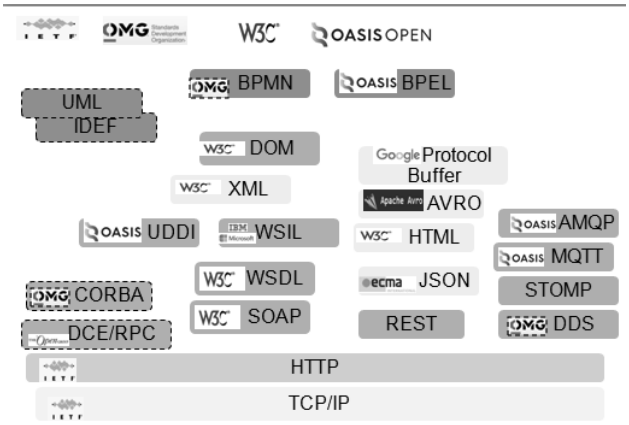


Рис. 5. Стандарти СОА

Для впорядкування досвіду та досягнення сумісності продуктів підтримки локальних мереж від різних виробників, які стрімко почали розвиватись, міжнародна організація із стандартизації OSI (Open system interconnection) розробила базову модель мережевої взаємодії, яку офіційно опублікувала 1983 року. Паралельно в процесі розвитку глобальних мереж Internet Інженерною радою Інтернету IETF (Internet Engeniering Task Force) була сформована альтернативна модель TCP/IP і відповідний стек протоколів, які поступово стали домінуючими, яка складалась із 4 шарів (замість 7 рівнів OSI):

Канальний рівень (network access layer) визначає параметри фізичного середовища передачі та спосіб кодування даних.

Міжмережевий рівень (Internet layer) регулює передачу даних з однієї мережі в іншу. На цьому рівні працюють маршрутизатори, які спрямовують пакети до потрібної мережі за її адресою.

Транспортний рівень визначає механізми перевірки правильності пересилки та доставки пакетів потрібному застосуванню в правильному порядку з підтвердженням доставки (TCP) або без підтвердження (UDP).

Прикладний рівень забезпечує узгоджені правила та формати обміну інформації між прикладними застосуваннями. Залежно від призначення прикладного застосування можуть бути різні протоколи прикладного рівня. Наприклад, протокол HTTP для інтернет-браузерів, протокол FTP - для передавання файлів, протоколу, SMTP для електронної пошти, в тому числі усі стандарти, що регламентують сервісно-орієнтовану взаємодію.

Протоколи прикладного рівня для взаємодії прикладних застосувань через мережу почали з'являтися із 1970-х років. Зокрема, стандарт DCE/RPC визначив механізм віддаленого виклику процедури, що виконується в іншому адресному просторі, з передачею необхідних параметрів та отриманням результатів через мережу, аналогічно передачі параметрів по значенню через стек ОП в монолітному застосуванні.

Пізніше компанія Microsoft розповсюдила свою компонентну технологію COM взаємодії різних об'єктних застосувань в одному адресному просторі (так званий механізм OLE) на розподілені системи DCOM, додавши механізм віддаленого виклику методів об'єкта, існуючого в застосуванні, що виконується на іншому комп'ютері (сервері) з маршалінгом (передачею даних між клієнтом та сервером через мережу) на тих же засадах, що і віддалений виклик процедур (RPC), використовуючи метаопис інтерфейсу методів об'єктів спеціальною мовою опису інтерфейсу (Interface Definition Language). Однак ця технологія та стандарт дозволяє поєднувати тільки застосування під операційною системою Windows.

Для формування кросплатформених механізмів взаємодії прикладних застосувань групою OMG (Object Management Group), до складу якої входять представники найбільших виробників програмного забезпечення, було розроблено технологію та відповідний стандарт CORBA (Common Object Reqest Broker). Технологія заснована на використанні спеціальної мови опису інтерфейсу (IDL-Interface Definition Language) з об'єктами, які виконуються на серверах розподіленої системи та проблемно-незалежних компонент (ORB core), які забезпечують створення/виявлення екземплярів об'єктів та віддаленого виклику їхніх методів, використовуючи програмні адаптери (Stubs/Sceleton), які генеруються автоматично за описом інтерфейсу мовою IDL.

Водночас швидко розвивається Всесвітня павутина WWW, заснована на доступі до текстової та графічної інформації на основі обміну контентом у форматі HTML між браузером та сервером через протокол прикладного рівня HTTP. Завдяки цьому підтримка протоколу HTTP вбудована практично в усі операційні системи та платформи.

Платформонезалежна базова модель структурованого документа (DOM), запропонована консорціумом W3C, що опікується стандартами WEB, та призначена для роботи з документами мовою розмітки документа HTML отримала реалізації практично усіма мовами програмування та широко використовувалась не тільки в браузерах для роботи з HTML-документами, а й в прикладних застосуваннях для роботи з документами в форматі розширюваної мови розмітки XML, також запропонована консорціумом для текстового представлення структурованих даних опису об'єктів.

Тому компанією Microsoft 1998 року було запропоновано та передано консорціуму W3C для подальшого супроводу та розповсюдження спрощений протокол SOAP (Simple Object Access Protokol) взаємодії прикладних застосувань через Інтернет на основі віддаленого виклику процедур обміну даними в текстовому форматі XML. Цей протокол було доповнено ще двома взаємопов'язаними специфікаціями, що забезпечують процеси публікації прикладних застосувань в Інтернет-середовищі (Вебсервісів) та їх використання:

SOAP – визначає порядок виконання обміну повідомленнями, синтаксис та основну структуру повідомлення, правила їхнього формування та інтерпретації у відправника та приймальника.

WSDL - визначає формальний опис програмного інтерфейсу (API) WEB сервісу, який по суті є XML схемою SOAP повідомлень за правилами XSD

UDDI - визначає стандартний API обмін з реєстром та XML-схему онтологічного опису SOAP WEB сервісів, в якому постачальники сервісів можуть розміщувати інформацію про себе, сервіси, які вони забезпечують, та порядок їх використання, в тому числі посилання на WSDL специфікацію сервісу.

Крім цієї трійки стандартів, регламентуючих основні етапи взаємодії SOAP сервісів, W3C визначила низку так званих WS-* XSD специфікацій API відносно різних аспектів організації взаємодії між сервісами – (автентифікація, безпека, розподілені транзакції тощо).

Протокол SOAP покладається на формат XML, обмежує його застосування, обумовлене накладними витратами у передаванні даних в інших форматах, більш пристосованих для конкретної прикладної галузі застосування сервісів. Зокрема, одним з найпоширених споживачів сервісів є WEB застосування, які працюють у середовищі браузерів та викликають сервіси скріптами мовою JAVAscript, прийнятої за стандартну, та яка має підтримку практично в усіх браузерах. У стандарті цієї мови визначений інший формат текстового уявлення структурованих даних JSON, відмінний від XML, та більш компактний, що ускладнює використання SOAP сервісів у WEB застосуваннях.

Тому 2000 року Рой Філдинг в своїй дисертації [6] виклав основні принципи технології взаємодії застосувань через інтернет на основі прямого використання протоколу HTTP без обмеження на формат передачі даних.

Ця технологія або архітектурний стиль отримала назву REST (REpresentational State Transfer). І, хоча не отримала «офіційного» статусу стандарту, але швидко стала домінуючою в галузі сервіс-орієнтованих застосувань завдяки забезпеченню корисних властивостей:

- широке розповсюдження базового HTTP протоколу;
- надійність (через відсутність необхідності зберігати стан клієнта під час взаємодії);
- продуктивність (через використання кешу);
- масштабування (через використання посередників – брокерів та балансувальників);

 простота інтерфейсів (відсутність необхідності підтримки формального опису).

Відсутність обмежень на формат повідомлень дозволяє використовувати будь-який найбільш доцільний формат повідомлень, виходячи з призначення сервісу, в тому числі той самий XML, як в SOAP. Однак більшої популярності набув формат JSON за рахунок своєї лаконічності в порівнянні з XML (що забезпечує більшу продуктивність через зменшення трафіку) та підтримці в JavaScript (вбудованої мови браузерів). Крім того, він дозволяє використання бінарних форматів, таких як Avro або Protocol Buffers, що здатні в деяких випадках ще більше скоротити трафік при взаємодії.

Асинхронна взаємодія застосувань передбачає застосування спільного сховища або брокера для тимчасового зберігання повідомлень. Водночас постачальники та споживачі повідомлень мають використовувати узгоджені протоколи та формати, що дозволяють виявити призначені їм повідомлення в спільному середовищі та обробити їх. Для цього авторитетні організації із розробки стандартів СОА пропонують протоколи підтримки асинхронної взаємодії за різними схемами, такими як крапка – крапка, або публікація/підписка:

AMQP – Advancing Message Queuing Protocol,

MQTT – Message Queuing Telemetry Transport,

STOMP – Simple/Streaming Text Oriented Protocol,

DDS- Data Distribution Service.

Розробка сервіс-орієнтованих застосувань покладається в основному на композицію процесів із готових рішень, що реалізують окремі операції – частини процесу. Для цього потрібно спочатку декомпозувати бізнес-процеси, що автоматизуються на окремі операції, визначити потрібні сервіси, які можуть їх виконати та знайти готові рішення, або розробити нові в разі відсутності потрібних. Така робота потребує методів моделювання процесів, зрозумілих як системним аналітикам–фахівцям у предметній галузі, так і програмістам, що їх реалізують. Для проєктування монолітних застосувань використовувались відповідні графічні нотації, що дозволяють наочно представити склад операцій їхньої взаємозв'язки та послідовність виконання. В структурних застосуваннях це нотації IDEF в об'єктно-орієнтованих - UML. Сервіс-орієнтовані застосування часто використовують подіє-орієнтовані механізми взаємодії, які не підтримуються в цих нотаціях. Тому групою OMG було розроблено нотацію BPMN (Buisines Process Modeling Notation), яка усуває цей недолік. Крім графічного представлення ця специфікація визначає еквівалентний формальний опис процесів мовою XML відповідної схеми.

Якщо в таку модель додати специфікацію реалізацій та інтерфейсів складових сервісів, а також умов, що визначають послідовність їх виконання, то таку модель можна інтерпретувати для виконання процесу. Таку специфікацію BPEL (Buisines Process Execution Language) для оркестрування SOAP сервісів, засновану мовою XML, також було запропоновано IBM та передано в OASIS для супроводу.

Програмна інженерія сервіс-орієнтованої парадигми

Технологія розробки, впровадження та супроводу сервіс-орієнтованих систем суттєво відрізняється від розробки систем монолітної архітектури через їхню розподілену природу. Сервіс-орієнтований підхід починає застосовуватись тоді, коли предметна галузь системи перетинає межі організацій та підрозділів та її об'єм починає перевищувати наявні обчислювальні можливості комп'ютерів а супровід стає погано керовану через необхідність реагування на зміни великої кількості процесів, поєднаних в монолітному застосуванні.

Для виконання процесів сервіс-орієнтовані системи використовують компоненти, які знаходяться в зоні відповідальності різних підрозділів і навіть організацій, і тому потребують узгодження програмних інтерфейсів для інформаційної взаємодії та послідовності їх виконання і відповідно проєктування та декомпозиції складних процесів.

Декомпозиція сервіс-орієнтованої системи на окремі компоненти – сервіси заснована на предметно-орієнтованому проєктуванні, згідно з яким до функціональності сервісу залучаються операції, пов'язані з відносно самостійною групою ресурсів, що гарантує слабку зв'язаність сервісу.

В основу декомпозиції системи на сервіси можна покласти принципи єдиної відповідальності (Single Responsibility Principle, SRP) та узгоджених змін (Common Closure Principle, CCP), запозичені з об'єктно-орієнтованого проєктування.

Згідно із принципом SRP кожен сервіс повинен мати лише одне призначення і відповідно єдину причину для зміни, а згідно із принципом CCP причини зміни операцій сервісу мають бути однакові.

Дотримання цих принципів декомпозиції дозволить мінімізувати кількість сервісів, які потрібно буде редагувати та розгортати зі зміною якоїсь вимоги.

Окремі операції спроєктованої системи можуть бути вже реалізовані різними провайдерами в сервісах, розгорнутих на їхніх ресурсах та доступних для використання . Щоб скористатися таким готовим сервісом, клієнт повинен бути обізнаним із функціоналом, програмним інтерфейсом (API), адресою сервісу та умовами надання. А для цього провайдер сервісу повинен опублікувати такий опис на ресурсах, доступних потенційним користувачам.

Для SOAP WEB сервісів був запропонований стандарт UDDI онтологічних мета описів сервісів та інтерфейс доступу до них для створення як публічних, так і відомчих репозиторіїв. Цей стандарт був відтворений у різних реалізаціях як репозиторіїв, так і їхніх клієнтів, що забезпечували публікацію мета-описів сервісів, пошук сервісів за елементами специфікації, отримання опису API.

Однак згодом Rest сервіси значно потіснили SOAP, тому UDDI репозиторії не виправдали очікуваного поширення. 2006р. IBM, Microsoft та SAP оголосили про закриття своїх загальнодоступних UDDI вузлів. 2007 року завершена підтримка UDDI в OASIS. 2010 року Microsoft оголосила про припинення підтримки UDDI в Windows Server. Тому пошук потрібних сервісів та їхніх постачальників може здійснюватися в основному за допомогою загальних пошукових механізмів інтернету на основі ключових слів, а отримання опису їх API здійснювати зі знайденого сайту. Питання створення репозиторіїв для забезпечення пошуку потрібних сервісів за їхнім онтологічним описом залишається відкритим. Водночас онтологічні схеми в таких репозиторіях не повинні обмежуватись одним стандартом та передбачати в своєму описі визначення протоколів та форматів взаємодії. Додавання в онтологію опису крім синтаксичного опису API ще семантики функціональності та повідомлень сервісу та існуючих екземплярів може розширити призначення репозиторію не тільки для пошуку потрібних сервісів, а й для виявлення потрібних сервісів в процесі виконання з балансуванням навантаження та оптимізацією витрат. Доцільно розглянути підхід створення розподілених репозиторіїв на базі WSIL (Wev Servcies inspection language), запропонований сумісно IBM та Microsoft.

Композитні сервіси, що реалізують поєднання окремих сервісів для виконання складних процесів відповідно до спроектованої бізнес-логіки, можуть розроблятися на будь-якій платформі з використанням бібліотек підтримки стандартних протоколів взаємодії . Для організації асинхронних відкладених викликів сервісів можна використовувати готові брокери повідомлень. Крім того, для композиції сервісів, спроєктованих у стандартних нотаціях (BPMN, BPEL), можна реалізовувати за допомогою спеціалізованих платформ та інтерпретаторів, що їх підтримують.

Деякі операції можуть потребувати втручання людини для вводу даних, необхідних для виконання сервісу та аналізу результатів його виконання для ухвалення рішень.

Такі операції реалізуються здебільшого WEB застосуванням, побудованим за AJAX технологією.

Водночас інтерфейс користувача для браузера формується за допомогою

спеціалізованих фреймворків, а для виклику сервісів використовуються засоби підтримки HTTP протоколу, присутні практично на усіх платформах.

Якщо для операцій не знайшлося готової реалізації, потрібно створити відповідні компоненти сервіси. Окремий сервіс розробляється як неінтерактивне монолітне застосування, яке відстежує стан ресурсів в окремій базі даних та видачу цього стану за запитами через програмні інтерфейси.

Сервіс розробляється, переважно, в об'єктно-орієнтованій парадигмі з використанням бібліотек підтримки протоколів та стандартних форматів повідомлень для обраної платформи розробки.

Прослуховування мережевого каналу, прийом та відправку повідомлень, як правило, делегується HTTP-серверу, взаємодія з яким здійснюється за обумовленими протоколами (CGI, JavaServlet …), підтримка яких вбудована у відповідні бібліотеки.

Методологія структурного проєктування та програмування SADT , що застосовувалась для розробки складних монолітних систем, рано чи пізно стикається з проблемою росту термінів випуску версій системи у зв'язку з неконтрольованим зростанням кількості залежностей. В результаті система не взмозі реагувати на зміни бізнес правил та умов виконання процесів у прийнятні терміни.

Розробка складних систем у сервіс-орієнтованій архітектурі взмозі швидко реагувати на зміни бізнес правил та умов використання за рахунок того, що вона складається з відносно невеликих частин. Команда ж розробників сервісу може бути невеликою та розвивати й підтримувати сервіс у разі зміни внутрішніх правил виконання операцій сервісу незалежно від інших пов'язаних сервісів, доки зовнішній узгоджений програмний інтерфейс залишається незмінним. Однак зберегти API вдається не завжди, тому для забезпечення працездатності усієї СОА-системи у разі заміни версії сервісу, потрібно забезпечувати сумісність версій або підтримку всіх версій API, доки вони використовуються його клієнтами. Подолати такі негаразди допомагає специфікація семантичного версіонування, згідно з якою номер версії формується з трьох частин, які інкрементуються під час випуску нової версії відповідно до характеру змін:

Major – під час внесення в API несумісних змін (видаляються параметри або змінюється їхня семантика).

Minor – під час зміни API з підтримкою зворотної сумісності (додавання нових атрибутів до запиту або відповіді; додавання нових операцій. Водночас сервіси повинні надавати значення за замовченням для нових атрибутів, а клієнти можуть ігнорувати будь-які зайві атрибути).

Patch - виправлення помилок з підтримкою зворотної сумісності.

Відповідність версії API сервісу та клієнта здійснюється на основі номера версії, що вказується в повідомленні. Вносячи мажорні зміни, важливо підтримувати і попередні версії API, для чого номер версії може бути частиною адреси сервісу.

Така методологія швидкого реагування на зміни бізнес-правил та постійного вдосконалення системи отримала назву екстремального програмування (AGILE), заснованого на 12 простих принципах [10], одним з яких є принцип тісної взаємодії розробників та користувачів системи (DEVelop & OPerationS).

Для дотримання цих принципів максимально автоматизується взаємодія розробників та користувачів, а також операції технологічного циклу випуску версій сервісів із застосуванням відповідних засобів для планування модифікацій та їх реалізації, тестування та розгортання.

Висновки

Сервіс-орієнтована парадигма побудови складних розподілених систем є закономірним результатом еволюції підходів до розробки від процедурних та об'єктно-орієнтованих монолітних систем в єдиному адресному просторі до взаємодіючих мережею слабко зв'язаних компонентів у розподіленому середовищі.

Сервіс-орієнтована парадигма уможливлює створення прикладних систем шляхом інтеграції готових компонент

в єдиний технологічний ланцюг без необхідності поєднання їх в монолітне застосування та розгортання на власних ресурсах. Використання їх, як є, на ресурсах провайдера на основі стандартів взаємодії та узгодженого програмного інтерфейсу (API) є практично доцільним. Rest API на основі HTTP та JSON домінує та практично витіснив рішення на основі стандартів взаємодії SOAP. Зокрема, загальні реєстри UDDI публікації онтологічних описів SOAP сервісів, які дозволяли здійснювати пошук готових сервісів, необхідних для компонування прикладної системи, стали неактуальними. Хоча потреба в пошуку готових рішень залишається. Тому створення аналогічних загальнодоступних реєстрів готових рішень з онтологією опису, не обмеженою протоколом SOAP, є актуальним завданням.

Функціонал, для якого не знайдено відповідного готового рішення розробляється як на традиційних компілюючих платформах, так і в сучасних інтерпретуючих платформах WEB програмування в об'єктно-орієнтованій парадигмі з дотриманням стандартів взаємодії SOAP та REST.

Відпрацьована формальна методологія SADT організації розробки монолітних застосувань витіснена більш гнучкою методологією екстремального програмування AGILE, яка значно скорочує тривалість технологічного циклу розробки/впровадження, та дозволяє швидше реагувати на зміни правил виконання бізнес процесів. Дотримання принципів екстремального програмування потребує використання спеціалізованих програмних засобів організації розробки включно із плануванням та відстеженням завдань, а також – автоматизацією тестування та розгортанням версій системи.

Література

1. А.Пискун, Краткий путеводитель по семейству нотаций IDEF. https://www.antonpiskun.pro/kratkij-putevoditel-po-semejstvu-notaczij-idef/
2. Donald Bell, UML basics: An introduction to the Unified Modeling Language. https://cs.nyu.edu/~jcf/classes/g22.2440-001_sp06/handouts/UMLBasics.pdf
3. С.Фейт, TCP/IP Архітектура, протоколи, реалізація. ISBN 5-85582-072-6 https://coollib.cc/b/163834-sidni-m-feyt-tcp-ip-arhitektura-protokolyi-realizatsiya-vklyuchaya-ip-versii-6-i-ip-security/read
4. Internet Engineering Task Force (IETF) Request for Comments: 7540 May 2015 Hypertext Transfer Protocol Version 2 (HTTP/2) https://datatracker.ietf.org/doc/html/rfc7540
5. D.Tidwell, J.Snell, P.Kulchenko, Programming Web Services with SOAP. https://theswissbay.ch/pdf/Gentoomen%20Library/Programming/CSharp/OReilly%20-%20Programming%20Web%20Services%20with%20Soap.pdf
6. R.Th.Fielding, Architectural Styles and the Design of Network-based Software Architectures, https://ics.uci.edu/~fielding/pubs/dissertation/top.htm
7. OASIS. Reference Model for Service Oriented Architecture 1.0 Committee Specification 1, 2 August 2006 https://groups.oasis-open.org/higherlogic/ws/public/download/19679/soa-rm-cs.pdf/latest
8. S.Gilbert, N.Lynch, Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services https://web.archive.org/web/20160213135959/http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.20.1495&rep=rep1&type=pdf
9. К.Ричардсон, Микросервисы: Патерны разработки и рефакторинга. Издательский дом "Питер", 2020. https://coderbooks.ru/mikroservisy-patterny-razrabotki-i-refaktoringa
10. Agile-маніфест розробки програмного забезпечення, https://agilemanifesto.org/iso/uk/manifesto.html

References

1. A.Piskun, A Brief Guide to the Notation Family IDEF. https://www.antonpiskun.pro/kratkij-putevoditel-po-semejstvu-notaczij-idef/
2. Donald Bell, UML basics: An introduction to the Unified Modeling Language. https://cs.nyu.edu/~jcf/classes/g22.2440-001_sp06/handouts/UMLBasics.pdf
3. S.Fate, TCP/IP Architecture, protocols, implementation. ISBN 5-85582-072-6

https://coollib.cc/b/163834-sidni-m-feyt-tcp-ip-arhitektura-protokolyi-realizatsiya-vklyuchaya-ip-versii-6-i-ip-security/read
4. Internet Engineering Task Force (IETF) Request for Comments: 7540 May 2015 Hypertext Transfer Protocol Version 2 (HTTP/2) https://datatracker.ietf.org/doc/html/rfc7540
5. D.Tidwell, J.Snell, P.Kulchenko, Programming Web Services with SOAP. https://theswissbay.ch/pdf/Gentoomen%20Library/Programming/CSharp/OReilly%20-%20Programming%20Web%20Services%20with%20Soap.pdf
6. R.Th.Fielding, Architectural Styles and the Design of Network-based Software Architectures, https://ics.uci.edu/~fielding/pubs/dissertation/top.htm
7. OASIS. Reference Model for Service Oriented Architecture 1.0 Committee Specification 1, 2 August 2006 https://groups.oasis-open.org/higherlogic/ws/public/download/19679/soa-rm-cs.pdf/latest
8. S.Gilbert, N.Lynch, Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services https://web.archive.org/web/20160213135959/http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.20.1495&rep=rep1&type=pdf
9. K.Richardson, Microservices: Patterns of development and refactoring. Publishing house "Piter", 2020. https://coderbooks.ru/mikroservisy-patterny-razrabotki-i-refaktoringa
10. Agile-software development manifesto, https://agilemanifesto.org/iso/uk/manifesto.html

Одержано: 12.04.2025
Внутрішня рецензія отримана: 19.04.2025
Зовнішня рецензія отримана: 28.04.2025

Про авторів:

[1]*Андон Пилип Іларіонович*,
академік НАН України.
https://orcid.org/0009-0000-1737-5579.

[1]*Суботін Сергій Васильович*,
науковий співробітник.
https://orcid.org/0000-0002-5958-0259.

[1]*Перконос Петро Іванович*,
науковий співробітник.
https://orcid.org/0000-0002-5958-0260.

[1]*Твердохліб Євген Миколайович*,
кандидат технічних наук,
старший науковий співробітник.
https://orcid.org/0000-0001-6594-5468.

Місце роботи авторів:

[1]Інститут програмних систем
НАН України,
тел. +38-044-522-62-42
E-mail: ukrprog@isofts.kiev.ua
www.iss.nas.gov.ua

УДК 004.056 https://doi.org/10.15407/pp2025.02.020

О.О. Летичевський, С.О. Євдокимов

ДВІЙНИКИ В СИСТЕМАХ ВИЯВЛЕННЯ ВТОРГНЕНЬ НА ОСНОВІ ГЛИБОКОГО НАВЧАННЯ

Дана робота спрямована на підвищення точності виявлення атак у програмних та апаратних системах шляхом використання цифрового двійника у формі алгебраїчної моделі в системах виявлення вторгнень (IDS), заснованих на нейронних мережах глибокого навчання (DNN). Цей підхід усуває недоліки навчання та недосконаліть набору даних, які призводять до численних помилкових спрацьовувань, невиявлених вторгнень та слабкої стійкості до атак супротивника. Пропонується архітектура IDS, яка поєднує нейронні мережі глибокого навчання з алгебраїчною моделлю на необхідному рівні абстракції. Ця композиція забезпечує високу точність виявлення та постійне самонавчання IDS на основі роботи моделі та збору даних, включаючи атаки нульового дня. Два приклади демонструють застосування цього підходу: виявлення атак у двійковому коді програмної системи та в програмованій інтегральній схемі.

Ключові слова: глибоке навчання, нейронна мережа, нейро-символьний підхід, цифрові двійники, алгебра поведінки

O.O. Letychevskyi, S.O. Yevdokymov

DIGITAL TWINS IN INTRUSION DETECTION SYSTEMS BASED ON DEEP LEARNING

This work aims to improve the accuracy of attack detection in software and hardware systems by utilizing a digital twin in the form of an algebraic model within intrusion detection systems (IDSs) based on deep learning neural networks (DNNs). This approach addresses the shortcomings of training and dataset imperfections that lead to numerous false positives, undetected intrusions, and weak resistance against adversarial attacks. We propose an IDS architecture that combines deep learning neural networks with an algebraic model at the required level of abstraction. This composition provides high detection accuracy and enables continuous self-learning of the IDS based on model operation and data acquisition, including zero-day attacks. Two examples demonstrate the application of this approach: detecting attacks in the binary code of a software system and in a programmable integrated circuit.

Keywords: deep learning, neural network, neurosymbolic approach, digital twins, behavior algebra

Introduction

The modern intrusion detection systems (IDSs), which are based on deep learning neural networks (DNNs), have benefits and shortcomings. The detection of attacks in real time is carried out by analyzing incoming traffic, from which it is possible to extract some traffic, statistical and time data.

The indisputable advantage of DNNs is that they can work in real time, quickly detecting intrusions. In general, IDSs can be divided into two functional groups [1].

The first variety refers to classifiers trained on data labeled with specific types of attacks or multiclassifications. Everything that does not fall under this dataset is considered benign traffic.

The second group is trained on examples of normal system operation, and any deviation is considered an intrusion. These systems are called anomaly-based systems.

Because datasets are not able to cover the full space of possible data, cases of false positives can arise. Moreover, a hacker can evade classification by manipulating sensitive features fed to the DNN input.

Modern intrusion detection systems work for both software and hardware systems, including those based on programmable

integrated circuits, namely field programmable gate arrays (FPGAs) that are very popular in the Internet of Things environment.

In software systems, IDSs are implemented in the form of firewalls that contain a neural component and block malicious traffic. Often, systems are created based on FPGA or application-specific integrated circuit (ASIC), which directly contain neural networks [2].

The internet protocols TCP/IP and UDP, from which the traffic is considered, are mostly encrypted data, so only available data from traffic packets and statistical and time data are generally used. It reduces the accuracy of classification as well.

There are a certain number of methods that attempt to eliminate such disadvantages, particularly the use of sandbox technology. However, running the application in an isolated environment significantly slows down traffic and cannot be used effectively in real time.

The authors have developed technology in which there is a digital twin [3] of the system to be protected. This notion is known as an established concept in various subject domains to study and analyze the original system. The digital twin is presented in the form of an algebraic model created at the appropriate level of abstraction for effective intrusion detection. We consider the architecture of the system, where both types of DNNs are involved, which interact with the digital twin.

The system also uses the augmentation of the dataset using the generation of new features via a digital twin. For this purpose, a limited initial run is performed in the digital twin environment to obtain the relevant data.

1. General scheme of the approach

The main idea behind the IDS approach proposed by the authors is to combine an algebraic method, which provides accuracy in detection, with a neural network of deep learning. This approach is considered neurosymbolic. This approach is used in practice and has already produced serious results that have significantly increased the accuracy of the DNN [4].

To implement the symbolic or algebraic component, a digital twin of the system is created in terms of algebraic specifications, and this twin is stored with the system and interacts with the DNN. One of the functions of the digital twin is to create constraints for confirming or refuting the classification result of the neural component.

The neural component extracts the relevant features from the traffic and provides classification. It interacts with a digital twin and retrains according to this interaction.

The neural component is a composite of two DNNs, one of which is trained for binary classification—that is, determines normal operation and intrusion. The other works in parallel and classifies the type of attack.

The front-end component perceives the traffic from which it extracts the relevant features and, when interacting with the digital twin, receives additional features from the model. These data are also used to retrain the neural components.

Constraint-matching component computes the truth value of the constraints according to the traffic data and then makes a final verdict on the presence of an attack taking into account the DNN classification.

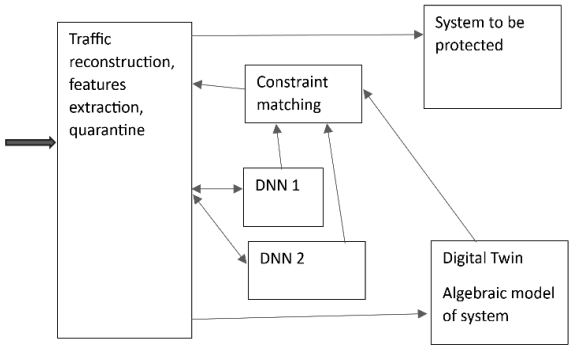


Fig. 1. General scheme of the approach

The algorithm for such a configuration is as follows:

1. A portion of the traffic is received, which extracts the relevant features of the

front-end component. Traffic is blocked in quarantine.

2. The portion of traffic is classified by both DNNs.

3. If both DNN components do not detect malicious traffic, then the traffic data are passed.

4. If at least one of the components detects an intrusion, the necessary features are submitted to the matching component with constraints data base:

a. If the matching with the constraints reveals an attack, then the intrusion is confirmed. Moreover, if one of the DNNs does not detect an attack, the received data are used for retraining.

b. If the matching with the constraints does not reveal an attack detected by the DNN with binary classification, then such an attack is considered unknown, and the traffic data are modeled on the algebraic model (digital twin). If the data lead to a violation of security properties, then the corresponding constraints are added to the constraint matching component, and both DNNs are retrained on the new case. All known security properties—such as unauthorized access, writing to forbidden memory, overflow, and others—are determined in advance and formalized as algebraic specifications.

2. The digital twin of the system as an algebraic model

The algebraic twin of the system is its model created via algebraic specifications. The basis of specifications is the algebra of behaviors [5], which is defined by operations and predicates on a set of actions and behaviors of different agents and environments. In turn, agents and environments are formal entities defined by a set of typified attributes. As such, we consider a system that is represented by its binary code, that is, a set of machine instructions for software systems or FPGA boot code for hardware.

The algebraic model is obtained automatically via appropriate translation.

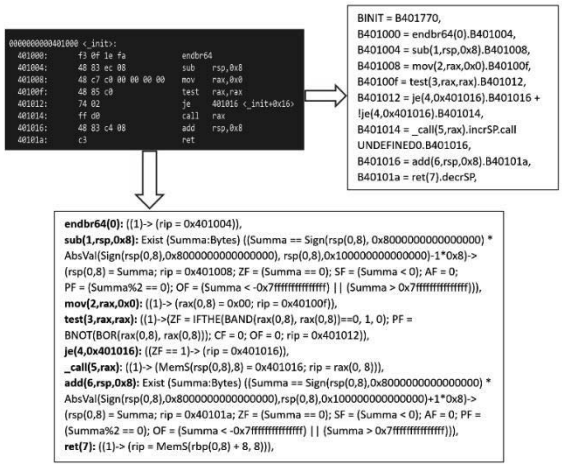


Fig. 2. Example of translation of binary code into algebraic specifications in the form of behavioral equations

Algebraic specifications are represented by the algebra of behaviors, which defines operations on the actions and behaviors of interacting agents. The operation "." is a prefixing operation that separates action from behavior. The "+" operation defines a nondeterministic choice of behaviors. Algebra is extended by operations of parallel and sequential compositions of behaviors. The behavior of the system is a set of equations in the right part of which there is an expression in terms of the algebra of behaviors. Action is a pair: a precondition that defines the state of the environment in which the action is possible and a postcondition that defines a change in the environment if the action is performed.

Thus, Figure 2 shows a set of machine instructions, which are translated into the equation of the algebra of behaviors and into the set of corresponding actions. In fact, the equations of the algebra of behaviors represent the control flow of the program, and the actions contain the semantics of the corresponding machine instruction.

For hardware, translation of the executable FPGA code is used, which is obtained after compiling the code from the hardware design language. In particular, this code is contained in the programming object file (pof).

The corresponding semantics of the commands of this file can be presented in the form of equations of the algebra of behaviors.

With an algebraic model, it is possible to apply formal methods, particularly algebraic modeling, to find vulnerabilities or possible attack scenarios.

Algebraic modeling is used to determine the reachability of some property. If the initial state of system S_0 is determined as a set of constraints F or specific values of system attributes $x_1, x_2, \ldots$

$$S_0 = F(x_1, x_2, \ldots) \quad (1)$$

Then, when choosing a possible action from the equations of the algebra of behaviors, we check the satisfiability of its precondition and environment formula, and if there are such values of attributes for which the formula is true, then we change the state according to the postcondition. Thus, during algebraic modeling, we obtain a set of symbolic states S_0, S_1, ..., namely set of formulas where the values of the attributes are not concrete; that is, each symbolic state covers a set of concrete states.

If the sought-after property $R(x_1, x_2, \ldots)$ is achievable, then there is a sequence of symbolic states that leads to the S_R state that the conjunction of the state and property is a satisfiable formula; that is, there are values $x_1, x_2, \ldots$ such that $S_R \wedge \mathrm{R}(x1,x2,\ldots)$ is true.

In addition to algebraic modeling, other formal methods can be applied, such as the search for state invariants or proving the completeness of the system (absence of deadlocks) and other useful procedures.

In the context of cybersecurity, we consider the conditions for the possibility of an attack detection or the search for vulnerabilities, which are specified by formulas over the attributes of the system, as the sought-after properties.

The algebraic twin of the system works in parallel with the neural component and performs the following functions:

1. Generates a set of constraints for classifying attacks. The generation method is described in the next section.

2. An augmented training dataset is generated based on a digital twin initial state to obtain the additional features.

3. A set of cases is generated for training a binary neural network. Generation is performed via algebraic modeling, here with the help of which possible scenarios of benign behavior of the system are built and from which appropriate features for training are extracted.

4. System behavior modeling is performed to identify an unknown attack.

5. The possible use of a digital twin is the implementation of a deceptive strategy. If the intrusion is for the purpose of reconnaissance or data theft, the digital twin can generate responses with fake data to give to the intruder. This property is not considered in the present work but is the subject of future research.

3. Attack patterns and generation of constraints

Using the algebra of behaviors, it is possible to describe the semantics of the attack. We call this pattern an algebraic attack signature.

This signature is determined by the equations of the algebra of behaviors, in which unknown or arbitrary behavior is involved. In the actions, additional constraints for triggering the attack may be present in the precondition.

The formal definition of an algebraic signature is as follows.

Algebraic signature B_i $(x_1, x_2 \ldots, Y_1, Y_2, \ldots)$ is a behavioral equation over the attributes of the system $x_1, x_2, \ldots$ and arbitrary behaviors $Y_1, Y_2 \ldots$, which determines the behavior of the intruder in the environment of the model, which is specified by the system of behavioral equations.

An attack is achievable in the model environment if there is a sequence of states that corresponds to a given algebraic signature of the attack. This correspondence is verified via algebraic modeling, particularly its variant, that is, the algebraic matching algorithm [6].

This sequence of states is used to generate constraints that are used in the intrusion detection system. If there is a sequence of states that leads to the triggering of a vulnerability or an intrusion $S_0, S_1, S_2, \ldots$, then we can to specify a constraint from the formula S_0that corresponds to a given attack.

In this case, it is necessary to strengthen the formula S_0 such that any set of attributes corresponding to this formula triggers the attack.

This is accomplished via backward algebraic modeling, where the initial state is defined as the state at which the attack is triggered and through the actions are performed for the given sequence of states leading to the initial state. The obtained formula of the initial state at which the attack is triggered is used as a constraint of the intrusion detection system corresponding to this attack.

When an attack is detected, it checks whether the constraint is true on a set of features that have been extracted from the traffic.

Note that this technology is possible when features that correspond to system attributes are used. We use a digital twin to identify such features and augment the training dataset.

Below, we consider specific examples of constraint generation and dataset augmentation.

4. Case Study

4.1 Stack Corruption Attack in a Software System

Consider the algebraic signature of the "stack corruption" vulnerability. This is a simple example of a vulnerability that can be exploited by an intruder by inputting specific data. The signature is considered as reduced only to illustrate the method.

We can describe this vulnerability via a behavioral equation:

StackDamage = syscall(0). Z; mov(ss, y, z, regn)

This *StackDamage* equation represents behavior that begins with the *syscall* action, continues with the arbitrary *Z* behavior, and ends with the *mov* action. Each action corresponds to an Intel x86 machine instruction.

The semantics of the *system* action with the specified parameters indicate an interruption when it comes to inputting information from a file. This defines the processor register *rax*, and the memory address to which the information will be written is in the *rsi* register. This memory is additionally labeled in the postcondition as *Dirty* or possibly harmful information. This will be taken into account during algebraic modeling.

syscall(0) : (rax(0,8) == 0)-> (Dirty(rsi(0,8)) = 1)

The final *mov* action contains the vulnerability itself, that is, a write outside the stack boundary in the stack memory. Additional action semantics are defined as follows:

mov(x, y, z, r) : (y+z >= top(0,8)) -> 1

The action parameters recognize x – memory type, in this case stack memory, y – stack address, z – memory size, and r – general purpose register from which the memory is written. The notation *top (0,8)* indicates the current top of the stack or the address of the stack pointer.

This signature is used in the algebraic matching algorithm, where the system model and this signature are input. Using the method of algebraic modeling, it is proven that this behavior is reachable. As a result, we obtain the final state in which the vulnerability is triggered.

This state contains constraints on system attributes such as register values and memory. Registers can be both stack and general purpose.

$$Mem(x) >= 0xffffffff \wedge Reg1 > 0xf001 \wedge SP == MEM\_LIMIT - 0x00ff \wedge \ldots$$

Next, it is necessary to determine the initial values of system attributes in the form of constraints under which the vulnerability is triggered.

We use backward modeling from the final state to the point where the input occurs via a syscall instruction.

As a result of backward modeling, we obtain a state that determines the constraints on the attributes under which the vulnerability can be triggered. For example, with the selected modeling, we determine the constraint for the initial memory, which is located at the address recorded in the rsi register. For example:

$$Memory(rsi) > 0xffffffffffff0011$$

Thus, when such data are entered from a file, a vulnerability is triggered, and if such data are present in the traffic, the program stack will be damaged and the system will stop working. Thus, we can determine the exploit of this vulnerability by defining one of the memory values that will be specific, for example, *0xffffffffffff00ff*.

The memory and register values may be given as the features that can be obtained from a partial initial run of the digital twin. The constraints related to such features are subsequently entered into a special database and matched with the appropriate classification of the neural component.

4.2 Violation of the Security Properties of Hardware

For hardware, the architecture that will make the digital twin effective has not yet been finalized. The model that exists in computer resources with access to traffic data in hardware is considered.

When hardware systems are designed, the design languages use local constraints or assertions, the violation of which is considered a security violation. Proving the impossibility of violating such a constraint, that is, invariant, is quite difficult because parts of the hardware design work mainly in parallel; therefore, the search can be endless or take considerable time.

Let us have a security property $R(x_1, x_2, \dots)$ at a certain location of the hardware system, which can be obtained already in the code that is loaded. It can be either password cracking or data overflow. We have a model of the system in terms of the algebra of behaviors. It is a translated from set of commands contained in .pof files for loading onto the FPGA.

We believe that we have a DNN that is trained to classify violations of security properties in the hardware system.

In this case, the generation of constraints can occur not according to the algebraic signature but as a result of modeling from the point of security violation $\neg R(x_1, x_2, \dots)$ before the start of receiving FPGA data from the external environment.

We obtain a set of scenarios from the modelling for values of the input signals. Also we obtain a set of constraints according to the behavior scenarios. The resulting constraint that is used will be the union of all the constraints at the data entry point.

In general, detecting attacks in hardware will be performed as in a software system. In addition to specific attacks, constraints for well-known attacks, such as side channel attacks, can be added.

Discussion and conclusions

We used a neurosymbolic approach in intrusion detection systems. In accordance with this architecture, several experiments were conducted using two DNNs and an algebraic component. DNNs were trained on available datasets [7] and augmented with features extracted from the limited initial launch of the digital twin. Data from registers and memory, which receive data from traffic, were used. For this purpose, an experimental prototype was developed, which uses standard traffic reconstruction frameworks and provides a limited initial launch of a digital twin that works with the DNN composition.

To evaluate the accuracy, a test dataset labeled with different types of attacks was selected. The first classification results were obtained in which there were no false positive detections, which confirmed the proof of concept.

The experiment was conducted on the separate use of components, which are planned to be integrated in the future into a single platform to create a software product.

In this architecture, the digital twin performs the function of generating constraints, which are used to confirm the classification of the DNN. Here, constraints are used as possible exploits in software or hardware systems to exploit a particular vulnerability. It can be assumed that the programmer wrote a program without vulnerabilities or used tools to verify vulnerabilities. On the other hand, compilers and operating systems have protection against classic vulnerabilities, such as writing to unauthorized memory. The program author cannot be certain that the compiled code that uses the interaction of different libraries does not contain vulnerabilities because vulnerabilities can be caused by the incorrect use of libraries, incompatibility of resources, and unknown paths that lead to violations. This especially applies to hardware, where there is a high degree of parallelism of various system components, which makes exhaustive verification impossible. When verifying the vulnerabilities of large systems, even at the level of binary code, different sequences of actions encounter a combinatorial explosion in the search space. Therefore, even before launching the system into permanent operation, it is impossible to exhaustively verify everything, and one of the functions of a digital twin is the permanent search for

vulnerabilities in the system and generation of appropriate constraints.

Although this architecture has a serious advantage in accuracy, several shortcomings must be considered.

We assume that a binary neural component that determines deviations from the normal operation, here trained on a certain amount of benign data, will always correctly determine the normal operation, but because it is impossible to cover the entire space of benign scenarios, false positive detections are possible. This is not an indisputable fact because a hacker can adapt to a normal scenario, even though this is quite a difficult task and requires studying the work of the classifier system that the hacker is likely to use and may be a case of not detecting the intrusion.

In this configuration, the result of detecting an attack by a binary neural component and not confirmed in the database of constraints must be verified by modeling, which determines the possibility of the reachability of the security property violation. This can cause traffic delays and system disruptions.

Thus, the presented architecture is an indisputable step in increasing the accuracy of intrusion detection systems based on the neurosymbolic approach, where the synergy of the speed of classification by the neural component and the accuracy of the algebraic approach is used. A certain number of vulnerabilities and attacks have been investigated, but a longer formalization and extension of the database of constraints and training of the system is appropriate for large critical systems that need to be protected. The optimal architecture of a digital twin for hardware has not yet been determined, so further experiments and more experience in finding vulnerabilities in hardware are required.

Acknowledgment

This work was carried out with the support of the project “Application of Algebraic Approach and Enhanced Artificial Intelligence Methods in Cybersecurity Tasks” within the NATO program “Science for Peace and Security Program.”

References

1 A. Pinto, L.-C. Herrera, Y. Donoso, and J. A. Gutiérrez, “Survey on intrusion detection systems based on machine learning techniques for the protection of critical infrastructure,” *Sensors*, vol. 23, no. 5, p. 2415, Mar. 2023, doi: 10.3390/s23052415.

2 D.-M. Ngo, A. Temko, C. C. Murphy, and E. Popovici, “FPGA hardware acceleration framework for anomaly-based intrusion detection system in IoT,” in *Proc. 31st Int. Conf. Field-Programmable Logic Appl. (FPL)*, Dresden, Germany, 2021, pp. 69–75, doi: 10.1109/FPL53798.2021.00020.

3 S. Haag and R. Anderl, “Digital twin—proof of concept,” *Manufacturing Letters*, vol. 15, pp. 64–66, Jan. 2018.

4 P. Hitzler, A. Eberhart, M. Ebrahimi, M. K. Sarker, and L. Zhou, “Neuro-symbolic approaches in artificial intelligence,” *National Science Review*, vol. 9, no. 6, Jun. 2022, doi: 10.1093/nsr/nwac035.

5 A. Letichevsky, “Algebra of behavior transformations and its applications,” in *Structural Theory of Automata, Semigroups, and Universal Algebra*, V. B. Kudryavtsev and I. G. Rosenberg, Eds. Dordrecht, The Netherlands: Springer, 2005, pp. 241–272.

6 O. Letychevskyi and V. Peschanenko, “Applying algebraic virtual machine to cybersecurity tasks,” in *Proc. 2022 IEEE 9th Int. Conf. Sci. Electron., Technol. Inf. Telecommun. (SETIT)*, Hammamet, Tunisia, 2022, pp. 161–169, doi: 10.1109/SETIT54465.2022.9875895.

7 Y. Mirsky, “Kitsune network attack dataset,” Kaggle. [Online]. Available: https://www.kaggle.com/datasets/ymirsky/network-attack-dataset-kitsune. [Accessed: 14.09.2024].

Одержано: 18.03.2025
Внутрішня рецензія отримана: 27.03.2025
Зовнішня рецензія отримана: 28.03.2025

Про авторів:

[1]*Летичевський Олександр Олександрович,*
доктор фізико-математичних наук,
завідувач відділу
https://orcid.org/0000-0003-0856-9771

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

Місце роботи авторів:

[1]Інститут кібернетики імені В. М.
Глушкова НАН України,
03187, м. Київ,
просп. Академіка Глушкова, 40.
Тел. (+38) (044) 526-20-08
Email: office@icyb.kiev.ua,
www.incyb.kiev.ua

[2]Херсонський державний університет
73003, м. Херсон (Юридична адреса),
вул. Університетська, 27.
76018, м. Івано-Франківськ (Фактична
адреса), вул. Шевченка, 14.
Тел. +38 (0552) 226263
Email: office@ksu.ks.ua

УДК: 004.056.55 https://doi.org/10.15407/pp2025.02.028

І.О. Шахматов, І.В. Замрій

ІНТЕГРОВАНА СИСТЕМА БЕЗПЕКИ ДЛЯ ЗАХИСТУ СИНХРОНІЗАЦІЇ ПЛАТЕЖІВ ВІД MITM-АТАК

У статті вирішується проблема захисту синхронізації платежів від MITM-атак у багатоканальних платіжних системах, де готівкові, карткові та онлайн-платежі інтегровані з CRM і бухгалтерськими платформами. Розглянуто сценарії атак «людина посередині» та їхній вплив на цілісність транзакцій. Запропоновано багаторівневу систему захисту з використанням методів штучного інтелекту, криптографії (цифрові підписи й часові мітки) та додаткової верифікації клієнтів. Модель підвищує стійкість до фальсифікацій, повторних атак і підміни даних, забезпечуючи високу надійність і масштабованість у системах, що працюють із електронними транзакціями.
Ключові слова: Захист платежів, MITM-атака, безпека фінансових транзакцій, штучний інтелект у кібербезпеці.

I.O. Shakhmatov, I.V. Zamrii

INTEGRATED SECURITY SYSTEM FOR PROTECTING PAYMENT SYNCHRONIZATION FROM MITM ATTACKS

The article addresses the challenge of securing payment synchronization against Man-in-the-Middle (MITM) attacks in multichannel payment systems, where cash, card, and online transactions are integrated with CRM and accounting platforms. It examines MITM attack scenarios and their impact on transaction integrity. A multi-layered security framework is proposed, leveraging artificial intelligence techniques, cryptographic methods (digital signatures and timestamps), and additional client verification mechanisms. The model enhances resilience against fraud, replay attacks, and data substitution, ensuring high reliability and scalability across electronic transaction systems.
Keywords: Payment security, MITM attack, financial transaction security, artificial intelligence in cybersecurity.

Вступ

Електронні платежі стали невід'ємною частиною фінансових операцій, забезпечуючи зручність, швидкість і широку географічну доступність транзакцій. Інтеграція CRM-систем та автоматизованих бухгалтерських платформ значно підвищує ефективність і прозорість обробки платежів, дозволяючи відстежувати кожен етап транзакції, автоматизувати розподіл коштів і спрощувати фінансову звітність. Однак разом із розвитком інноваційних платіжних рішень зростає й ризик кібератак, серед яких особливо небезпечними є атаки типу «людина посередині» (Man-in-the-Middle, MITM). Вони становлять загрозу не лише для конфіденційності, а й для цілісності фінансових даних та репутації компаній, що обробляють великі обсяги платежів.

Значну вразливість виявляють складні платіжні системи, що поєднують кілька способів оплати (готівка, банківські картки, онлайн-платежі), а також використовують реєстрацію чеків через ПРРО (Checkbox) та зовнішні сервіси на кшталт Portmone. Через інтеграцію між платіжними пристроями, серверами, CRM-модулями та бухгалтерськими системами утворюється комплексна інфраструктура обміну даними, яка може стати привабливою ціллю для кіберзловмисників. Зокрема, несанкціоноване перехоплення та модифікація платіжної інформації в реальному часі може призвести до фінансових втрат, витоку конфіденційних даних чи порушення цілісності всієї системи (Рис. 1). MITM-атаки характеризуються здатністю зловмисника непомітно «встава-

ти» між учасниками платіжної комунікації, підмінюючи або перехоплюючи дані без прямого виявлення. Ця проблема стає особливо актуальною в умовах, коли традиційні засоби автентифікації (наприклад, базові протоколи шифрування чи цифрові підписи) не завжди забезпечують належний рівень захисту від витончених методів атак, що можуть використовувати динамічну модифікацію трафіку, повторне відтворення (Replay) транзакцій та маніпуляції часовими мітками.

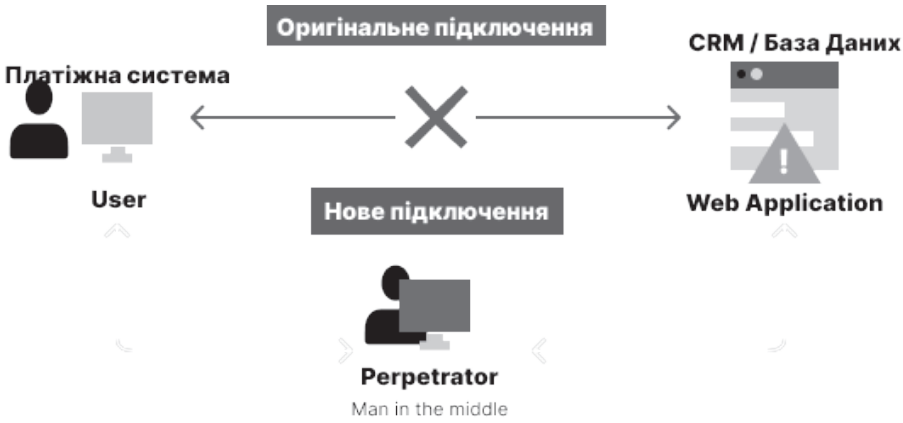


Рис. 1. Схема потенційної MITM-атаки в платіжній системі

У зв'язку з цим зростає потреба у створенні багаторівневих та адаптивних механізмів захисту, здатних ефективно протидіяти MITM-атакам. Важливу роль у забезпеченні безпеки відіграють засоби криптографічного захисту, зокрема продумані протоколи шифрування, цифровий підпис і маркери часу, що ускладнюють маніпуляції з боку зловмисників.

Аналіз літературних даних

Зростання безготівкових платежів супроводжується ризиками атак MITM, що загрожують цілісності та безпеці фінансових транзакцій. Основні уразливості мобільних платіжних систем пов'язані з недостатнім рівнем автентифікації, слабким шифруванням даних та можливістю перехоплення інформації під час синхронізації платежів. Інтегрована система безпеки, яка поєднує шифрування, механізми виявлення аномалій та динамічні протоколи перевірки даних, мінімізує ризик атак MITM та підвищує стійкість платіжних платформ [1].

Для захисту платіжних транзакцій широко застосовуються методи шифрування та автентифікації. Використання багаторівневих механізмів, зокрема багатофакторної автентифікації та наскрізного шифрування, забезпечує значне зниження ризиків компрометації даних [2]. Одним із ефективних рішень є використання зашифрованих систем керування, які не лише забезпечують конфіденційність, а й виявляють аномалії, що можуть свідчити про можливу атаку. Дослідження показують, що ефективність таких систем залежить від типу шифрування, параметрів його налаштування та механізмів перевірки цілісності даних [3].

Традиційні методи захисту не завжди ефективні проти сучасних атак, які використовують алгоритми машинного навчання для обходу класичних систем безпеки. Впровадження адаптивних та самонавчальних систем дозволяє оперативно реагувати на загрози та передбачати потенційні вразливості. Інтеграція ШІ в мобільні платіжні системи сприяє аналізу поведінкових патернів, виявленню аномального трафіку та покращеному шифруванню, що суттєво знижує ризик атак MITM [4]. Дослідження показують, що використання алгоритмів класифікації, таких як Random Forest та глибоке навчання, дозволяє ефективно розрізняти нормальну та

аномальну поведінку в платіжних системах, що удосконалює виявлення загроз у режимі реального часу [5].

Одним із перспективних напрямків у сфері кібербезпеки є застосування технології блокчейну для захисту синхронізації платежів. Завдяки своїй децентралізованій природі, блокчейн гарантує незмінність записів і прозорість операцій, що значно ускладнює підробку або модифікацію даних [6]. Поєднання технологій федеративного навчання та блокчейну дозволяє ідентифікувати шахрайські транзакції без передачі конфіденційних даних між фінансовими установами. Використання смарт-контрактів забезпечує автоматичну перевірку платіжних запитів, що значно підвищує рівень безпеки [7]. Крім того, нові виклики в кібербезпеці вимагають розгляду технологій, стійких до квантових атак. Квантова криптографія забезпечує захист фінансових даних завдяки механізмам, що залишаються безпечними навіть у разі появи квантових комп'ютерів [11]. Інтеграція цієї технології разом із технологією розподіленого реєстру (DLT) дозволяє створити децентралізовану систему, що гарантує незмінність історії транзакцій та захищає їх від підробки.

Оптимальний рівень захисту забезпечується завдяки поєднанню кількох методів, таких як штучний інтелект, блокчейн та адаптивні алгоритми аналізу мережевого трафіку. Наприклад, комбінація методів XGBoost і Random Forest дозволяє ефективно класифікувати та блокувати атакуючі запити, а використання алгоритмів градієнтного бустингу покращує ідентифікацію аномальних операцій у фінансових потоках [8, 12]. Додатково білінійний протокол транзакцій із подвійною автентифікацією підсилює безпеку за рахунок двоетапного підтвердження операцій, унеможливлюючи компрометацію платежів навіть у разі витоку облікових даних [12]. Інтеграція штучного інтелекту в систему безпеки платіжних операцій забезпечує динамічну адаптацію до нових загроз, аналізуючи мережевий трафік та відстежуючи шахрайські схеми в режимі реального часу [13]. Використання ймовірнісних нейронних мереж (PNN) значно підвищує ефективність виявлення MITM-атак, оскільки дозволяє зменшити кількість хибнопозитивних спрацьовувань та підвищити швидкість реакції на загрози [14]. Застосування алгоритмів машинного навчання для аналізу транзакцій у режимі реального часу забезпечує ефективний захист від складних шахрайських схем. Використання GBDT для аналізу фінансових потоків дозволяє знаходити приховані патерни шахрайства та блокувати потенційно шкідливі запити до їх виконання [15]. Поєднання блокчейну з механізмами моніторингу транзакцій створює додатковий рівень прозорості операцій та забезпечує захист від атак MITM [16, 17].

Атаки MITM залишаються серйозною загрозою для синхронізації платежів, однак сучасні технології дозволяють ефективно їм протидіяти. Поєднання методів шифрування, машинного навчання, квантової криптографії та блокчейну забезпечує комплексний захист фінансових транзакцій. Впровадження децентралізованих систем перевірки транзакцій, механізмів виявлення аномалій та розширених криптографічних протоколів дозволяє мінімізувати ризик атак MITM та забезпечити високу стійкість платіжних платформ [10, 17].

Навіть за умови значної кількості досліджень та впровадження різноманітних криптографічних і блокчейн-рішень, наявні напрацювання залишають відкритими кілька ключових питань. Перш за все, не має єдиної цілісної методики, яка б одночасно охоплювала всі рівні платіжної інфраструктури – від фізичних пристроїв прийому платежів до інтегрованих CRM-рішень. Окремі методи машинного навчання або блокчейн-технології використовуються переважно як самостійні модулі, не завжди забезпечуючи узгоджену взаємодію між етапами опрацювання транзакцій. Додатковою проблемою є адаптивність систем безпеки до нових форм MITM-атак. Більшість наявних рішень передбачає виключно реагування на типові загрози, не враховуючи еволюцію шахрайських схем і можливість складних комбінацій атак (наприклад, одночасне використання фішингових методів і перехоплення

трафіку). Попри високий потенціал нейромережевих моделей і технологій квантового шифрування, їх практичне впровадження у фінансових платформах усе ще обмежується великою кількістю конфігураційних параметрів та вимогами до обчислювальних ресурсів.

Також залишається недостатньо дослідженим питання динамічної корекції системи в реальному часі. Більшість систем виявлення аномалій вимагає попереднього навчання на значному наборі історичних даних, що ускладнює миттєву адаптацію до нових сценаріїв атак. Водночас комплексна інтеграція блокчейну, квантових протоколів і машинного навчання перебуває на стадії концептуальних чи пілотних проєктів. Немає статистично верифікованих оцінок ефективності цих підходів у середовищах із високим навантаженням і великим обсягом транзакцій, що є типовим для комерційних платіжних систем. З урахуванням наведених обмежень та викликів, подальші дослідження мають зосередитися на розробці інтегрованих, масштабованих і динамічно адаптивних рішень, здатних оперативно реагувати на нетипові сценарії атак MITM. Це підкреслює актуальність і нагальну потребу у комплексних системах захисту, які будуть не лише вчасно виявляти, а й ефективно блокувати спроби перехоплення, підміни або повторного відтворення платіжних даних у різних каналах обробки фінансових транзакцій.

Зважаючи на необхідність постійно розвивати системи захисту MITM атак, необхідно впроваджувати методи поведінкового аналізу, які дозволяють виявляти аномальні транзакції та підозрілі запити, використовуючи інтелектуальні алгоритми для аналізу закономірностей взаємодії користувачів із системою. Додатковим рівнем безпеки стає багатоетапна верифікація, яка передбачає повторну ідентифікацію користувача у разі виявлення підозрілої активності, що мінімізує ризик несанкціонованого доступу. Важливою складовою є також система моніторингу та логування, яка забезпечує детальний запис усіх ключових дій, дозволяючи оперативно виявляти нові вектори атак і регулярно оновлювати протоколи безпеки.

Дослідження спрямоване на створення комплексної моделі захисту синхронізації платежів, здатної оперативно виявляти та блокувати MITM-атаки в режимі реального часу. Основна мета полягає у розробці гнучкої системи безпеки, яка відповідала б вимогам як малих, так і великих фінансових та CRM-платформ. Важливим аспектом такої системи є механізм виявлення аномальних транзакцій, що базується на методах аналізу даних та алгоритмах штучного інтелекту. Ефективний захист від повторних атак забезпечується впровадженням криптографічних часових маркерів, що унеможливлюють їх використання. Динамічний пороговий аналіз дозволяє швидко ідентифікувати загрози та своєчасно на них реагувати, що підвищує загальний рівень безпеки системи. Додатковим рівнем захисту стає інтеграція багаторівневих механізмів перевірки автентичності, які зміцнюють стійкість фінансових платформ до потенційних загроз.

Запропонований підхід у межах цього дослідження, може бути застосований як у вже існуючих багатофункціональних платіжних системах, так і в нових розробках фінансових сервісів, де безпека даних та безперервність процесів є пріоритетом. Крім того, запропонована модель стане основою для розробки універсальних рішень у галузі кібербезпеки, спрямованих на ефективну детекцію й нейтралізацію складних MITM-атак у платіжних та CRM-середовищах.

Опис системи

Платіжна система, що розглядається, поєднує кілька каналів проведення транзакцій, інтегрованих між собою для підвищення швидкості та зручності обробки платежів. Одним із ключових компонентів є пристрій сплати, який підтримує різні способи оплати, включаючи готівкові розрахунки, безготівкові платежі через POS-термінал та онлайн-оплату за допомогою QR-коду через платформу Portmone. Ку-

пюроприймач автоматично фіксує внесену суму та перевіряє непідробність банкнот, а POS-термінал забезпечує проведення операцій за допомогою магнітної стрічки, чипа або безконтактної NFC-технології. Під час використання онлайн-оплати на екрані пристрою генерується QR-код, який клієнт сканує для переходу до платіжного сервісу та підтвердження транзакції. Важливу роль у функціонуванні системи відіграє сервіс реєстрації чеків Checkbox. Після успішної оплати автоматично створюється електронний чек, який реєструється у програмному реєстраторі розрахункових операцій (ПРРО) відповідно до вимог законодавства. Користувач отримує посилання на чек через SMS або E-mail, що дозволяє переглядати зареєстрований документ у зручний спосіб.

CRM-система забезпечує інтеграцію платіжного процесу з бухгалтерським обліком та додатковими механізмами верифікації. Вона отримує дані про транзакції від пристрою сплати та Checkbox, автоматично передаючи їх до бухгалтерської системи для формування звітів, рахунків і закриття фінансових періодів. Для підвищення рівня безпеки можливе підтвердження особи платника через SMS-код, що вводиться у відповідне поле під час проведення операції. Додатково CRM дозволяє швидко знаходити деталі платежу за номером чека, що значно спрощує ідентифікацію клієнта, обробку повернень коштів та коригування платежів. Згаданий комплекс взаємодії між купюроприймачем, POS-терміналом, онлайн-платіжним сервісом Portmone, Checkbox та CRM-системою створює складне багатоканальне середовище. Кожний канал водночас є потенційним вектором атаки, тому забезпечення надійної взаємодії між усіма компонентами є пріоритетом.

Атаки типу «людина посередині» (Man-in-the-Middle, MITM) становлять серйозну загрозу для платіжних сервісів, оскільки дозволяють зловмиснику перехоплювати або змінювати дані, що передаються між різними компонентами системи, залишаючись непомітними для учасників транзакції. Одним із небезпечних сценаріїв є фальсифікація транзакцій, коли змінюються сума платежу, реквізити отримувача або призначення платежу. У таких випадках реальний платіж може бути перенаправлений на інший рахунок, а відображені дані у бухгалтерському обліку та CRM можуть не відповідати дійсності, що створює ризики шахрайства та фінансових втрат.

Ще одним поширеним методом є повторна атака, коли зловмисник перехоплює дані успішної платіжної операції, наприклад, пакет із підтвердженням оплати, і повторно відтворює його, щоб знову списати кошти або отримати доступ до вже оплаченої послуги. Такі атаки особливо небезпечні в системах, які не перевіряють унікальність і «свіжість» транзакцій, зокрема в разі відсутності механізмів перевірки одноразових маркерів чи часових міток. Окрему загрозу становить перехоплення SMS-коду, який використовується для аутентифікації або підтвердження платежу. Якщо зловмисник отримує доступ до мобільного пристрою жертви або використовує вразливості мережі (як-от, у протоколі SS7) чи шкідливі програми, він може несанкціоновано підтвердити транзакцію від імені користувача. Це дає можливість отримати контроль над фінансовими операціями без відома власника облікового запису. Ще одним способом маніпуляції є підміна відповіді від сервісів платіжного шлюзу або системи реєстрації чеків, таких як Checkbox чи Portmone. У цьому випадку злочинець модифікує відповідь сервера, надсилаючи підроблену інформацію про нібито успішне проведення платежу. Як наслідок, CRM-система чи бухгалтерський модуль можуть зберегти некоректні дані про оплату, що відкриває можливості для шахрайських дій, неправильного формування звітності та спотворення фінансових операцій. Усі зазначені загрози вимагають комплексного підходу до безпеки платіжних систем, що включає різнопланові механізми захисту, такі як криптографічні методи шифрування, багатофакторна автентифікація, поведінковий аналіз транзакцій та моніторинг мережевого трафіку.

У рамках дослідження основна увага зосереджена на протидії атакам типу «людина посередині» (MITM), оскільки

вони становлять особливу небезпеку для платіжних сервісів, дозволяючи зловмисникам втручатися у процес передачі даних та змінювати фінансову інформацію без відома учасників транзакції. Ефективний захист від MITM є одним із ключових аспектів безпеки всієї системи, адже його відсутність може зробити вразливими навіть найбільш захищені компоненти інфраструктури, зокрема, CRM та бухгалтерські модулі, які обробляють фінансові операції. Для мінімізації ризиків була запропонована багаторівнева система захисту, що включає механізми формування платіжного запиту, верифікації транзакцій, детекції атак та підтвердження платежу перед його передачею в CRM та бухгалтерську систему (Рис. 2). Такий підхід дозволяє забезпечити комплексну безпеку на кожному етапі обробки транзакцій, мінімізуючи можливості для несанкціонованого втручання.

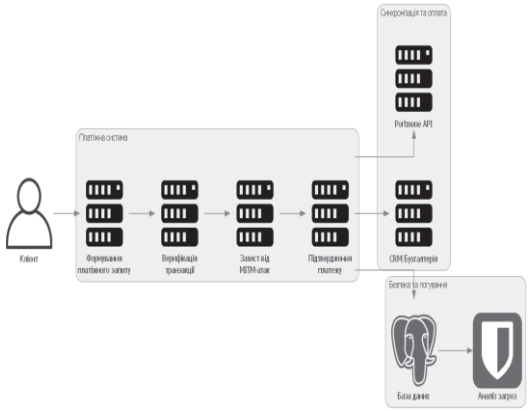


Рис. 2. Діаграма розгортання системи захисту

Модель та методологія дослідження

Запропонована модель виявлення аномалій у платіжних операціях та механізмів протидії атакам MITM поєднує пороговий метод детектування, статистичний аналіз та алгоритми машинного навчання, що забезпечує надійний захист від шахрайських дій. В основі системи лежить набір вхідних параметрів, які використовуються для ідентифікації та аналізу кожної транзакції. Для кожної операції формується унікальний ідентифікатор, що однозначно визначає платіж у системі. Платіжні реквізити містять дані про суму операції, момент її ініціалізації у секундах та спосіб оплати, що може включати готівковий розрахунок, використання банківської картки або онлайн-платіж. Інформація про електронний чек включає його унікальний номер, що дозволяє відстежувати операцію, а також ідентифікатор ПРРО, який генерується системою Checkbox і має фіксовану довжину. Верифікаційні дані містять номер телефону, представлений у міжнародному форматі, та SMS-код, який використовується для підтвердження транзакції. Код може бути чотири- або шестизначним, залежно від визначеного формату безпеки. Відповідь від сервісів Portmone або Checkbox використовується для верифікації результату операції та містить ключові параметри. Статус відповіді представлений булевою змінною, що сигналізує про успішне або невдале виконання платежу. Додатково передається текстове повідомлення, яке уточнює причину успіху чи помилки, надаючи системі можливість аналізу потенційних загроз та коригування механізмів безпеки.

Повний набір даних X_i для i-ї транзакції:

$$X_i = \{T_i, P_i, C_i, V_i, R_i\}$$

Кожен елемент X_i є категоріальним, що дає змогу детально аналізувати платіж та визначати аномальні відхилення. T_i – ідентифікатор i-ї транзакції, унікальне ціле число в діапазоні $[1, 10^{12}]$, що однозначно визначає платіж у системі. P_i– платіжні реквізити, а саме: сума операції, час ініціалізації, спосіб оплати. C_i, – дані електронного чека: номер чека, ідентифікатор ПРРО. V_i – верифікаційні дані: номер телефону, SMS-код. R_i – відповідь від сервісу Portmone або Checkbox: статус відповіді, текстовий код.

Функція аномальності $\mathrm{D}(X_i)$ відображає наскільки характеристики поточної транзакції X_i відхиляються від «нормальної» поведінки, зафіксованої у системі.

$$\mathrm{D}(X_i) = \sigma(X_i),$$

де $\sigma(X_i) \geq 0$, це алгоритмічна операція, що повертає невід'ємне число і відображає ступінь «відхилення» транзакції X_i. Чим вище $\mathrm{D}(X_i)$, тим більша ймовірність, що операція містить аномальні чи потенційно небезпечні характеристики. На

цій основі встановлюється динамічний поріг δ ($\delta \geq 0$), що визначає межу допустимого відхилення:

$$u(T_i) = \begin{cases} 1, \text{якщо } \left|\mathrm{D}(X_i) - \breve{D}\right| < \delta, \\ 0, \text{якщо } \left|\mathrm{D}(X_i) - \breve{D}\right| \geq \delta . \end{cases}$$

У даному формулюванні:

$u(T_i) = 1$ свідчить про те, що транзакція не перевищує встановлений поріг аномальності й вважається умовно безпечною (однак може бути перевірена додатково).

$u(T_i) = 0$ позначає, що рівень аномальності надто високий і платежу присвоюється статус «підозрілий» із подальшим застосуванням механізмів блокування або додаткової верифікації.

Оскільки атака MITM змінює принаймні один із параметрів транзакції (наприклад, суму або чекові дані), імовірність потрапляння такої зміненої транзакції за межі порогу δ є високою, за умови належного налаштування $\mathrm{D}(X_i)$. Рівень детектування атаки визначається показником P_d у межах [0,1]:

$$P_d = 1 - (P_f)^N,$$

де P_f – імовірність хибного негативу $\in$ [0,1], тобто вірогідність, що система не виявить справжню атаку. N – кількість перевірок, $\in N$, $N \geq 1$, під час яких аналізуються різні складові транзакції (верифікаційний код, відповідь від платіжного сервісу, часові мітки). Із зростанням N і покращенням якості оцінки аномальності (зменшення P_f), величина P_d наближається до 1, що означає майже гарантовану детекцію спроби фальсифікації даних.

Для підвищення точності виявлення складних видів шахрайства запроваджується нейромережева модель f_θ, що оперує вектором X_i. Результатом обчислень є ризик-функція $R(T_i)$, значення якої лежить в інтервалі [0,1]:

$$R(T_i) = f_\theta(X_i),$$

значення якої лежить в інтервалі [0,1]. $R(T_i)$ відображає ймовірність того, що транзакція T_i є атакованою. Тут f_θ – функція нейромережі, параметризована векторами вагами θ, Ваги θ це числові коефіцієнти (вектори та матриці), які визначають зв'язки між нейронами на суміжних шарах мережі. Під час навчання мережі ці ваги коригуються так, щоб мінімізувати обрану функцію втрат і забезпечити найбільш точну класифікацію транзакцій (виявлення атак).

Для оцінки загрози встановлюється порогове значення $\tau \in [0,1]$. Якщо

$$R(T_i) > \tau,$$

то транзакція вважається потенційно атакованою. На відміну від статичного δ, поріг τ може динамічно переоцінюватися залежно від змін у платіжному потоці та появи нових сценаріїв зловмисних дій.

Запропонована модель складається з п'яти шарів, що послідовно обробляють вхідний вектор ознак X_i і формують вихідну оцінку ризику $R(T_i)$. Вхідний шар (Input Layer) отримує вектор ознак розмірності d, який утворюється після попередньої підготовки даних (масштабування числових параметрів, кодування категоріальних ознак). Усі елементи вектора X_i надходять до наступного шару без додаткової обробки. Прихований щільний шар (Dense Layer 1) містить 128 нейронів, кожен з яких обчислює лінійну комбінацію вхідних сигналів із вектором ваг. Така конфігурація дозволяє відсікати від'ємні значення та підвищує здатність моделі виявляти складні залежності. Прихований щільний шар (Dense Layer 2) містить 64 нейрони, зменшення кількості нейронів порівняно з попереднім шаром допомагає мережі узагальнювати дані та уникати перенавчання. Прихований щільний шар (Dense Layer 3) містить 32 нейрони, застосування третього щільного шару підвищує глибину моделі та дає змогу виявляти більш витончені патерни шахрайства, характерні для MITM-атак. Вихідний шар (Output Layer) складається з одного нейрона із сигмоїдною активацією, яка обмежує вихідне значення в інтервалі [0,1]. Цей вихід трактується як імовірність атаки (ризик-функція $R(T_i)$), де значення, близькі до 1, свідчать про високу вірогідність MITM-атаки.

Під час навчання мережі вагові коефіцієнти θ усіх шарів коригуються методом зворотного поширення помилки (backpropagation). Застосовується функція втрат (наприклад, бінарна крос-ентропія)

та оптимізатор (наприклад, Adam) із заданими гіперпараметрами швидкості навчання. У результаті зменшується різниця між фактичною міткою (атака або нормальна транзакція) та виходом моделі $R(T_i)$. Після завершення процесу навчання мережа здатна з високою точністю розрізняти транзакції, які містять ознаки MITM-атак, і нормальні операції.

Якщо в результаті перевірки транзакція визначається як аномальна або потенційно шахрайська, система активує механізми додаткової верифікації та захисту. Це може включати повторне надсилання SMS-коду автентифікації, що вимагає від користувача підтвердження своєї особи. У деяких випадках потрібне втручання адміністратора, а система тимчасово блокується. За необхідності транзакція може бути повністю скасована, а клієнт негайно отримує відповідне повідомлення про причину відмови. Додатковим захисним заходом є створення нового платіжного середовища, що унеможливлює повторне використання скомпрометованих даних. Для цього система генерує новий QR-код або унікальне платіжне посилання для сервісу Portmone, що дозволяє безпечно повторити спробу оплати. Паралельно з цими заходами адміністратор отримує сповіщення про спробу атаки. Всі подібні випадки фіксуються у журналах моніторингу, що дає змогу відстежувати тенденції атак і вдосконалювати механізми кіберзахисту.

Реалізація захисту транзакцій від MITM-атак базується на багаторівневій схемі перевірки платіжного запиту, що включає автентифікацію вхідних даних, аналіз аномалій та перевірку на повторні атаки. Верифікація здійснюється шляхом перевірки відповідності платіжного запиту встановленим шаблонам, що дозволяє виключити підміну даних. Використання нейронної мережі типу Autoencoder дозволяє виявити приховані аномалії на основі статистичних характеристик транзакції. У разі виявлення підозрілих операцій система ініціює додаткові контрзаходи, такі як повторна верифікація або блокування платежу. Додатково перевіряється унікальність транзакції за допомогою timestamp-based nonce для виключення повторних атак (Replay Attack) (Рис. 3).

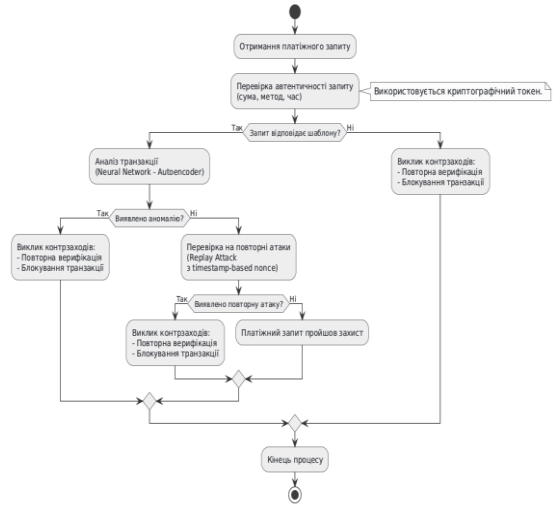


Рис. 3. Блок-схема процесу перевірки та захисту платіжних транзакцій

Застосування таких кроків значно знижує ризик успішного проведення MITM-атаки. Кожен виявлений випадок потенційно шкідливої дії фіксується в базі даних, що дає змогу вдосконалювати функцію $D(X_i)$ та перенавчати нейромережеву модель f_θ, підвищуючи загальну надійність системи.

Для виявлення нетипових операцій застосовується рекурентна нейронна мережа на базі LSTM (Long Short-Term Memory). Цей підхід ґрунтується на здатності LSTM ефективно обробляти послідовності даних, включаючи часові ряди транзакцій та послідовність подій із систем Portmone, Checkbox та CRM, забезпечуючи збереження контексту. Метод дозволяє точно виявляти патерни, характерні як для окремих транзакцій, так і для їх послідовностей, що сприяє виявленню аномальних операцій. Наведений приклад демонструє LSTM-підхід, який забезпечує високу гнучкість у процесі аналізу часової та подієвої залежності транзакцій, зокрема, серії платежів від одного клієнта протягом короткого проміжку часу.

Вхідні параметри системи містять детальну інформацію про кожну транзакцію, що аналізується для виявлення аномалій. Ідентифікатор транзакції є унікальним номером, який однозначно ідентифікує операцію. Тип пристрою визначає ка-

нал проведення платежу – термінал для готівкових операцій, POS-термінал або онлайн-сервіс, наприклад, Portmone. Метод оплати чітко вказує використання готівки, банківської картки або QR-коду. Сума платежу подається числовим значенням в обраній валюті, що є критично важливим параметром для виявлення маніпуляцій під час атак. Час транзакції зберігається у форматі дати та часу або у вигляді кількості секунд від початку доби чи епохи Unix, що дозволяє здійснювати аналіз часових аномалій. Верифікація підтверджує платіж за допомогою SMS-коду, push-повідомлення або інших методів, і під час використання SMS-коду система фіксує проходження верифікації. Статус відповіді від зовнішніх сервісів, таких як Checkbox або Portmone, містить дані про схвалення транзакції, її відхилення або помилку через тайм-аут. Результати роботи системи виявлення аномалій відображаються числовою оцінкою, що виявляє ймовірність підозрілості транзакції за шкалою від 0 до 1 або за іншою прийнятою метрикою. Порогова оцінка або кінцева класифікація чітко визначають блокування операції або переведення її на додаткову перевірку за допомогою логічних значень, таких як «0» і «1», або текстових міток «Normal» та «Attack». У разі визначення транзакції як шахрайської зазначається тип загрози, наприклад, MITM-атака, повторна транзакція (Replay Attack) або фальсифікація даних.

Таблиця 1.

Приклади класифікації

Flow ID	Attack Type	#Packets	Ground Truth	Anomaly Score	Model Output
101	ARP MitM	3500	Attack	0.89	Attack
102	- (normal)	2000	Normal	0.10	Normal
103	Active Wiretap	4580	Attack	0.78	Attack
104	- (normal)	1900	Normal	0.45	Normal
105	- (normal)	2100	Normal	0.70	Attack (FP)
…	…	…	…	…	…

Оскільки до впровадження запропонованої розробки в платіжній системі не існувало раніше реалізованого механізму виявлення MITM-атак, порівняння з попередньою версією неможливе. Натомість для тестування обрано відкритий набір даних Kitsune Network Attack Dataset [19], який містить мережеві потоки та рівні аномалій (Табл. 1), серед яких позначено (Ground Truth) реальні MITM-атаки.

Розрахунок основних показників:

TP (True Positive): атаки, які модель визначила як «атака».

FN (False Negative): атаки, які модель пропустила (позначила як «нормальні»).

Обчислення «коефіцієнта виявлення» повноти (Recall):

$$\text{Recall} = \frac{TP}{TP + FN}$$

Після класифікації зразків із Kitsune Network Attack Dataset модель виявила 98% усіх реальних MITM-атак, тобто з усіх потоків, де справді був MITM, 98% було правильно позначено як «атака».

Висновки

У межах проведеного дослідження було проаналізовано проблему захисту інтегрованих платіжних систем від MITM-атак та запропоновано підхід, що поєднує ефективні методи криптографічного захисту й виявлення аномалій на базі як порогових правил, так і алгоритмів машинного навчання. Запропонована модель є універсальним рішенням для всіх ключових етапів платіжної транзакції – від взаємодії з пристроєм сплати й сервісом реєстрації чеків (Checkbox) до обробки інформації в CRM-системі.

Гібридний підхід до захисту платіжних операцій поєднує пороговий детектор та алгоритми штучного інтелекту, що уможливлює ефективну реакцію на типові аномалії та виявлення складніших шахрайських схем. Використання порогових правил дозволяє швидко ідентифікувати очевидні відхилення, тоді як нейромережеві моделі аналізують глибші закономірності, мінімізуючи ризики атак типу «людина посередині». Така архітектура сприяє зниженню ймовірності фальсифікації даних та забезпечує багаторівневий захист платіжної системи. Модель базується на аналізі значного набору параметрів кожної транзакції, включаючи дані про платіж, електронний чек, верифікаційні процедури та відповіді від зовнішніх сервісів. Це дозволяє ідентифікувати аномалії на різних рівнях, від підміни SMS-коду чи номера телефону до фальсифікації відповідей платіжних шлюзів. Така комплексність значно підвищує чутливість системи до ознак шахрайства, дозволяючи оперативно виявляти потенційні загрози.

Захист охоплює всі рівні платіжної інфраструктури, включаючи фізичні пристрої, такі як купюроприймачі та POS-термінали, що можуть бути вразливими до перехоплення даних у мережевому трафіку. Онлайнові сервіси, зокрема, Portmone та Checkbox, захищені від атак із використанням підроблених відповідей. CRM-система, яка зберігає конфіденційні відомості про клієнтів та аналітику фінансових операцій, отримує додатковий рівень безпеки завдяки криптографічним методам, зокрема, цифровим підписам та часовим міткам. Поєднання цих заходів формує комплексну стратегію захисту, яка протидіє широкому спектру загроз. Адаптивність системи забезпечується впровадженням механізмів самонавчання, що дозволяють їй динамічно підлаштовуватися під нові сценарії атак. Нейромережа оновлює свої параметри на основі безперервного збору даних про транзакції, що особливо важливо в умовах швидкої еволюції кіберзагроз. Статичні системи захисту часто виявляються недостатньо ефективними, тоді як запропонований підхід дозволяє передбачати нові типи шахрайських дій та адаптуватися до них у режимі реального часу.

Гнучкість і масштабованість архітектури роблять її придатною для інтеграції як у невеликі платіжні рішення, включаючи POS-термінали та міні-CRM, так і в складні корпоративні системи з великою кількістю користувачів та різними типами транзакцій. Регульовані параметри порогових правил у поєднанні з можливістю розширення машинного навчання дозволяють масштабувати рішення без кардинальних змін у його логіці, забезпечуючи ефективний захист незалежно від розміру платіжної інфраструктури. Подальший розвиток системи передбачає вдосконалення моделей машинного навчання шляхом впровадження спеціалізованих нейромережевих структур, таких як рекурентні блоки LSTM або трансформери, що дозволять точніше прогнозувати аномальні патерни. Додаткове розширення набору ознак включатиме поведінкові фактори, такі як геолокація та біометричні дані, а також аналіз мережевого трафіку, що включає IP-адреси та часові характеристики запитів. Окремий акцент робиться на активному логуванні атак, що дозволяє систематично аналізувати спроби несанкціонованого доступу, виявляти нові вразливості та оперативно впроваджувати оновлення для підвищення рівня безпеки.

З метою кількісної оцінки ефективності було проведено тестування запропонованої моделі на відкритому наборі даних із платформи Kaggle (Kitsune Network Attack Dataset). Аналіз відмовостійкості та рівня виявлення аномалій підтвердив потенціал запропонованого рішення, продемонструвавши 98% детекції та низький рівень хибних спрацювань у тестовому середовищі.

References:

1. Moon I. T., Shamsuzzaman M., Mridha M. M. R., Rahaman A. S. M. Towards the advancement of cashless transaction: A security analysis of electronic payment systems // Journal of Computer and Communications. – 2022. – Т. 10, № 7. – DOI: 10.4236/jcc.2022.107007.
2. Mishra S. Exploring the impact of AI-based cyber security financial sector management //

Applied Sciences. – 2023. – T. 13, № 10. – Article 5875. – DOI: 10.3390/app13105875.

3. Rabbani H., Shahid M. F., Khanzada T. J. S., Siddiqui S., Jamjoom M. M., Ashari R. B., Ullah Z., Mukati M. U., Nooruddin M. Enhancing security in financial transactions: A novel blockchain-based federated learning framework for detecting counterfeit data in fintech // PeerJ Computer Science. – 2024. – T. 10. – Article e2280. – DOI: 10.7717/peerj-cs.2280.
4. Kawano T., Okada Y. Experimental validation of the attack-detection capability of encrypted control systems using man-in-the-middle attacks // IEEE Transactions on Industrial Informatics. – 2023. – T. 19, № 1. – C. 123–132. – DOI: 10.1109/TII.2023.1234567.
5. Obonna U. O., Opara F. K., Mbaocha C. C., Obichere J.-K. C., Akwukwaegbu I. O., Amaefule M. M., Nwakanma C. I. Detection of man-in-the-middle (MitM) cyber-attacks in oil and gas process control networks using machine learning algorithms // Future Internet. – 2023. – T. 15, № 8. – Article 280. – DOI: 10.3390/fi15080280.
6. Ahuja N., Singal G., Mukhopadhyay D. Ascertain the efficient machine learning approach to detect different ARP attacks // Computers & Electrical Engineering. – 2022. – T. 99. – Article 107757. – DOI: 10.1016/j.compeleceng.2022.107757.
7. Kampourakis V., Kambourakis G., Chatzoglou E., Zaroliagis C. Revisiting man-in-the-middle attacks against HTTPS // Network Security. – 2022. – T. 2022, № 3. – C. 8–16. – DOI: 10.12968/S1353-4858(22)70028-1.
8. Muzammil M. B., Bilal M., Ajmal S., Shongwe S. C., Ghadi Y. Y. Unveiling vulnerabilities of web attacks considering man-in-the-middle attack and session hijacking // IEEE Access. – 2024. – T. 12. – C. 6365–6375. – DOI: 10.1109/ACCESS.2024.3350444.
9. Alenezi M. A., Alabdulkreem E. A. Encryption algorithms modeling in detecting man-in-the-middle attacks // International Journal of Advanced Computer Science and Applications. – 2020. – T. 11, № 5. – C. 1–7.
10. Al-Abadi A. A. J., Mohamed M. B., Fakhfakh A. Enhanced random forest classifier with K-means clustering (ERF-KMC) for detecting and preventing distributed-denial-of-service and man-in-the-middle attacks in Internet-of-Medical-Things networks // Computers. – 2023. – T. 12, № 12. – Article 262. – DOI: 10.3390/computers12120262.
11. Agrawal S. Harnessing quantum cryptography and artificial intelligence for next-gen payment security: A comprehensive analysis of threats and countermeasures in distributed ledger environments // International Journal of Science and Research. – 2024. – T. 13, № 3. – C. 682–687. – DOI: 10.21275/SR24309103650.
12. Saranya A., Naresh R. Dual authentication for payment request verification over cloud using bilinear dual authentication payments transaction protocol // International Journal of Advanced Computer Science and Applications. – 2022. – T. 13, № 7. – C. 25–30. – DOI: 10.14569/IJACSA.2022.0130737.
13. Luo B., Zhang Z., Wang Q., Ke A., Lu S., He B. AI-powered fraud detection in decentralized finance: A project life cycle perspective // ACM Computing Surveys. – 2024. – T. 57, № 4. – Article 96. – DOI: 10.1145/3705296.
14. Omer N., Samak A. H., Taloba A. I., Abd El-Aziz R. M. A novel optimized probabilistic neural network approach for intrusion detection and categorization // Alexandria Engineering Journal. – 2023. – T. 72. – C. 351–361. – DOI: 10.1016/j.aej.2023.03.093.
15. Ren Y., Ren Y., Tian H., Song W., Yang Y. Improving transaction safety via anti-fraud protection based on blockchain // Connection Science. – 2023. – T. 35, № 1. – Article 2163983. – DOI: 10.1080/09540091.2022.2163983.
16. Ashfaq T., Khalid R., Yahaya A. S., Aslam S., Azar A. T., Alsafari S., Hameed I. A. A machine learning and blockchain based efficient fraud detection mechanism // Sensors. – 2022. – T. 22, № 19. – Article 7162. – DOI: 10.3390/s22197162.
17. Zamrii I., Shakhmatov I., Yaskevych V. BlockchainSQLSecure: Integration of blockchain to strengthen protection against SQL injections // Bulletin of Taras Shevchenko National University of Kyiv. Series: Physics and Mathematics. – 2024. – T. 78, № 1. – C. 160–168. – DOI: 10.17721/1812-5409.2024/1.29.
18. Yisroel Mirsky. Kitsune Network Attack Dataset: Nine labeled attacks with extracted features and the original network capture [Електронний ресурс] / Kaggle. – Режим доступу: https://www.kaggle.com/datasets/ymirsky/network-attack-dataset-kitsune

Одержано: 10.03.2025
Внутрішня рецензія отримана:20.03.2025
Зовнішня рецензія отримана: 23.03.2025

Про авторів:

Замрій Ірина Вікторівна,
доктор технічних наук, професор,
завідувач кафедри
https://orcid.org/0000-0001-5681-1871

Шахматов Іван Олександрович,
аспірант
https://orcid.org/0009-0004-9628-0365

Місце роботи авторів:

Державний університет
інформаційно-комунікаційних
технологій, Київ, Україна
(096)827-76-50
i.zamrii@duikt.edu.ua
i.shahmatov@duikt.edu.ua

УДК 004.41 http://doi.org/10.15407/pp2025.02.040

Г.Б. Мороз

ЗАСТОСУВАННЯ ГЛИБОКОГО НАВЧАННЯ З ПІДКРІПЛЕННЯМ ДЛЯ ДОВГОСТРОКОВОЇ ДИНАМІЧНОЇ КОМПОЗИЦІЇ ВЕБСЕРВІСІВ

У сервісно-орієнтованій архітектурі програмне забезпечення та системи абстрагуються як вебсервіси, які можуть викликатися іншими системами. Композиція сервісу – це технологія, яка будує складну систему шляхом поєднання існуючих простих вебсервісів. Із розвитком сервісно-орієнтованих архітектур та технології вебсервісів починають з'являтися масові вебсервіси з однаковими функціями. Вони обслуговуються різними організаціями та мають різну якість обслуговування. Отож, як вибрати відповідні вебсервіси, щоб уся система забезпечувала найкращу загальну якість обслуговування, стало ключовою проблемою в дослідженні композиції вебсервісів. Крім того, через складність і динаміку мережевого середовища якість обслуговування з часом може змінюватися. Отже, як динамічно налаштувати систему композиції, щоб адаптуватися до мінливого середовища та забезпечити якість складеного вебсервісу є ще однією проблемою. Для вирішення вищезазначених проблем, пропонується підхід до композиції вебсервісів, заснований на прогнозуванні якості вебсервісів та навчанні з підкріпленням. Зокрема, ми використовуємо нейронну мережу довгої короткотривалої пам'яті для прогнозування якості обслуговування, а потім робимо динамічний вибір вебсервісів за допомогою навчання з підкріпленням. Цей підхід можна добре адаптувати до динамічного мережевого середовища.

Ключові слова: Сервісно-орієнтоване обчислення, якість обслуговування, композиція вебсервісу, композитний вебсервіс, глибоке навчання з підкріпленням.

H.B. Moroz

APPLICATION OF DEEP REINFORCED LEARNING FOR LONG-TERM DYNAMIC COMPOSITION OF WEB SERVICES

In the service oriented architecture, software and systems are abstracted as web services to be invoked by other systems. Service composition is a technology, which builds a complex system by combining existing simple services. With the development of service oriented architecture and web service technology, massive web services with the same function begin to spring up. These services are maintained by different organizations and have different quality of service. Thus, how to choose the appropriate service to make the whole system to deliver the best overall quality of service has become a key problem in service composition research. Furthermore, because of the complexity and dynamics of the network environment, quality of service may change over time. Therefore, how to dynamically configure the composition system to adapt to the changing environment and ensure the quality of the composed web service is another problem. To address the above challenges, we propose a service composition approach based on quality of web service rediction and reinforcement learning. Specifically, we use long short-term memory neural network to predict the quality of web service hrough reinforcement learning**.** This approach can be well adapted to a dynamic network environment.

Keywords: Service-oriented computing, quality of service, web service composition, composite web service, deep reinforcement learning.

Вступ

Мережеві технології мають великий вплив на розробку програмного забезпечення в багатьох сферах застосування. Все більше підприємств і організацій надають своє програмне забезпечення, системи, обчислювальні ресурси та ресурси зберігання тощо у формі вебсервісів для використання іншими користувачами. Тому питання про те, як ефективно інтегрувати різні вебсервіси, стало фундаментальною проблемою дослідження.

Композиція вебсервісів (*web services composition*, WSC) в основному вивчає, як створити програмну систему, яка може задовольнити складні функціональні вимоги користувачів шляхом поєднання існуючих

вебсервісів. При цьому потрібно враховувати, що вебсервіси з однаковою функціональністю можуть надаватися різними постачальниками і з різною якістю обслуговування (*quality of service*, QoS), яка в основному використовується для позначення нефункціональних атрибутів вебсервісів, як-то ціну, зручність використання, надійність, час відгуку, репутацію, пропускну здатність мереж і так далі. Через динамічну зміну мережевого середовища, коливання продуктивності самого вебсервісу та зміну шаблонів доступу користувачів QoS також можуть динамічно змінюватися з часом. Тож, методи WSC необхідно адаптувати до динамічно змінюваного середовища [1].

Наразі процес WSC, зазвичай, представляється як робочий. Кожний абстрактний вебсервіс описує його необхідні функціональні можливості і може мати кілька *реальних сервісів-кандидатів (компонентів)*, які відповідають функціональним вимогам. WSC на основі робочого процесу зосереджується на QoS. Для кожного абстрактного вебсервісу використовуючі QoS визначається кращий реальний вебсервіс, що дозволяє отримати оптимальний результат WSC, який буде якнайбільше задовольняти вимоги користувача.

Час перебування конкретних компонентів у складі композитного вебсервісу (*composite web service*, CWS**)** сильно залежать від їхньої здатності підтримувати довгострокові вимоги до QoS [2]. Розробка довгострокових CWS породжує проблему вибору компонентів, які задовольняють вимоги до QoS протягом тривалих періодів часу. Такий вибір передбачає прогнозування довгострокових трендів QoS (тобто QoS протягом тривалого періоду). Основна проблема, пов'язана з прогнозуванням довгострокової QoS, полягає в тому, що її значення можуть коливатися в майбутньому. Існує принаймні три переваги використання довгострокових тенденцій QoS під час WSC. *По-перше*, розробники покладаються на прогнозовану QoS, щоб точно вибрати найкращі вебсервіси, які відповідатимуть вимогам до CWS протягом тривалого періоду часу. Це забезпечує довготривале перебування компонентів у складі CWS. *По-друге,* реальні вебсервіси можуть зазнавати кількох змін протягом свого життєвого циклу (як-от, припинення роботи вебсервісу), що може призвести до розриву зв'язку між CWS та його компонентами. В подальшому розробники зможуть замінити проблемні компоненти новими, з кращими характеристиками QoS. *По-третє***,** постачальники вебсервісу (наприклад, хмарні постачальники) можуть покладатися на прогноз QoS для кращого управління ресурсами та балансування робочого навантаження. Наприклад, вони можуть використовувати більше ресурсів сервера (процесор і пам'ять) у періоди пік. Точне прогнозування QoS дозволяє провайдерам налаштовувати свої хмарні ресурси, щоб якомога точніше задовольняти вимоги користувачів у майбутньому. Це знижує вірогідність порушення зв'язку між довгостроковим CWS та його компонентами через зміни QoS.

Існуючі в літературних джерелах підходи до WSC на основі QoS здебільшого зосереджені на статичних значеннях QoS, які спостерігаються під час композиції [3,4,5]. Проте на практиці через динамічну зміну мережевого середовища та самого вебсервісу його QoS буде змінюватися з часом, а, отже, такі майбутні варіації QoS необхідно враховувати під час розробки CWS.

Запропонований підхід поєднує прогнозування значення QoS та навчання з підкріпленням, де перше використовується для оцінки атрибутів QoS і покращення точності вибору вебсервісів у динамічному середовищі, а останнє забезпечує самооптимізацію та адаптивну здатність для композиції вебсервісів. Це приводить до нового адаптивного методу WSC з урахуванням QoS, яка може адаптуватися до динамічного мережевого середовища.

Суміжні роботи

Враховуючи динаміку та складність мережевого середовища, QoS вебсервіса можуть постійно змінюватися, тому необхідний метод WSC, який дозволяє сприймати зміни середовища та автоматично на-

лаштовувати систему. Ця проблема привернула широку увагу, і було розроблено низку рішень. У роботі [6] автори використовують метод штучного інтелекту для планування складу вебсервісу. Але цьому методу бракує загального контролю над усім CWS. У [7,8] автори розглядають задачу оптимізації композиції вебсервісу як задачу цілочисельного програмування, а потім отримують оптимальний розв'язок за допомогою суміжної технології. У [9, 10] автори описують зв'язки між різними вебсервісами за допомогою карти плану, моделюють композицію вебсервісів за допомогою моделі діаграми плану, а потім шукають вебсервіси низької якості та замінюють їх, використовуючи жадібний алгоритм. Автори роботи [11] використовують цілочисельне програмування для глобальної оптимізації WSC. Напочатку, завдяки використанню технології горизонту (skyline), значно зменшується кількість сервісів-кандидатів. Отож, складність цілочисельного програмування також суттєво зменшується. Так само в [12] автори використовують технологію горизонту для фільтрації вебсервісів. Основна ідея наведених вище методів полягає у вирішенні задачі композиції вебсервісу за допомогою цілочисельного програмування або змішаного цілочисельного програмування. Але такі методи застосовні лише до невеликих проблем і не мають здатності до самоадаптації. Таким чином, вони не можуть вирішити проблему композиції вебсервісів у великомасштабному та динамічному середовищі. Еволюційні алгоритми також використовувалися для вибору та композиції вебсервіса. У [13] була запропонована оптимізація мурашиної колонії для підтримки глобальної оптимізації QoS композитного вебсервіса. Але цим методом важко позбутися від проблеми локальної оптимізації. У [14] розглядався баланс між дослідженням та експлуатацією для композиції хмарного вебсервісу. Для цього стратегія орла (eagle strategy) була об'єднана з алгоритмом оптимізації кита (whale optimization). Цей метод може уникнути локального оптимуму, але він не враховує зміну QoS з часом, що впливає на адаптивність.

В останні роки одним із основних напрямків досліджень у галузі інформаційних технологій стає машинне навчання. Деякі вчені намагалися застосувати навчання з підкріпленням (*reinforcement learning*, RL) до композиції вебсервісів. У роботі [15] використовано модель марковського процесу ухвалення рішень *(Markov decision processes,* MDP) та навчання з підкріпленням для виконання адаптивної WSC. У [16] автори використовують ієрархічний алгоритм RL для підвищення ефективності композиції вебсервісів. Автори робіт [17 – 19] для збільшення ефективності та якості результату WSC застосовують багатоагентну технологію. У роботі [20] була запропоновано трирівневу модель WSC із підтримкою довіри. Використовуючи переваги методу нечіткої комплексної оцінки, автори представили інтегровану модель довірчого управління. Ця модель може добре аналізувати переваги користувачів, щоб отримати кращу композицію вебсервісу. Проте наведені вище алгоритми, засновані на навчанні з підкріпленням, хоч і мають певну адаптивність, але вони ігнорують деякі особливі проблеми в композиції вебсервісів. Більш конкретно, вебсервіси часто розгортаються в мережевому середовищі. QoS часто змінюється з часом через динамічне мережеве середовище та можливі коливання продуктивності самого вебсервісу. Тому для отримання кращих результатів композиції вебсервісів необхідно ефективно передбачити змінену якість вебсервісу. Незважаючи на те, що традиційний метод RL має певну здатність до самоадаптації, він усе ще не може точно оцінити якість вебсервісу, що призводить до незадовільних результатів композиції вебсервісів у високодинамічному середовищі. Отже, необхідно певним чином передбачити зміну QoS, чого можна досягти за допомогою її прогнозування.

Останнім часом багато вчених вивчають проблему прогнозування QoS. Найпоширенішим дослідженням є підходи на основі спільної фільтрації [21–26]. Цей вид методів передбачає, що подібні користувачі можуть отримати подібну QoS від вебсервісу. Вони класифікують користувачів за деякими функціями та забезпечують якісне

прогнозування для інших подібних користувачів, використовуючи історичні дані, створені користувачем, який послуговувався вебсервісом. Прогноз QoS на основі спільної фільтрації більше стосується відмінностей QoS між різними користувачами. Але в композиції вебсервісів робочий процес усієї композиції вебсервісів розроблений відповідно до вимог певного користувача. Кожний вебсервіс викликається цим користувачем. Зміни якості обслуговування викликані не зміною розташування або стану абонента вебсервісу, а тим, що мережеве середовище на той момент змінилося. QoS змінюється з часом і може бути описана як дані часового ряду, подібні до вартості акцій. Зважаючи на це, деякі вчені намагаються застосувати технологію прогнозування часових рядів для прогнозування QoS. У [27] запропоновано прогнозувати часові ряди QoS на основі моделі авторегресійної інтегрованої ковзної середньої. Однак ця модель вимагає стабільності даних часових рядів, тому її не можна застосовувати до сценаріїв із дуже динамічними змінами. Останнім часом, з розвитком технології глибокого навчання, деякі дослідники вивчали методи прогнозування послідовності на основі *рекурентних нейронних мереж* (RNN), які досягли певного прогресу в інших сферах [28–30]. Наприклад, у [31] RNN використовувалися для багатокрокового прогнозування, яке може відповідати ширшому діапазону шаблонів даних. Загалом методи на основі глибокого навчання краще відповідають складним функціям порівняно з традиційними статистичними моделями і можуть забезпечити кращу ефективність прогнозування у роботі зі складними часовими рядами.

Попередні відомості

У цьому розділі представлено деякі попередні відомості, включаючи навчання з підкріпленням та прогнозування часових рядів, які мають безпосереднє відношення до предмету дослідження.

Прогнозування часових рядів

Прогноз часових рядів призначений для оцінки майбутніх результатів на основі минулих даних. У сфері вебсервісних обчислень через динамічну зміну мережевого середовища та коливання продуктивності самого вебсервісу його QoS часто змінюється з часом. Ця зміна є певною закономірністю. Таким чином QoS можна розглядати як дані часових рядів, і наша мета полягає в тому, щоб передбачити майбутню QoS на основі історичних даних. Припускається, що дані часового ряду x_t задовольняють $x_t = g(T)$, де $g(T)$ втілює основні правила зміни x_t з часом. Метод прогнозування часових рядів зазвичай виходить з того, що майбутні дані можна передбачити на основі минулих даних. Тобто існує функція f, така, що

$$g(t + 1) \approx f(x_t, x_{t-1}, x_{t-2}, \ldots, x_0).$$

Метою прогнозування часових рядів є знаходження функції f, щоб майбутні дані можна було оцінити на основі даних минулих часових рядів [32].

Класичні методи прогнозування часових рядів включають метод ковзного середнього, метод експоненційного згладжування, авторегресійну модель і модель авторегресійного підсумовування ковзного середнього [33]. Загальний принцип цих методів полягає в припущені, що f є *лінійною* функцією. Це дозволяє вибрати необхідну модель відповідно до характеристик історичних даних. Він встановлює форму $f(\theta)$ із параметром θ, потім визначає параметри функції, аналізуючи історичні дані, і використовує цю функцію для прогнозування майбутніх даних. Більшість цих методів базується на короткострокових даних і припускають, що цільова функція f є лінійною, а це добре впливає на короткострокове прогнозування або прогнозування простого часового ряду. Однак, коли часові ряди змінюються складним чином, який неможливо описати лінійними моделями, точність передбачення вищезазначених моделей буде відносно низькою.

Навчання з підкріпленням

RL займає проміжне місце між контрольованим та неконтрольованим навчаннями, оскільки вимагає скалярного сигналу винагороди за серію вихідних дій. Воно добре досягає деяких цілей у невідомому середовищі, постійно коригуючи поведінку

агента, як-от, керування роботом для знаходження виходу у лабіринті. RL моделює дослідницьку поведінку людини чи інших організмів у невідомому середовищі. У такому середовищі агент постійно з ним взаємодіє, отримує зворотний зв'язок, а потім коригує власну поведінку. Кінцевою метою є максимізація винагороди, отриманої від середовища [34]. Існує багато різновидів навчання з підкріпленням. На рис.1 показана базова структура навчання з підкріпленням. Агент виконує дію a_t. Після її виконання він отримує із середовища винагороду r_{t+1}, а потім переходить у новий стан s_{t+1}. Агент буде коригувати вибір дій відповідно до винагороди, і, нарешті, сформує власну стратегію вибору дій, яка отримує максимальне значення винагороди. Фактично агент і середовище утворюють систему зі зворотним зв'язком.

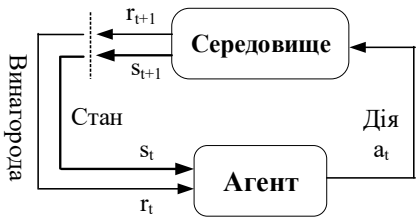


Рис. 1. Базова структура навчання з підкріпленням.

Вхідні дані алгоритму навчання з підкріпленням включають набір можливих станів середовища, допустимі дії в кожному стані, вартість винагороди та новий стан середовища, які будуть отримані в результаті виконання дії. Результатом є стратегія, яка вибирає дії на основі станів, котрі мають максимізувати очікування загальної вартості винагороди.

Зазвичай вхідними даними навчання з підкріпленням є *марковський процес ухвалення рішень* (MDP), який можна визначити як 4-кортеж MDP = ⟨S, A(·), P(·), R(·)⟩, де

S – скінченна множина станів середовища;

A(s) – скінченний набір дій, які можна виконати в стані $s \in$ S;

P($s'|s, a$) – функція розподілу ймовірностей переходу середовища від поточного стану s до наступного стану s' при виконанні дії $a \in$ A(s).

R(s, a) – функція негайної винагороди, якщо при поточному стані середовища s вибрана дії $a \in$ A(s).

Результатом навчання з підкріпленням є стратегія вибору дії, виражена як π: S → A, тобто вибір дії $a = \pi(s)$ у стані s. Найкраща стратегія вибору дій повинна максимізувати очікування загальної *цінності* винагороди. Загальна *цінність* винагороди агента (G_t), отримана за певною послідовністю *дій стану,* початковим станом якої є s_t, може бути визначена як:

$$G_t = \sum_{i=0}^{\infty} \gamma^i r_{t+i} \qquad (1)$$

де r_{t+i} є винагорода, отримана при переході зі стану s до стану s' в момент часу $t+i$. Змінна γ ($0 \le \gamma \le 1$) є ставкою дисконтування, яка гарантує, що короткотривала винагорода є більшою, ніж довготривала, і вплив нещодавньої винагороди відіграє важливішу роль у виборі дії. Функція «стан- цінність» визначається як очікування випадкової величини G_t:

$$V^{\pi}(s_t) = E^{\pi}[G_t | s_t = s] = E^{\pi}\left[\sum_{i=0}^{\infty} \gamma^i r_{t+i} | s_t = s\right] \qquad (2)$$

де $E^{\pi}(*)$ позначає очікуване значення випадкової величини за стратегією π. Величина $V^{\pi}(s_t)$ є очікуванням загальної винагороди від початкового стану s_t за стратегією π. Її рекурсивний вираз

$$V^{\pi}(s) = \sum_{a \in A} \pi(a|s) \sum_{s' \in S} p(s' | s, a)[r + \gamma V^{\pi}(s')] \qquad (3)$$

де p($s'|s, a$) представляє ймовірність переходу від стану s до наступного стану s'. Відповідно до формули (3) оптимальну стратегію вибору дії π^* можна виразити у вигляді:

$$\pi^* = \underset{\pi}{\operatorname{argmax}} V^{\pi}(s), \quad \forall s \in S \qquad (4)$$

У поєднанні визначення MDP з формулами (3) та (4), кінцеву найкращу стратегію вибору дії можна також представити формулою

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\left(\sum_{s'} p(s' | s, \pi(s))[r + \gamma V^{\pi}(s')]\right) \qquad (5)$$

Як результат вищезазначеного обговорення, входом алгоритму навчання з

підкріпленням є MDP, а виходом - стратегія вибору дії $a = \pi(s)$ у стані s, яка може бути виражена формулою (5).

Q-навчання. Це широко використовуваний алгоритм RL [35]. Його основна ідея полягає в оцінці винагороди, отриманої за дію, за допомогою таблиці значень Q, де значення Q є оцінкою реальної винагороди. Якщо фактичне та оцінене значення відрізняються після виконання дії, таблиця значень Q оновлюється різницею між двома значеннями. Значення Q може поступово наближатися до реального [36]. Після певної кількості ітерацій таблиця значень Q може мати точнішу оцінку реальної вартості винагороди. Алгоритм Q-навчання використовує Q(*s, a*) для вказівки оціненої винагороди за дію *a* у стані *s*. Стратегія вибору дій задана формулою (6):

$$\pi(s) = arg\max_a Q(s,a), a \in \text{A(s)} \quad (6)$$

Відповідно до цієї стратегії реальна вартість винагороди за дію *a* на основі стану *s* може бути виражена формулою (7), де s′ представляє наступний стан, а нижній індекс «real» представляє реальну вартість винагороди, тоді як Q(*s, a*) представляє значення в таблиці значень Q для оцінки реального значення Q:

$$Q_{real}(s,a) = R(s'|s,a) + \gamma \max Q(s',a') \quad (7)$$

Значення в таблиці Q буде оновлено відповідно до формули (8).

$$Q(s,a) = Q(s,a) + \alpha\left(R(s'|s,a) + \gamma \max Q(s',a') - Q(s,a)\right) \quad (8)$$

В алгоритмі Q-навчання вибір теоретичної дії виконується за формулою (6). Але в реальному процесі навчання, якщо щоразу, коли ми вибираємо дію з найбільшим значенням Q, вона може легко впасти в локальний оптимум. Отже, у виборі дій фактичного алгоритму Q-навчання застосовується є − жадібна стратегія. Тобто вибір дії здійснюється за формулою (6) з імовірністю є та за допомогою випадкового відбору з імовірністю 1 − є. Ця стратегія може допомогти алгоритму навчання з підкріпленням досягти певного балансу між дослідженням і використанням досвіду, а також уникнути занадто ранього потрапляння в локальний оптимум. Алгоритм Q-навчання наведено в Алгоритмі 1.

Алгоритм 1: Q-навчання

Ініціалізуйте Q(*s, a*), параметри, введіть модель MDP

repeat

встановіть поточний стан s

repeat

Виберіть *a* в A(s) на основі

є − жадібної стратегії;

Виконайте *a*, отримайте винагороду $r = R(s' \mid s, a)$ і новий стан s′;

$Q(s,a) = Q(s,a) + \alpha(r + \gamma \max Q(s',a') - Q(s,a))$

$s \leftarrow s'$

until досягнення певного стану завершення

until умова закінчення алгоритму

Глибоке навчання

Глибоке навчання — це підмножина машинного навчання, в якому використовуються нейронні мережі для розпізнавання шаблонів у наданих даних для прогнозного моделювання на прихованих даних. Дані можуть бути табличними, текстовими, графічними або мовними. Існує багато типів архітектур нейронних мереж, використання яких залежить від особливостей проблеми, що розглядається. Далі коротко представлені три з них, які стосуються проблеми, що нас цікавить.

Нейронна мережа. Базова структура нейронної мережі показана на рис. 2. Вона містить вхідний, прихований та вихідний шари. Сила зв'язку між нейронами задається вагами [37].

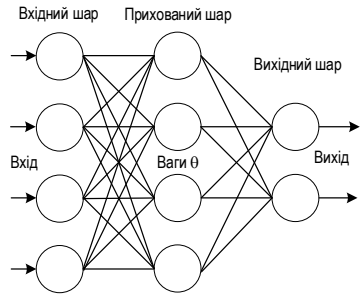


Рис. 2. Базова структура нейронної мережі.

Нейронну мережу можна представити у вигляді певної функції $y = f(x;\theta)$, де x — вхідні дані, y — вихідні дані, а θ — параметри нейронної мережі, зокрема, ваги. Вихідні дані ненавченої нейронної мережі

часто незадовільні, тобто існує помилка між виходом і очікуваним результатом. Зі зміною параметрів нейронної мережі помилка на певному наборі даних може ставати більшою або меншою. Величину такої помилки можна визначити за допомогою функції помилки L(θ). Отже, процес навчання нейронної мережі фактично полягає в знаходженні набору параметрів θ для мінімізації значення функції помилки. Для вирішення цієї проблеми часто використовують метод градієнтного спуску. Основний принцип якогополягає в тому, що напрямок градієнта є (локально) найбільш швидко змінюваним напрямком для функції, а отже, значення функції може бути зменшено через коригування незалежної змінної вздовж напрямку градієнта. Крок повторюється багато разів, поки значення функції не досягне найнижчої точки. Потім знаходиться локальне мінімальне значення. Відповідно до цього принципу оновлення параметрів нейронної мережі здійснюється за формулою:

$$\theta = \theta - \eta \frac{\partial L(\theta)}{\partial \theta} \qquad (9)$$

де η – швидкість навчання.

Рекурентні нейронні мережі. RNN є особливим типом нейронних мереж. Вони (на відміну від звичайних нейронних мереж) призначені для *обробки послідовних даних* шляхом підтримки прихованого стану, який фіксує інформацію про попередні вхідні дані [38,39]. Базова архітектура складається з вхідного шару, прихованого шару та вихідного шару. На відміну від нейронних мереж прямого зв'язку, RNN мають рекурентні з'єднання, як показано на рис. 3, що дозволяє інформації циклічно переміщатися в мережах.

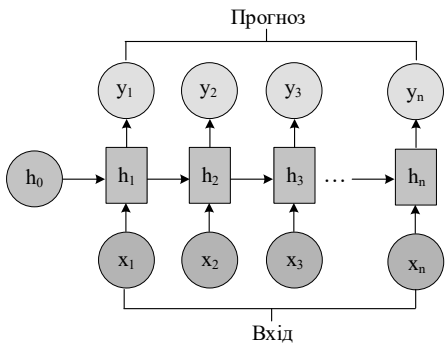


Рис. 3. Архітектура простої RNN

На кожному часовому кроці t RNN приймає вхідний вектор x_t та оновлює свій прихований стан h_t, використовуючи наступну формулу:

$$h_t = tanh(W_{xh}x_t + W_{hh}h_{t-1} + b_h),$$

де W_{xh} - вагова матриця між вхідним та прихованим шаром, W_{hh} - вагова матриця для рекурентного з'єднання, b_h - вектор зміщення, а tanh - функція гіперболічного тангенсна. Ця функція активації зводить вхідні значення до діапазону [−1, 1], роблячи її нульцентричною та придатною для моделювання послідовностей як з позитивними, так і з негативними значеннями.

Вихідний сигнал на кожному часовому кроці t визначається так:

$$y_t = \sigma(W_{hy}h_t + b_y),$$

де W_{hy} – вагова матриця між прихованим та вихідним шарами, b_y – вектор зміщення, а σ – сигмоїдальна функція активації для вихідного шару.

Нейронна мережа довгої короткотривалої пам'яті (*Long Short-Term Memory,* LSTM). Мережу LSTM було представлено Hochreiter і Schmidhuber [40] головним чином для вирішення проблеми зникнення градієнта, пов'язаної з простими RNN, що ускладнювало виявлення довготривалих залежностей у даних.

Ключовою інновацією в LSTM є використання механізмів стробування (*gating mechanisms*) для керування потоком інформації через мережу. Це дозволяє мережам LSTM підтримувати й оновлювати свій внутрішній стан протягом тривалих періодів, що робить їх ефективними для завдань, які вимагають моделювання довгострокових залежностей.

Типова внутрішня структура комірки нейронної мережі LSTM показана на рис. 4. Комірка містить три вентилі: вхідний, забувальний і вихідний, які регулюють стан комірки C_t і прихований стан h_t. Ці вентилі визначають, яку частину вхідних даних враховувати, яку частину попереднього стану потрібно забути, та яку частину стану комірки виводити.

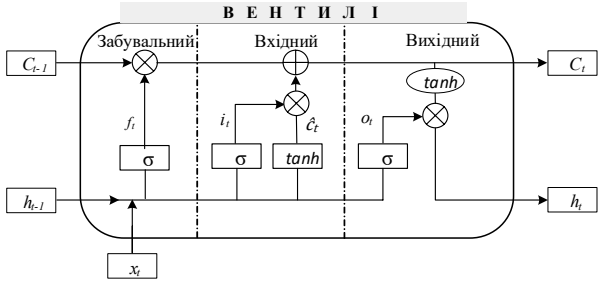


Рис. 4. Внутрішня структура комірки нейронної мережі LSTM.

Рівняння оновлення LSTM такі:

$$f_t = \sigma\big(W_f \cdot [h_{t-1}, x_t] + b_f\big) \quad (10)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (11)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (12)$$

$$\hat{c}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (13)$$

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \hat{c}_t \quad (14)$$

$$h_t = o_t \otimes tanh\,(C_t) \quad (15)$$

де f_t - забувальний вентиль, i_t - вхідний вентиль, o_t - вихідний вентиль, C_t - стан комірки, $\hat{c}_t$ - стан комірки-кандидата на стан комірки, h_t - прихований стан, σ - сигмоїдна функція, tanh -функція гіперболічного тангенса, $\otimes$ - поелементний добуток, W_f, W_i, W_0, W_C - вагові матрці, а b_f, b_i, b_0, b_C − вектори зміщення.

У забувальному вентилі швидкість забувавання обчислюється за формулою (10), і знаходиться в діапазоні від 0 до 1.

Вхідний вентиль визначає, які дані потрібно додати до стану комірки. Для цього використовуються формули (11) та (13). Нарешті, новий стан комірки контролюється спільно результатами розрахунку забувального та вхідного вентилів за (14).

Вихід мережі LSTM визначається новим станом комірки, поточним входом і виходом прихованого *шару* в останній момент часу за формулою (15).

Додавши структуру з трьома вентилями до RNN, мережа LSTM може розрізняти та контролювати інформацію, запам’ятовувати важливу інформацію протягом тривалого часу та зберігати її в стані комірки. Отже, вона може бути використаною для прогнозування часових рядів QoS.

Глибоке навчання з підкріпленням (*Deep Reinforcement Learning,* DRL) виникло як нова техніка, яка поєднує методи навчання з підкріпленням та глибокого навчання. Метою DRL є вивчення оптимальних дій, які максимізують нашу винагороду за всі стани, в яких може перебувати наше середовище. Ми робимо це, взаємодіючи зі складними, багатовимірними середовищами, випробовуючи дії та навчаючись на зворотному зв'язку. Галузь глибокого навчання стосується апроксимації функцій у багатовимірних задачах; задачах, які настільки складні, що табличні методи більше не можуть знайти точні рішення [41]. Процес навчання DRL показаний на рис. 5, який розділений на три етапи.

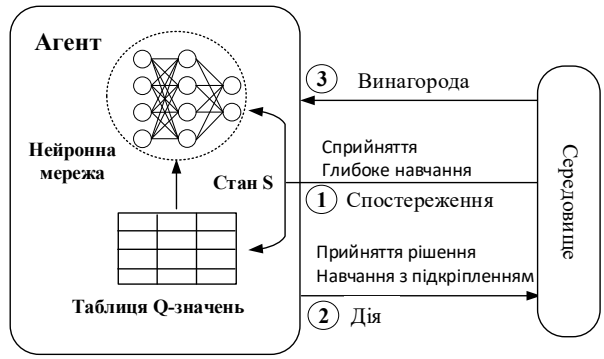


Рис. 5. Принцип глибокого навчання з підкріпленням

1. Агент отримує спостереження за допомогою взаємодії із навколишнім середовищем (через RL) і передає результати до нейронної мережі для вивчення абстрактних представлень;

2. Агент оцінює дію, виходячи з величини винагороди та зіставляє поточний стан з відповідною дією за допомогою стратегії поведінки;

3. Середовище реагує на дію та отримує наступне спостереження. Зрештою, з вивченням вищезгаданих процесів може бути досягнуто оптимальної стратегії.

Щоб RL досягло оптимального результату та конвергенції, потрібні дві важливі передумови: (1) доступна таблиця пошуку для зберігання значень Q; (2) Середовище має відповідати властивості Маркова. Перша передумова робить RL придатним лише для проблем невеликого масштабу через недостатню здатність до вираження та узагальнення. Тоді, як друга передумова передбачає, що агент чітко знає всю інформацію про навколишнє середовище, і наступний стан може бути визначений лише поточним станом і дією.

Довгострокова композиція вебсервісу на основі прогнозування QoS і DRL

У цьому розділі ми пропонуємо адаптивний метод WSC, що поєднує прогнозування QoS із навчанням з підкріпленням**.** Щоб допомогти зрозуміти проблему спочатку описується типовий сценарій композиції вебсервісу.

При WSC ми вибираємо відповідні сервіси-кандидати за допомогою RL, щоб максимізувати загальне значення QoS композитного вебсервісу. Враховуючи динаміку та складність мережевого середовища, QoS постійно змінюється, тому для її прогнозування пропонується метод на основі LSTM. Далі ми поєднуємо прогнозування QoS і навчання з підкріпленням для вирішення проблеми довгострокової композиції вебсервісу, що може покращити якість композитного вебсервісу в динамічному середовищі.

Приклад сценарію композиції вебсервісу. Типовий процес WSC показано на рис. 6, де BC_j, j=1,2,3,4 є абстрактні вебсервіси різного типу, як то готельний сервіс, авіасервіс, сервіс погоди тощо; a_i – вебсервіси-кандидати одного і того ж типу, але від різних постачальників. Для кожного абстрактного вебсервісу нам необхідно визначити його конкретні вебсервіси та сформулювати конкретний робочий процес композиції вебсервісу. Наприклад, ґрунтуючись на робочому процесі композиції вебсервісу на рис. 6, можна отримати 24 варіанти компонування вебсервісів, 2 з яких показано на рис. 7.

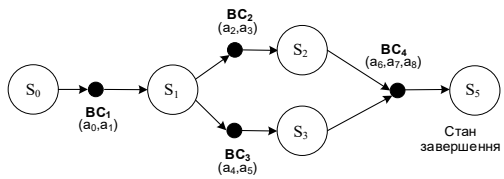


Рис. 6. Простий робочий процес створення композитного вебсервісу.

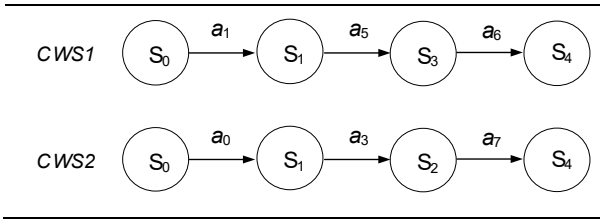


Рис. 7. Можливі результати композиції вебсервісу.

Легко побачити, що зі збільшенням кількості сервісів-кандидатів і ускладненням робочого процесу композиції вебсервісів кількість можливих результатів композиції вебсервісів різко збільшуватиметься.

Завдання композиції вебсервісу полягає в тому, щоб перевірити великий простір кандидатів і отримати оптимальний результат, який *максимізує цінність винагороди*.

Модель композиції вебсервісу в динамічному середовищі. Відповідно до проблеми, описаної в попередньому розділі, пропонується модель *динамічної композиції вебсервісу* (Dynamic Web Service Composition) на основі MDP, або коротко «модель DWSC». Ця модель являє собою 6-кортеж $DWSC=\langle S, S_0, S_r, A(\cdot), P(\cdot), R(\cdot)\rangle$, де

S – скінченна множина станів середовища;

S_0 – початковий стан, з якого починається процес композиції вебсервісу.

S_r –набір станів завершення: при досягненні будь-якого з них, процес композиції вебсервісів припиняється.

A(s) – скінченна множина вебсервісів, які можуть бути викликані в стані $s \in S$;

P(s′|*s,a*) – Імовірність переходу від поточного стану *s* до наступного стану *s′* коли викликається вебсервіс *a.*

R(s′|*s,a,t*) – функція негайної винагороди в результаті переходу середовища з поточного стану *s* до стану *s′* викликається дією *a,* де *t* представляє час виклику вебсервісу.

У складі вебсервісу цінність винагороди зазвичай визначається атрибутами QoS вебсервісу.

Для повного опису робочого процесу композиції вебсервісу можна використовувати модель DWSC. Розглянемо, як приклад, простий робочий процес на рис.6. Цей робочий процес можна описати за допомогою моделі DWSC, у якій набір станів $S = \{S_0, S_1, S_2, S_3, S_4\}$, початковий стан – S_0, набір станів завершення – $\{S_4\}$. Вебсервіси, які можна викликати в різних станах, включають $A(S_0) = \{a_0, a_1\}$ і $A(S_1) = \{a_2, a_3, a_4, a_5\}$ тощо. Приклади ймовірності переходу включають $P(S_1|S_0, a_0) = 1$, $P(S_2|S_1, a_2) = 1$ і т. д. Значення винагороди обчислюється за допомогою QoS, отриманої під час виклику вебсервісу.

Після визначення робочого процесу, процес WSC починається з початкового стану, обирається конкретний вебсервіс для кожного стану, а потім здійснюється перехід до нового стану. Коли досягнуто стану завершення, робочий процес композиції вебсервісу завершується. Цей робочий процес, який складається з кількох конкретних вебсервісів, є результатом WSC. Хороший результат композиції вебсервісів має зробити загальну цінність винагороди якомога більшою.

Функція винагороди в складі вебсервісу. Алгоритм навчання з підкріпленням видає найкращу стратегію вибору дій, використовуючи модель MDP, що робить його придатним для проблеми вибору вебсервісів. Для використання алгоритму навчання з підкріпленням, нам потрібно спочатку визначити функцію винагороди.

Оскільки під час композиції вебсервісу керуються, насамперед, вимогами користувачів до QoS, то логічно функцію винагороди визначати через атрибутами QoS. Проте нам спочатку потрібно нормалізувати ці атрибути та відобразити їх на [0, 1], так як різні атрибути QoS можуть мати різні діапазони значень. Крім того, враховуючи, що деякі атрибути QoS позитивно корельовані (наприклад, пропускна здатність), а деякі атрибути QoS корельовані негативно (наприклад, час відповіді), для визначенння винагороди будемо використовувати дві формули:

$$r = \frac{QoS - min}{max - min} \quad (15)$$

$$r = \frac{max - QoS}{max - min} \quad (16)$$

Формула (15) використовується для нормалізації атрибутів QoS, які позитивно корельовані, а формула (16) використовується для нормалізації атрибутів QoS, які корельовані негативно. Змінна *r* є результатом нормалізації значення атрибута. QoS представляє значення її атрибута після виклику вебсервісу, а max і min представляють максимальне та мінімальне значення атрибута QoS. Якщо вебсервіс має кілька атрибутів QoS, вартість винагороди буде розрахована за формулою (17).

$$R = \sum_{i=1}^{m} w_i * r_i \quad (17)$$

де R являє собою загальне значення винагороди, m - кількість розглянутих атрибутів QoS, r_i - нормалізоване значення і-го атрибута QoS, а w_i являє собою вагу і-го атрибута. Вага відображає важливість різних атрибутів. Як правило, вона налаштовується відповідно до переваг користувачів щодо різних атрибутів так, що $\sum_{i=1}^{m} w_i = 1$.

Прогнозування часових рядів на основі LSTM. У динамічному мережевому середовищі QoS часто змінюється з часом, і алгоритм навчання з підкріпленням не може оцінити тенденцію зміни якості обслуговування. У цьому розділі буде представлено метод прогнозування часових рядів на основі LSTM, щоб допомогти оцінити QoS у динамічному середовищі.

Основна ідея прогнозування часових рядів полягає в тому, щоб передбачити майбутню QoS, використовуючи минулі дані. У цій роботі використовується нейронна мережа LSTM для прогнозування

часових рядів, які можна представити таким чином:

$$x_t = LSTM(x_{t-1}, x_{t-2}, x_{t-3}, \dots, x_0; \theta), \quad (18)$$

де x_t є прогнозовані дані на наступному часовому інтервалі, а

$LSTM(x_{t-1}, x_{t-2}, x_{t-3}, \dots, x_0; \theta)$ представляє нейронну мережу LSTM із параметром θ.

Вхідними даними нейронної мережі є всі минулі дані. На практиці передбачити за всіма минулими даними часто важко. Тож, вхід нейронної мережі LSTM фактично є даними попередніх n часових інтервалів. Крім того, нейронна мережа LSTM може передбачати більше, ніж один часовий інтервал. Таким чином, передбачення нейронної мережі LSTM можна виразити формулою (19), в якій m є кількістю часових інтервалів передбачення, а n є кількістю часових інтервалів вхідних даних.

$$x_{x+m-1}, \dots, x_{t+1}, x_t = LSTM(x_{t-1}, x_{t-2}, \dots, x_{t-n}; \theta) \quad (19)$$

Після визначення входу та виходу нейронної мережі можна побудувати нейронну мережу LSTM, подану на рис. 8, де показано розширення нейронної мережі LSTM у часі. Тобто нейронна мережа, представлена на рис.8 як A, є тією самою нейронною мережею, і дані передаються зліва направо в послідовності часу. На фазі введення дані про n часових інтервалів по черзі вводяться в нейронну мережу LSTM.

Вихідні дані останньої нейронної мережі також будуть частиною поточних вхідних даних. На етапі виведення нейронна мережа LSTM спочатку виводить дані прогнозу x_t. Після цього нейронна мережа більше не має вхідних даних із зовнішніх даних (оскільки це тепер майбутній час, який більше не має фактичних даних). Вона приймає лише прогнозоване значення з попереднього моменту як вхід поточного моменту, як показано на рис. 8, і виводить прогнозоване значення на кожному кроці. Коли отримано m прогнозованих значень, завдання прогнозування завершується.

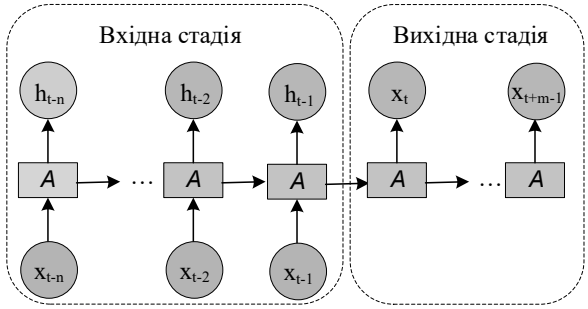


Рис. 8. Структура LSTM для прогнозування часових рядів.

Після побудови нейронної мережі LSTM для її навчання потрібна велика кількість історичних даних. Навчальний набір даних потрібно сегментувати, і кожні безперервні n+m точки даних використовуються як навчальні дані. Перші n даних є вхідними даними нейронної мережі, а дані останніх m часових кроків представляють очікувані результати прогнозування для обчислення помилки нейронної мережі. Помилка обчислюється за формулою (20), в якій x_t представляє фактичне значення моменту t, x'_t представляє прогнозоване значення в момент t, а N є кількістю точок даних у навчальному наборі. Використана функція похибки – це середня квадратична похибка:

$$L = \frac{1}{N}\sum_{t=0}^{N}(x'_t - x_t)^2 \quad (20)$$

Після навчання метод градієнтного спуску у формулі (9) використовується для налаштування параметрів нейронної мережі. Процес навчання повторюється багато разів, поки значення помилки не стане стабільним на мінімумі. Після навчання нейронну мережу можна використовувати для прогнозування майбутнього QoS.

Адаптивний алгоритм композиції вебсервісу. Розроблено новий адаптивний алгоритм композиції вебсервісів, який поєднує навчання з підкріпленням і прогнозування QoS. Перш ніж представити алгоритм, нам потрібно вирішити дві проблеми оцінки QoS у композиції вебсервісу. По-перше, у композиції вебсервісу якість сервісу-кандидата пов'язана не лише з атрибутом QoS самого вебсервісу, а й з подальшим робочим процесом, що є результатом вибору цього вебсервісу. Наприклад,

хоча вебсервіс має кращі атрибути QoS, QoS у подальшому процесі з вибором цього вебсервісу може бути поганим, що призводить до незадовільної загальної якості композиції вебсервісу. Тому в оцінці QoS ми використовуємо алгоритм Q-навчання та значення Q для представлення QoS у композиції вебсервісу. Відповідно до формули значення Q (7), значення Q фактично є лінійною суперпозицією кожного значення винагороди за вебсервіс, тому його також можна оцінити методом прогнозування часових рядів. Крім того, якщо значення Q оцінюється за допомогою нейронної мережі LSTM, формулу (7) потрібно переписати так:

$$Q_{real}(s,a) = R(s'|s,a) + \gamma \max LSTM(s',a') \quad (21)$$

де LSTM(s′, a') представляє прогнозоване значення нейронної мережі LSTM, що відповідає стану s′ і вебсервісу a′, яка є оцінкою значення Q для пари стан–дія (s', a').

До того ж, із запуском алгоритму композиції вебсервісу нові дані продовжуватимуть додаватися до набору історичних даних. Взагалі нейронна мережа навчається на фіксованому навчальному наборі, а потім застосовується до нових даних. Але в процесі створення вебсервісу історичні дані поступово накопичуються, тому потрібне поступове навчання нейронної мережі. Навчання нейронної мережі необхідно проводити, якщо додається певна кількість нових даних. Крім того, з накопиченням історичних даних вони можуть займати багато місця і також впливатимуть на ефективність навчання нейронної мережі. Тому необхідно обмежити розмір набору історичних даних і відкинути попередні історичні дані, що може підвищити ефективність навчання нейронної мережі. Детальний алгоритм наведено в Алгоритмі 2.

Алгоритм 2. ***WSC на основі прогнозування QoS і навчання з підкріпленням***

Ініціалізація LSTM(*s,a*), випадкових параметрів дослідження є, довжину вхідних даних n, ємність історичних даних d, поріг оновлення LSTM u; вхідної моделі DWSC;
repeat
  Встановити s = s_0;
  repeat
    Вибрати вебсервіс a за є − жадібною стратегією (використання LSTM для оцінки якості обслуговування);
    Викликати a, отримати винагороду за формулою (17), перейти в стан s′;
    Обчислити значення Q за формулою (21), додати ці дані до набору історичних даних;
  if кількість даних у наборі історичних даних set = d, **then**
    Відмова від найдавніших історичних даних;
  end if
  if кількість нових даних у наборі історичних даних = u,
    then
      Навчіть LSTM за формулою (20);
  end if
  s ← s′;
  until досягти стану завершення;
until Збіжність (або до заданої кількості циклів);

Алгоритм закінчується. (Прогнозне значення QoS кожного сервісу може визначити найкращий сервіс-кандидат в кожному стані для завершення композиції вебсервісу)

Висновок і майбутня робота.

Композиція вебсервісу пропонує один із найважливіших способів повторного використання програмного забезпечення шляхом поєднання існуючих вебсервісів для створення нових систем, які відповідають складним вимогам користувачів. Зі збільшенням кількості функціонально схожих вебсервісів виникає необхідність вибору відповідних вебсервісів із великого пулу кандидатів для формування якісного композитного вебсервісу. Однак у динамічному мережевому середовищі QoS не є статичним, і якість може коливатися з часом. Таким чином, метод композиції вебсервісів потрібно динамічно коригувати, щоб адаптуватися до змін навколишнього середовища. Запропонований підхід враховує особливості динамічної зміни QoS. Для оцінки часових рядів QoS розроблено рекурентну модель на основі нейронної мережі. Модель DWSC використовується для формального представлення складу вебсервісу. Розроблено адаптивний алгоритм композиції вебсервісів, який інтегрує навчання з підкріпленням і динамічне прогнозування QoS.

Запропонований метод адаптивної композиції вебсервісів на основі навчання з підкріпленням використовує таблицю значень Q

і рекурентні нейронні мережі. Це вимагає встановлення відповідної рекурентної нейронної мережі для кожного можливого вебсервісу-кандидата, що призводить до високої обчислювальної вартості для великої кількості вебсервісів-кандидатів. Для цієї проблеми ми плануємо вивчити глибоке навчання з підкріпленням і використати відображення нейронної мережі, щоб замінити таблицю Q-значення в нашій майбутній роботі. Ми також плануємо вивчити багатоагентну систему, де кілька агентів спілкуються та координують один одного для підвищення ефективності.

References

1. Sangsanit, W. Kurutach, S. Phoomvuthisarn, REST web service composition: A survey of automation and techniques, in: 2018 International Conference on Information Networking, ICOIN, 2018, pp. 116–121.
2. Kritikos, K., Pernici, B., Plebani, P., Cappiello, C., Comuzzi, M., Benbernou, S.: I, B., Kertŕesz, A., Parkin, M., Carro, M.: A survey on service quality description. ACM Comput. Surv. 46(1), 1:1–1:58 (2013)
3. Mistry, S., Bouguettaya, A., Dong, H., Qin, A.K.: Predicting dynamic requests behavior in long-term iaas service composition. In: IEEE International Conference on Web Services (ICWS) 2015, pp. 49–56. IEEE (2015)
4. Wang, S., Zhu, X., Yang, F.: Efficient QoS management for qos-aware web service composition. Int. J. Web Grid Serv. 10(1), 1–23 (2014)
5. Zeng, L., Benatallah, B., Ngu, A.H., Dumas, M., Kalagnanam, J., Chang, H.: Qos-aware middleware for web services composition. IEEE Trans. Softw. Eng. 30(5), 311–327 (2004)
6. Y. Yan, P. Poizat, L. Zhao, Repair vs. recomposition for broken service compositions, in: International Conference on Service-Oriented Computing, Springer, 2010, pp. 152–166.
7. D. Ardagna, B. Pernici, Adaptive service composition in flexible processes, IEEE Trans. Softw. Eng. 33 (6) (2007) 369–384.
8. F. Li, L. Zhang, Y. Liu, Y. Laili, QoS-aware service composition in cloud manufacturing: a Gale-Shapley algorithm-based approach, IEEE Trans. Syst.Man Cybern. Syst. (2018) 1–12.
9. Y. Yan, P. Poizat, L. Zhao, Self-adaptive service composition through graphplan repair, in: Web Services (ICWS), 2010 IEEE International Conference on, IEEE, 2010, pp. 624–627.
10. Y. Yan, P. Poizat, L. Zhao, Repairing service compositions in a changing world, in: Software Engineering Research, Management and Applications 2010, Springer, 2010, pp. 17–36.
11. M. Alrifai, D. Skoutas, T. Risse, Selecting skyline services for QoS-based web service composition, in: Proceedings of the 19th International Conference on World Wide Web, ACM, 2010, pp. 11–20.
12. V.A. Permadi, B.J. Santoso, Efficient skyline-based web service composition with QoS - awareness and budget constraint, in: 2018 International Conference on Information and Communications Technology, ICOIACT, 2018, pp.855–860.
13. O. Hammas, S.B. Yahia, S.B. Ahmed, Adaptive web service composition insuring global QoS optimization, in: 2015 International Symposium on Networks, Computers and Communications, ISNCC, IEEE, 2015, pp. 1–6.
14. S.K. Gavvala, C. Jatoth, G. Gangadharan, R. Buyya, QoS-aware cloud service composition using eagle strategy, Future Gener. Comput. Syst. 90 (2019) 273–290.
15. H. Wang, X. Zhou, X. Zhou, W. Liu, W. Li, A. Bouguettaya, Adaptive service composition based on reinforcement learning, in: International Conference on Service-Oriented Computing, Springer, 2010, pp. 92–107.
16. H. Wang, G. Huang, Q. Yu, Automatic hierarchical reinforcement learning for efficient large-scale service composition, in: 2016 IEEE International Conference on Web Services, ICWS, 2016, pp. 57–64.
17. H. Wang, X. Chen, Q. Wu, Q. Yu, Z. Zheng, A. Bouguettaya, Integratingon-policy reinforcement learning with multi-agent techniques for adaptive service composition, in: Service-Oriented Computing, Springer, 2014, pp. 154–168.
18. Y. lei, P.S. Yu, Multi-agent reinforcement learning for service composition, in: 2016 IEEE International Conference on Services Computing, SCC, 2016, pp. 790–793.
19. P. Kendrick, T. Baker, Z. Maamar, A. Hussain, R. Buyya, D. Al-Jumeily, An efficient multi-cloud service composition using a distributed multiagentbased, memory-driven approach, IEEE Trans. Sustain. Comput. (2019) 1–13.
20. W. Li, J. Cao, K. Hu, J. Xu, R. Buyya, A trust-based agent learning model for service composition in mobile cloud computing environments, IEEE Access 7 (2019) 34207–34226.

21. D. Billsus, M.J. Pazzani, Learning collaborative information filters, in: Icml, vol. 98, 1998, pp. 46–54.
22. J.L. Herlocker, J.A. Konstan, A. Borchers, J. Riedl, An algorithmic framework for performing collaborative filtering, in: SIGIR '99: Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval, August 15–19, 1999, Berkeley, CA, USA, 1999, pp. 230–237.
23. H.N. Kim, A.T. Ji, I. Ha, G.S. Jo, Collaborative filtering based on collaborative tagging for enhancing the quality of recommendation, Electron. Commer. Res. Appl. 9 (1) (2010) 73–83.
24. K. Chen, H. Mao, X. Shi, Y. Xu, A. Liu, Trust-aware and location-based collaborative filtering for web service QoS prediction, in: 2017 IEEE 41st Annual Computer Software and Applications Conference, COMPSAC, vol. 2, 2017, pp. 143–148.
25. C. Wu, W. Qiu, X. Wang, Z. Zheng, X. Yang, Time-aware and sparsitytolerant QoS prediction based on collaborative filtering, in: 2016 IEEE International Conference on Web Services, ICWS, 2016, pp. 637–640.
26. G. Zou, M. Jiang, S. Niu, H. Wu, S. Pang, Y. Gan, QoS aware web service recommendation with reinforced collaborative filtering, in: International Conference on Service-Oriented Computing, Springer, 2018, pp. 430–445.
27. R.N. Calheiros, E. Masoumi, R. Ranjan, R. Buyya, Workload prediction using ARIMA model and its impact on cloud applications x2019; QoS IEEE Trans. Cloud Comput. 3 (4) (2015) 449–458.
28. I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: Advances in Neural Information Processing Systems, 2014, pp. 3104–3112.
29. A. Graves, A.-r. Mohamed, G. Hinton, Speech recognition with deep recurrent neural networks, in: Acoustics, Speech and Signal Processing (Icassp), 2013 Ieee International Conference on, IEEE, 2013, pp. 6645–6649.
30. R.-G. Cirstea, D.-V. Micu, G.-M. Muresan, C. Guo, B. Yang, Correlated time series forecasting using multi-task deep neural networks, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, ACM, 2018, pp. 1527–1530.
31. L. Yunpeng, H. Di, B. Junpeng, Q. Yong, Multi-step ahead time series forecasting for different data patterns based on LSTM recurrent neural network, in: 2017 14th Web Information Systems and Applications Conference, WISA, 2017, pp. 305–310.
32. J.D. Hamilton, Time Series Analysis, Princeton University Press, Princeton, NJ, 1994.
33. J. Contreras, R. Espinola, F.J. Nogales, A.J. Conejo, ARIMA Models to predict next-day electricity prices, IEEE Trans. Power Syst. 18 (3) (2003) 1014–1020
34. R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction, Adaptive computation and machine learning, MIT Press, 1998
35. D.E. Goldberg, J.H. Holland, Genetic algorithms and machine learning, Mach. Learn. 3 (2) (1988) 95–99.
36. R.H. Crites, A.G. Barto, Elevator group control using multiple reinforcement learning agents, Mach. Learn. 33 (2–3) (1998) 235–262.
37. Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
38. A. Karpathy, J. Johnson, L. Fei-Fei, Visualizing and understanding recurrent networks, 2015, ArXiv preprint, arXiv:1506.02078.
39. C. Goller, A. Kuchler, Learning task-dependent distributed representations by backpropagation through structure, in: Neural Networks, 1996, IEEE International Conference on, vol. 1, IEEE, 1996, pp. 347–352.
40. S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
41. A. Plaat, Deep Reinforcement Learning, Springer Nature, 2022.

Одержано: 30.06.2025
Внутрішня рецензія оримана: 07.07.2025
Зовнішня рецензія оримана: 09.07.2025

Про автора:
Мороз Григорій Борисович,
кандидат технічних наук,
старший науковий співробітник,
провідний науковий співробітник.
https://orcid.org/0000-0001-8666-9503

Місце роботи автора:
Інститут програмних систем
НАН України
тел. +38-044-526-60-33
E-mail: moroz170@gmail.com

УДК 004.932.2:796.332 https://doi.org/10.15407/pp2025.02.054

І.Б. Івасенко, С.С. Бішир

ПІДВИЩЕННЯ ТОЧНОСТІ ВИЯВЛЕННЯ М'ЯЧА У ВІДЕО ФУТБОЛЬНИХ МАТЧІВ ЗА ДОПОМОГОЮ МЕХАНІЗМІВ УВАГИ В CNN-МОДЕЛЯХ НА ОСНОВІ FPN

Попри значний прогрес у виявленні гравців завдяки моделям глибокого навчання в спортивній аналітиці, точне розпізнавання футбольного м’яча залишається складною задачею через його малий розмір, швидкий рух, часті оклюзії та візуальну подібність до інших елементів, таких як гетри гравців, логотипи та розмітка поля. Ці обмеження значно знижують ефективність автоматизованих систем для комплексного аналізу футбольних матчів, особливо в таких задачах, як розпізнавання тактичних подій, класифікація ударів і прогнозування ігрових станів. У цій роботі запропоновано метод підвищення точності виявлення м’яча у відео футбольних матчів шляхом удосконалення наявної архітектури на основі Feature Pyramid Networks (FPN). Базова модель на основі FPN, хоча й ефективна для виявлення гравців, демонструє обмежену продуктивність у розпізнаванні дрібних об’єктів, таких як м’яч. Для вирішення цієї проблеми ми інтегрували легкі механізми уваги, які дозволяють моделі краще зосереджуватись на релевантних просторових та семантичних ознаках. Зокрема, ми впроваджуємо шари Squeeze-and-Excitation (SE) у базову мережу для переналаштування ознак на рівні каналів, а також додаємо модуль CBAM (Convolutional Block Attention Module) до голови виявлення м’яча для уточнення просторової та канальної уваги. Ці модифікації покликані покращити здатність мережі відрізняти м’яч від візуально схожих об’єктів і перевантаженого фону. Наші експерименти, проведені на наборах даних ISSIA-CNR та Soccer Player Detection, демонструють, що запропонована модель з увагою досягає кращої точності класифікації м’яча порівняно з базовим підходом, без погіршення точності виявлення гравців. Отримані результати підтверджують ефективність легких механізмів уваги в задачах виявлення дрібних об’єктів та відкривають перспективи для створення більш надійних і реалістичних систем аналізу футбольних відео у реальному часі.

Ключові слова: виявлення м’яча, глибоке навчання, аналіз футбольного відео, виявлення об’єктів, механізми уваги, Feature Pyramid Network

I.B. Ivasenko, S.S. Bishyr

ENHANCING BALL DETECTION IN FOOTBALL VIDEOS USING ATTENTION MECHANISMS IN FPN-BASED CNNS

While deep learning models have significantly advanced player detection in sports analytics, accurately identifying the football remains a persistent challenge due to its small size, rapid movement, frequent occlusions, and visual similarity to other elements such as player socks, logos, and field markings. This limitation significantly reduces the effectiveness of automated systems in comprehensively analyzing football matches, particularly in applications such as tactical event recognition, shot classification, and game state prediction. In this paper, we propose a method to improve ball detection accuracy in football videos by enhancing an existing architecture based on Feature Pyramid Networks (FPN). The original FPN-based model, although efficient for detecting large-scale players, shows limited performance in detecting small objects such as the ball. To address this, we integrate lightweight attention mechanisms to help the model focus on more relevant spatial and semantic features. Specifically, we introduce Squeeze-and-Excitation (SE) layers into the backbone of the network to perform channel-wise feature recalibration and embed a Convolutional Block Attention Module (CBAM) into the ball detection head to refine both spatial and channel-level attention. These modifications are designed to enhance the network’s ability to distinguish the ball from cluttered backgrounds and visually similar objects. Our experiments, conducted on the ISSIA-CNR and Soccer Player Detection datasets, demonstrate that the proposed attention-augmented model achieves improved ball classification accuracy compared to the baseline, with no degradation in player detection performance. These results validate the utility of lightweight attention mechanisms in the context of small object detection and provide a promising direction for more robust and real-time football video analysis systems.

Keywords: Ball Detection, Deep learning, Football Video Analysis, Object Detection, Attention mechanisms, Feature Pyramid Network

Introduction

Ball detection plays a crucial part in the automated analysis of football matches, enabling advanced tasks such as event detection, match analysis, and performance assessment [1] [2]. However, accurate ball detection remains challenging because of its small size, fast movement, occlusions, and similar appearance to other elements, such as player socks, goalkeeper gloves, or field lines [3] [4]. While deep learning methods, especially convolutional neural networks (CNNs), have significantly advanced the state of object detection in sports analytics, existing approaches often struggle with reliably identifying the ball under diverse match conditions [5] [6].

Feature Pyramid Networks (FPN) are a promising approach to object detection in complex scenes [7]. They allow for effective multi-scale feature extraction by combining low and high-level features. A recent study proposed an FPN-based approach as an integrated ball and player detector in footage from football matches [3]. The approach demonstrated a strong performance in player detection. Nonetheless, the same approach showed comparatively lower accuracy in the ball detection tasks because of the small size, high speed, frequent occlusions, and visual similarity with other objects. As a result, even the state-of-the-art models for object detection, such as YOLO [8] and SSD [9], frequently missidentify or completely miss the detection of small, fast-moving targets [10] [11] [12]. This indicates a need for further refinement to increase the effectiveness of detecting small and fast-moving objects.

This paper addresses that specific limitation. We aim to enhance the ball detection performance of an existing FPN-based architecture by integrating lightweight attention mechanisms. Our approach is based on the recent success of applying an attention mechanism to improve the performance of small object detection in remote sensing [15], aerial imagery [16], and medical imaging [17]. Additionally, the idea of enhancing FPNs with attention is supported by the work Attentional Feature Pyramid Network (AFPN) proposed by Min et al. [18].

The remainder of this paper is organized in the following structure:

- Section 2 discusses related work on object detection in football and attention mechanisms.
- Section 3 provides the methodology, including the original architecture and our proposed enhancements.
- Section 4 describes the setup and the outcome of the experiments.
- Section 5 presents an analysis of the work.
- In conclusion, Section 6 summarizes the work and outlines future research directions.

Related Work

Ball Detection in Football Analytics. Object detection has become essential to football video analytics, helping recognize players, the ball, and key events such as shots [1] [2]. Traditional computer vision approaches relied on handcrafted features and motion tracking [5] but struggled in scenarios involving occlusion, fast motion, or cluttered backgrounds. With the help of deep learning, CNN-based methods have achieved better performance in sports analytics tasks.

Recent studies have employed architectures like YOLO [8] and SSD [9] for real-time player and ball detection. However, these models often struggle to detect small objects like the ball, especially in low-resolution frames or when the ball is partially occluded [5] [6]. The FPN-based base model used in this work represents an improvement by leveraging multi-scale feature maps, improving the detection of large and small objects [3] [19]. Despite this, the detection accuracy for the ball remained lower than for players, motivating further research into specialized enhancements.

Feature Pyramid Networks (FPN). The Feature Pyramid Network (FPN) [7], introduced by Lin et al., is a widely adopted architecture for multi-scale object detection. It enhances a backbone CNN (e.g., ResNet) by creating a

top-down pathway and lateral connections that fuse semantic-rich features from higher layers with detailed spatial features from earlier layers. FPN models are especially effective in object detection within the same image at different scales [10]. They are well-suited for complex scenes like football fields, where players and the ball vary in size and appearance. However, even with FPN's multi-scale approach, small objects like the ball can remain hard to detect due to weak spatial cues or low contrast. Some works, such as the Attentional Feature Pyramid Network (AFPN) [18], further enhance FPNs by introducing attention mechanisms to better focus on important features at multiple scales.

Attention Mechanisms in CNNs. Attention mechanisms are powerful tools that enhance feature representation in CNNs, emphasizing important information while suppressing irrelevant noise. Two modules used in our work are:

- Squeeze-and-Excitation (SE) blocks, proposed by Hu et al. [13], introduce channel-wise attention by modeling the interdependencies between feature channels. This allows the network to recalibrate the importance of different channels, leading to improved discriminative ability, especially in cluttered scenes.
- Convolutional Block Attention Module (CBAM), proposed by Woo et al. [14], extends this idea by incorporating channel and spatial attention. CBAM sequentially applies channel attention followed by spatial attention to refine the feature maps, making it particularly effective for tasks involving small and occluded objects.

Several studies have demonstrated that integrating SE or CBAM modules into standard CNNs improves performance across tasks such as remote sensing [15] [16], image classification, object detection [10] [11] [12], and segmentation [17]. However, their application to sports analytics, particularly for small object detection in dynamic environments, has been limited. In this paper, we explore the benefits of applying SE and CBAM to enhance the ball detection capability of an FPN-based network.

Methodology

In this section, we first describe the baseline architecture (FootAndBall) [3] that serves as the foundation for our work. Then, we present the proposed modifications that involve integrating attention mechanisms — Squeeze-and-Excitation (SE) [13] and Convolutional Block Attention Module (CBAM) [14] — to improve the detection of small, challenging objects such as a ball.

Integration of SE Block in Backbone. We add a Squeeze-and-Excitation (SE) [13] module after the first, third, and fifth convolutional blocks (Conv1, Conv3, and Conv5) in the backbone. The SE block works by performing global average pooling across each channel of the feature map, creating a channel descriptor that passes through two fully connected layers with a ReLU and sigmoid activation to learn the importance of each channel. The output is used to reweight the input feature map channels:

$$F_{scaled} = F \cdot \sigma\left(W_1 \cdot ReLU\left(W_2 \cdot GAP(F)\right)\right), \quad (1)$$

where F is the input feature map GAP is global average pooling, and W_1, W_2 are the learned weights. This allows the network to focus on informative feature channels and improve the representation of small objects [13] [20]. Fig. 1 represents a diagram of the SE block.

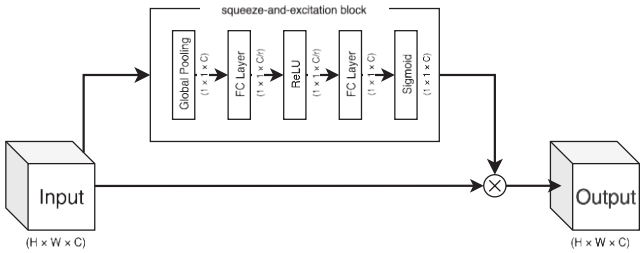


Fig. 1. Squeeze-and-excitation block

CBAM in Ball Classifier Head. The Convolutional Block Attention Module (CBAM) [14] enhances channel and spatial attention. We apply CBAM to the output feature map before the ball classification head. CBAM sequentially applies:

1. Channel attention uses average and max pooling along the spatial dimension followed by shared MLP layers.
2. Spatial attention, using a convolution over concatenated average-pooled and max-pooled feature maps across channels.

The schematic representation of the CBAM architecture is illustrated in Fig. 2.

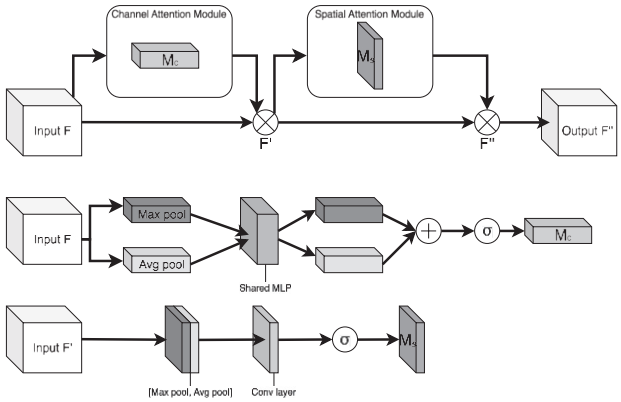


Fig. 2. Convolutional Block Attention Module (CBAM). The top diagram provides a general overview of the CBAM architecture. The middle diagram details the Channel Attention Module. The bottom diagram illustrates the Spatial Attention Module

This results in a refined feature map:

$$CBAM(F) = SA\big(CA(F)\big) \cdot F, \qquad (2)$$

$$CA(F) = \sigma\left(W_1\left(W_0\left(F^c_{avg}\right)\right) + W_1\left(W_0(F^c_{max})\right)\right), \qquad (3)$$

$$SA(F) = \sigma\left(f^{7\times7}\left(\left[F^s_{avg}, F^s_{max}\right]\right)\right), \qquad (4)$$

Modified Network Architecture Overview. The overall architecture remains fully convolutional and lightweight but with improved attention modeling. The SE-enhanced backbone generates richer feature maps, while the CBAM-augmented detection head improves ball localization precision. Fig. 3 illustrates the modified network architecture, where the SE modules are integrated into the first, third, and fifth convolutional blocks, and the CMAB module is integrated into the ball classification layer. A schematic comparison between the original and modified models is provided in Table 1.

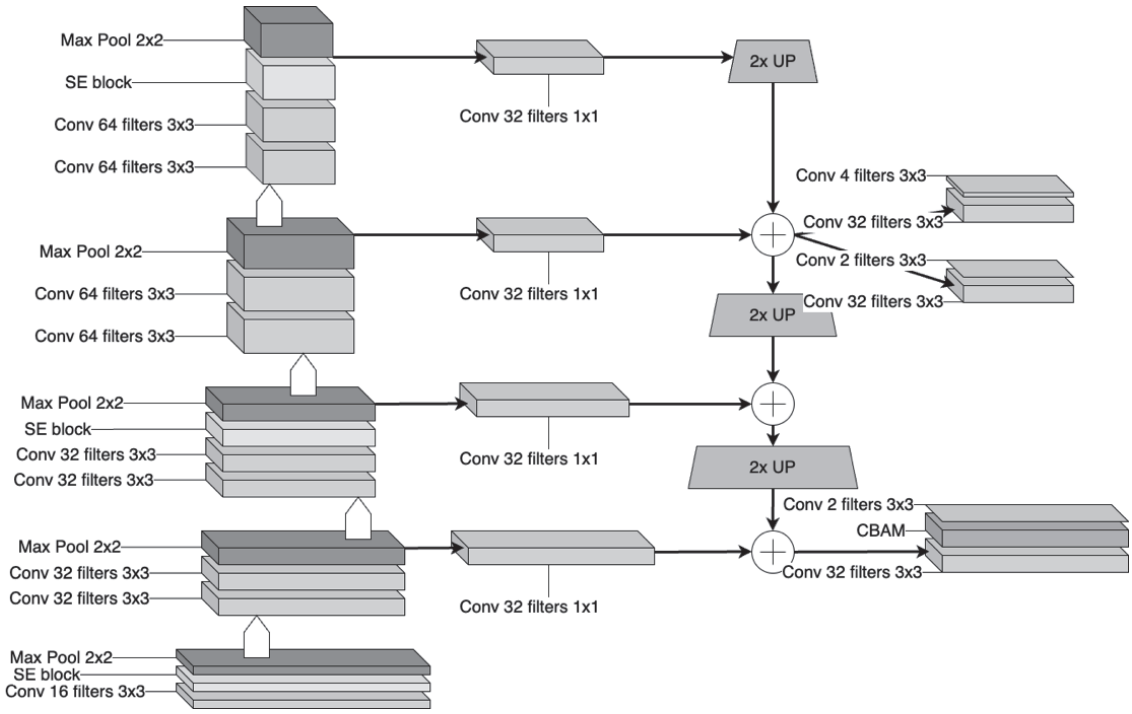


Fig. 3. The modified network architecture includes SE layers in blocks Conv1, Conv3, and Conv5, and a CBAM layer in the Ball classifier head

Table 1.

Comparison of original and modified network architectures

Block	FootAndBall layers	Modified Network Architecture layers	Output size
Conv1	16 filters 3x3 MaxPool 2x2	16 filters 3x3 SE block MaxPool 2x2	w/2, h/2, 16
Conv2	32 filters 3x3 32 filters 3x3 MaxPool 2x2	32 filters 3x3 32 filters 3x3 MaxPool 2x2	w/4, h/4, 32
Conv3	32 filters 3x3 32 filters 3x3 MaxPool 2x2	32 filters 3x3 32 filters 3x3 SE block MaxPool 2x2	w/8, h/8, 32
Conv4	64 filters 3x3 64 filters 3x3 MaxPool 2x2	64 filters 3x3 64 filters 3x3 MaxPool 2x2	w/16, h/16, 64
Conv5	64 filters 3x3 64 filters 3x3 MaxPool 2x2	64 filters 3x3 64 filters 3x3 SE block MaxPool 2x2	w/32, h/32, 64
1x1Conv1	32 filters 1x1	32 filters 1x1	w/4, h/4, 32
1x1Conv2	32 filters 1x1	32 filters 1x1	w/8, h/8, 32
1x1Conv3	32 filters 1x1	32 filters 1x1	w/16, h/16, 32
1x1Conv4	32 filters 1x1	32 filters 1x1	w/32, h/32, 32
Ball classifier	32 filters 3x3 2 filters 3x3 Sigmoid	32 filters 3x3 CBAM 2 filters 3x3 Sigmoid	w/4, h/4, 1
Player classifier	32 filters 3x3 2 filters 3x3 Sigmoid	32 filters 3x3 2 filters 3x3 Sigmoid	w/16, h/16, 1
BBox regressor	32 filters 3x3 4 filters 3x3	32 filters 3x3 4 filters 3x3	w/16, h/16, 4

Loss Function. We adopt the same loss function as in the original FootAndBall model, consisting of:

- Binary cross-entropy losses for ball and player classification.
- Smooth L1 loss for bounding box regression, as used in SSD [9] [21].

Let L_b, L_p, L_{bbox} represent the ball classification loss, player classification loss, and player bounding box loss, respectively. The total loss is computed as:

$$L = \frac{1}{N}\left(\alpha L_b + \beta L_p + L_{bbox}\right) \qquad (5)$$

where α and β are weighting coefficients, and N is the number of examples in a batch.

Experiments

In this section, we describe the experimental configuration used to evaluate the effectiveness of the proposed modifications to the FootAndBall architecture. We assess the performance of our proposed architecture, which integrates SE and CBAM modules, and compare it with the original model.

Datasets. We used the same two datasets that were used in the baseline study:

- ISSIA-CNR Soccer Dataset [5]: Contains 20,000 annotated frames from professional matches recorded using six synchronized Full HD cameras. Each frame is labeled with ball positions and player bounding boxes.
- Soccer Player Detection Dataset [22]: Composed of 2,019 images captured from two professional football matches, annotated with over 22,000 player locations. Ball positions are not annotated in this dataset.

As in the original paper, we split each dataset into 80% for training data and 20% for evaluation [3]. Both datasets contain a range of challenges like motion blur, occlusions, and background clutter.

Implementation details. We implemented the model in PyTorch and trained it using Adam optimizer [23] with a 4-step learning rate scheduler. The initial learning rate was set to 0.001 and decreased by a factor of 10 at the 10^{th}, 25^{th}, 50^{th}, and 75^{th} epochs. This gradual decay enabled the model to converge quickly in the early training phases and allowed a fine-grained adjustment in later stages. The training was performed on the NVIDIA RTX 4000 Ada Generation GPU. The hyperparameters for the training are represented in Table 2.

Table 2.

Training hyperparameters

Optimizer	Adam
Initial learning rate	0.001
Learning rate decay	x0.1 at epochs 10, 25, 50, and 75
Epochs	100
Batch size	16

To enhance the generalization, we applied data augmentation techniques, including random cropping and flipping [24].

Evaluation metrics. We use the Average Precision (AP) metric, a standard object detection metric described in the Pascal VOC challenge [25]. Ball detection AP is computed based on maximum values in the confidence map matching the ground truth position. Player detection is calculated based on predicted bounding boxes with an Intersection over Union (IoU) threshold of 0.5. We also report model size (number of trainable parameters) to evaluate efficiency.

Results. Table 3 compares the original model with the proposed enhanced version. We compare the Average Precision for Ball and Player detection on the ISSIA-CNR dataset and player detection on the Soccer Player Detection dataset.

Table 3.

Evaluation results of the original model in comparison with the enhanced model

Model	Ball AP	Player AP (ISSIA)	Mean AP	Player AP (SPD)	Params
FootAndBall	0.909	0.921	0.915	0.885	199K
SE + CBAM	0.927	0.917	0.922	0.871	200K

Our final model with SE and CBAM blocks shows the highest ball detection accuracy, outperforming the baseline by 2% AP gain. Player detection performance is also maintained at the same level. Despite the added attention layers, the model remains lightweight with a slightly increased number of parameters. Fig. 4 illustrates a comparative analysis of classification outcomes between the original and proposed models, highlighting instances where the original results are inadequate. In contrast, the proposed model successfully classifies the ball, demonstrating its enhanced efficacy.

Fig. 4. Comparison of ball classification results: the top row displays failed classifications from the original model, while the bottom row illustrates successful classifications from the proposed model

Discussion

The experimental results show that the integration of Squeeze-and-Excitation (SE) [13] and Convolutional Block Attention Module (CBAM) [14] blocks into the FootAndBall architecture improves the performance of the model for the task of ball detection. This is a significant advancement, as accurate ball detection remains one of the most complicated tasks in football video analysis owing to its small size, frequent occlusions, motion blur, and visual similarity to player gear and background elements [3] [4].

Adding SE blocks in the backbone enhances the model's ability to emphasize informative feature channels while suppressing less relevant ones. This aligns with prior findings that SE improves model sensitivity to subtle visual cues in cluttered scenes [13] [20]. In our case, the SE-enhanced backbone produces stronger features for ball detection. Similar channel-wise recalibration strategies have also proven effective in other small object detection contexts, such as traffic sign detection [11].

Including CBAM in the ball detection head applies channel and spatial attention. It allows the model to focus on small spatial re-

gions with high semantic relevance, such as regions that contain fast-moving objects. This spatial attention appears to help to distinguish false positives like white socks, pitch lines, or advertisements from the ball distractors, which is one of the frequent issues in the baseline model.

Combining SE and CBAM yields the highest accuracy, confirming their complementary nature. SE enhances global channel interactions during feature extraction, while CBAM introduces localized attention refinements before detection [14]. Similar hybrid attention strategies have succeeded in medical image analysis [17] and aerial image object detection [15] [16], where high-level semantics and spatial precision are critical.

Despite the additional attention layers, the proposed model remains comparably small and capable of real-time performance. This echoes trends in lightweight attention integration found in mobile-focused detection models like MobileNetV3 [26]. Our enhancements increased detection accuracy without a significant trade-off in model size.

However, some challenges remain. The model occasionally fails in edge cases involving heavy occlusion or extreme motion blur, conditions common in real-world sports footage. Fig. 5 illustrates challenging frames where the model either could not detect the ball or incorrectly identified it in its absence. Because our system processes frames independently, it cannot exploit temporal continuity to reinforce uncertain predictions. Techniques such as temporal feature aggregation or recurrent modules have been shown to improve consistency [27] [28] in video-based detection tasks and could be beneficial here.

(a) *(b)*

Fig. 5. Examples of model misidentifications, showing a false positive detection (a) and missed detections (b) of the ball

Overall, the results support our hypothesis that attention mechanisms considerably enhance the detection of small, context-sensitive objects in sports videos. The proposed approach balances accuracy and computational efficiency, making it suitable for real-time sports analytics systems.

Conclusion

This paper presents an enhanced deep learning architecture for joint player and ball detection in football match videos. Building on the original FootAndBall model, we introduce two attention mechanisms — Squeeze-and-Excitation (SE) [13] and Convolutional Block Attention Module (CBAM) [14] — to enhance the accuracy of ball detection, a task known to be difficult due to its small size, high motion, and frequent occlusion [4].

By integrating SE blocks into the feature extraction backbone, we enabled the network to adaptively recalibrate channel-wise feature responses adaptively, enhancing its discriminative power in complex scenes [13] [20]. Additionally, incorporating CBAM into the ball detection head improved the network's ability to focus on relevant spatial regions, significantly increasing its precision in identifying the ball amidst cluttered backgrounds. We also proposed a 4-step learning rate schedule, which helped improve training stability and convergence over time.

Our experiments on the ISSIA-CNR [5] and Soccer Player Detection [22] datasets demonstrated that the proposed attention-based enhancements lead to notable improvements in detection accuracy, particularly for the ball, increasing the accuracy by 2%, while maintaining real-time inference speed and model efficiency. These results validate the effectiveness of lightweight attention modules in sports video analysis systems.

While the proposed model achieved strong results, several opportunities for further improvement exist, such as temporal modeling. Our current approach operates on single frames, without leveraging temporal consistency. Incorporating temporal information through optical flow, frame-level feature aggregation, or recurrent networks (e.g., ConvLSTM or 3D CNNs) could enhance robust-

ness, especially in motion blur or occlusion scenarios [27] [28].

Overall, our results highlight that attention mechanisms are a promising avenue for improving small-object detection in sports analytics. The proposed system offers a solid foundation for future research and real-world applications in football match analysis by combining architectural innovation with efficiency considerations.

References

1. Bialkowski, P. Lucey, P. Carr, Y. Yue, S. Sridharan and I. Matthews, "Large-Scale Analysis of Soccer Matches Using Spatiotemporal Tracking Data," in *2014 IEEE International Conference on Data Mining*, December 2014. doi: 10.1109/ICDM.2014.133
2. M. Manafifard, H. Ebadi and H. Moghaddam, "A Survey on Player Tracking in Soccer Videos. Computer Vision and Image Understanding," *Computer Vision and Image Understanding,* vol. 159, pp. 19-46, June, 2017. doi: 10.1016/j.cviu.2017.02.002
3. J. Komorowski, G. Kurzejamski and G. Sarwas, "FootAndBall: Integrated Player and Ball Detector," in *15th International Conference on Computer Vision Theory and Applications*, pp. 47-56, Valletta, Malta, January, 2020. doi: 10.5220/0008916000470056
4. P. Kamble, A. Keskar and K. Bhurchandi, "A deep learning ball tracking system in soccer videos," Opto-Electronics Review, vol. 27, no. 1, pp. 58-69, March, 2019. doi: 10.1016/j.opelre.2019.02.003
5. T. D'Orazio, M. Leo, N. Mosca, P. Spagnolo and P. L. Mazzeo, "A Semi-automatic System for Ground Truth Generation of Soccer Video Sequences," in Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, Genova, Italy, September, 2009. doi: 10.1109/AVSS.2009.69
6. T. Wang and T. Li, "Deep Learning-Based Football Player Detection in Videos," Computational Intelligence and Neuroscience, pp. 1-8, 2022. doi: 10.1155/2022/3540642
7. T. -Y. Lin, P. Dollár, R. Girshick, K. He, H. B. and S. Belongie, "Feature Pyramid Networks for Object Detection," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 936-944, Honolulu, HI, USA, 2017. doi: 10.1109/CVPR.2017.106
8. J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv:1804.02767, 2018. doi: 10.48550/arXiv.1804.02767
9. W. Liu, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu and A. Berg, "SSD: Single Shot MultiBox Detector," in European Conference on Computer Vision, pp 21–37, 2016. doi: 10.1007/978-3-319-46448-0_2
10. Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li and S. Hu, "Traffic-Sign Detection and Classification in the Wild," in Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, June, 2016. doi: 10.1109/CVPR.2016.232
11. Y. Chen, J. Wang, Z. Dong, Y. Yang, Q. Luo and M. Gao, "An Attention Based YOLOv5 Network for Small Traffic Sign Recognition," in IEEE 31st International Symposium on Industrial Electronics (ISIE), Anchorage, AK, USA, June, 2022. doi: 10.1109/ISIE51582.2022.9831717
12. S. Du, W. Pan, N. Li, S. Dai, B. Xu, H. Liu, C. Xu and X. Li, "TSD‐YOLO: Small traffic sign detection based on improved YOLO v8," ET Image Processing, vol. 18, June, 2024. doi: 10.1049/ipr2.13141
13. J. Qu, Z. Tang, L. Zhang, Y. Zhang and Z. Zhang, "Remote Sensing Small Object Detection Network Based on Attention Mechanism and Multi-Scale Feature Fusion," Remote Sensing, vol. 15, p. 2728, May, 2023. doi: 10.3390/rs15112728
14. J. Rabbi, N. Ray, M. Schubert, S. Chowdhury and D. Chao, "Small-Object Detection in Remote Sensing Images with End-to-End Edge-Enhanced GAN and Object Detector Network," Remote Sensing, vol. 12, p. 1432, April, 2020. doi: 10.3390/rs12091432
15. O. Oktay, J. Schlemper, L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Hammerla, B. Kainz, B. Glocker and D. Rueckert, "Attention U-Net: Learning Where to Look for the Pancreas," arXiv:1804.03999, April, 2018. doi: 10.48550/arXiv.1804.03999
16. K. Min, G.-H. Lee and S.-W. Lee, "Attentional feature pyramid network for small object detection," Neural Networks, vol. 155, p. 439–450, November, 2022. doi: 10.1016/j.neunet.2022.08.029
17. V. Renò, N. Mosca, R. Marani, M. Nitti and E. Stella, "Convolutional Neural Networks Based Ball Detection in Tennis Games," in IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, June, 2018. doi: 10.1109/CVPRW.2018.00228

18. J. Hu, L. Shen and G. Sun, "Squeeze-and-Excitation Networks," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, June, 2018. doi: 10.1109/CVPR.2018.00745
19. S. Woo, J. Park, J.-Y. Lee and I. Kweon, "CBAM: Convolutional Block Attention Module," in European Conference on Computer Vision (ECCV), Munich, Germany, 2018.
20. H. Li, P. Xiong, J. An and L. Wang, "Pyramid Attention Network for Semantic Segmentation," 10.48550/arXiv.1805.10180, pp. 3-19, September, 2018. doi: 10.1007/978-3-030-01234-2_1
21. R. Girshick, "Fast R-CNN," in 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, December, 2015. doi: 10.1109/ICCV.2015.169
22. K. Lu, J. Chen, J. Little and H. He, "Light Cascaded Convolutional Neural Networks for Accurate Player Detection," 10.48550/arXiv.1709.10230, September, 2017. doi: 10.48550/arXiv.1709.10230
23. D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in International Conference on Learning Representations, December, 2014. doi: 10.48550/arXiv.1412.6980
24. C. Shorten and T. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," Journal of Big Data, vol. 6, no. 60, July, 2019. doi: 10.1186/s40537-019-0197-0
25. M. Everingham, L. Van Gool, C. Williams, J. Winn and A. Zisserman, "The Pascal Visual Object Classes (VOC) challenge," International Journal of Computer Vision, vol. 88, pp. 303-338, 2010. doi: 10.1007/s11263-009-0275-4
26. A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. Le and H. Adam, "Searching for MobileNetV3," 10.48550/arXiv.1905.02244, pp. 1314-1324, Seoul, Korea (South), 2019. doi: 10.1109/ICCV.2019.00140
27. L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang and L. Van Gool, "Temporal Segment Networks for Action Recognition in Videos," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 11, pp. 2740-2755, November, 2019. doi: 10.1109/TPAMI.2018.2868668
28. A. Kompella and R. Kulkarni, "A semi-supervised recurrent neural network for video salient object detection," Neural Computing and Applications, pp. 2065–2083, vol. 33, no. 6, March 2021. doi: 10.1007/s00521-020-05081-5

Одержано: 20.05.2025
Внутрішня рецензія отримана: 30.05.2025
Зовнішня рецензія отримана: 30.05.2025

Про авторів:

[1,2]Івасенко Ірина Богданівна,
доктор технічних наук,
старший науковий співробітник,
професор
e-mail: iryna.b.ivasenko@lpnu.ua
https://orcid.org/0000-0003-3795-9779

[2]Бішир Сергій Сергійович,
аспірант першого року навчання,
Національний університет
«Львівська політехніка»
e-mail: serhii.s.bishyr@lpnu.ua
https://orcid.org/0009-0009-1008-9292

Місце роботи авторів:
[1]Фізико-механічний інститут
Ім. Г. В. Карпенка НАН України
Тел.: +3(032) 263-30-88
79060, м. Львів, вул. Наукова 5,
e-mail: pminasu@ipm.lviv.ua

[2]Національний університет
«Львівська політехніка»
Тел.: +3(8032) 258-22-82,
79013, м. Львів, вул. Степана Бандери, 12,
e-mail: coffice@lpnu.ua, com.centre@lpnu.ua

УДК 004.8:004.724 https://doi.org/10.15407/pp2025.02.063

І.П. Сініцин, Ю.В. Рогушина, К.Ю. Юрченко

ІНТЕГРАЦІЯ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ ІЗ ЗАСОБАМИ СЕМАНТИЧНОЇ ОБРОБКИ ЯК ІНСТРУМЕНТ ЦИФРОВІЗАЦІЇ ЗНАНЬ

У роботі розглядається задача автоматизації аналізу, генерації та управління складними природномовними документами на основі інтеграції генеративного штучного інтелекту із семантичними технологіями, зокрема, Semantic MediaWiki. Аналізується, яким чином застосування онтологічних моделей предметних областей та семантичної розмітки дозволяє запобігати таким критичним недолікам великих мовних моделей, як схильність до "галюцинацій" (генерації неправдивих тверджень) та відсутність прозорості у поясненні рішень.

Така інтеграція досліджується на прикладі інструментальної системи “ЛІНЗА”, яка розробляється для автоматизованої інтелектуальної обробки контенту розрізнених документів зі складною слабоформалізованою структурою з метою генерації природномовних звітів за заданими вимогами у різних галузях, таких як публічне адміністрування, юриспруденція, цифровізації знань, сертифікація та стандартизація. Система базується на поєднанні гнучкості та адаптивності великих мовних моделей із формалізованими онтологічними знаннями та підтримкою семантичних запитів щодо пертинентних фактів у середовищі Semantic MediaWiki, або зовнішніх джерел (Retrieval-Augmented Generation). Запропонований підхід дозволить значно знизити ризики помилок, типових для генеративних моделей, та забезпечити фактичну правдивість і прозорість процесу ухвалення рішень. Особлива увага приділяється механізмам прозорості, достовірності та можливості контролю людиною для підвищення довіри до згенерованих даних, що особливо важливо у сферах із підвищеними вимогами до безпеки інформації. Такий підхід також забезпечує більшу довіру до автоматично створених документів.

Багаторівнева архітектура системи характеризує задачі агентів і сервісів, що виконують спеціалізовані функції збору, аналізу, перетворення та перевірки даних, і забезпечує гнучкість, масштабованість та адаптивність системи до зміни вхідних даних і вимог.

Ключові слова: агентні технології, великі мовні моделі, LLM, Semantic MediaWiki, семантичні технології, база знань, формалізовані документи.

I.P. Sinitsyn, Yu.V. Rogushina, K.Yu. Yurchenko

INTEGRATION OF LARGE LANGUAGE MODELS WITH SEMANTIC PROCESSING TOOLS AS AN INSTRUMENT FOR KNOWLEDGE DIGITIZATION

The paper addresses the task of automating the analysis, generation, and management of complex natural language documents based on the integration of generative artificial intelligence with semantic technologies, in particular Semantic MediaWiki. It analyzes how the use of ontological models of subject domains and semantic markup makes it possible to prevent such critical shortcomings of large language models as the tendency to “hallucinations” (generation of false statements) and the lack of transparency in decision explanations.

This integration is explored using the example of the instrumental system “LINZA,” which is being developed for automated intelligent processing of content from heterogeneous documents with complex and weakly formalized structure, with the aim of generating natural language reports according to specified requirements in various domains, such as public administration, jurisprudence, certification, and standardization. The system is based on the combination of the flexibility and adaptability of large language models with formalized ontological knowledge and support for semantic queries about pertinent facts in the Semantic MediaWiki environment or external sources (Retrieval-Augmented Generation). The proposed approach will significantly reduce the risks of typical errors in generative models and ensure factual accuracy and transparency in the decision-making process.

Special attention is paid to mechanisms of transparency, reliability, and the possibility of human control to increase trust in the generated data, which is especially important in areas with high information security requirements, and ensures greater confidence in automatically created documents.

The multi-level architecture of the system defines the tasks of agents and services that perform specialized functions of data collection, analysis, transformation, and verification, and ensures flexibility, scalability, and adaptability of the system to changes in input data and requirements.

Keywords: agent technologies, large language models, LLM, Semantic MediaWiki, semantic technologies, knowledge base, formalized documents.

Вступ

Сучасні інформаційні системи дедалі частіше застосовують технології *генеративного штучного інтелекту* (ГШІ) для автоматизації рутинних завдань, зокрема, створення, підтримки та адаптації документів. У цьому контексті особливої важливості набуває розробка гібридних платформ, що поєднують не лише швидке, а й семантично узгоджене та перевірене генерування документів на основі спеціального підкласу ГШІ – великих мовних моделей (Large Language Models, LLM) з перевіреними структурами збереження знань [1].

Більшість наявних рішень, які зараз використовують штучний інтелект для автоматизованого формування документації, обмежуються використанням шаблонів або спрощених структур для опису потрібних документів [2]. Але розвиток LLM-моделей дозволяє створювати потужніші системи зі значно ширшим функціоналом, здатні генерувати комплексні, формалізовані документи з поясненням логіки ухвалених рішень.

Значна частина чинних систем на базі LLM базуються на графах знань або векторних базах, що забезпечують лише часткову семантичну підтримку [3]. Однак такі підходи мають суттєві обмеження: вони не завжди здатні пояснити результати генерації, а використання LLM без контролю призводить до ризику появи "галюцинацій" — некоректних або вигаданих даних [4].

Попри значні досягнення у галузі обробки *природної мови* (ПМ), сучасні LLM все ще виявляють певні слабкі сторони, особливо в роботі зі спеціалізованими предметними областями (ПрО), що використовують специфічну термінологію та правила побудови документів.

Це пояснюється тим, що робота LLM базується на виявленні статистичних закономірностей у великих обсягах даних, а для таких специфічних ПрО обсяг даних для обробки може бути недостатнім (або ж інформація, релевантна до цієї ПрО, не виокремлена з усього масиву даних, що аналізуються). Використання статистичних моделей дозволяє LLM ефективно генерувати стилістично коректні тексти, проте не завжди забезпечує фактичну правдивість та прозорість процесу ухвалення рішень. Це може призвести до значних перешкод у застосуванні LLM в тих сферах, де потрібен високий ступінь довіри, аудит та можливість експертної інтервенції, зокрема, у сфері підвищеної секретності.

З огляду на зазначені обмеження, виникає об'єктивна необхідність у доповненні можливостей LLM інструментами менеджменту знань. Це передбачає створення гібридної архітектури, яка поєднує різні технології обробки інформації для досягнення синергетичного ефекту. LLM можуть ефективно використовуватися для первинного аналізу великих обсягів неструктурованих текстових даних, ідентифікації ключових сутностей та формування початкових версій документів. Системи управління знаннями, такі як SMW, забезпечать формалізоване представлення витягнутих даних, дозволяючи їх структуроване зберігання, легке редагування, валідацію експертами та побудову чітких логічних висновків.

Формулювання задачі

У роботі аналізується доцільність інтеграції семантичних технологій із великими мовними моделями LLM з метою автоматизованого створення документів складної структури на основі гетерогенних даних – природномовних документів, таблиць, баз даних та знань, онтологій та тезаурусів ПрО, мультимедійної інформації тощо. А також вимог щодо подання результатів аналізу та відомостей про користувача. Результуючі документи мають відповідати формалізованим вимогам щодо складу, логіки побудови та оформлення, що визначаються нормативними або внутрішніми регламентами. Прикладами таких задач є побудова спеціального профілю захищеності інформаційної системи, уза-

гальнення досвіду діяльності в певній сфері та його впровадження, генерація персональної траєкторії навчання для здобувача освіти.

Складність кожної конкретної задачі залежить від ступеня формалізованості вхідних даних (особливо – вимог до результату); складності правил, за якими елементи вхідних даних перетворюються на елементи результуючих документів, та від обсягу бази знань ПрО, яка потрібна для побудови цих правил. Але, незалежно від цього, всі подібні задачі потребують однакового набору операцій над вхідними даними (складність задачі впливає лише на час аналізу), і тому для їх розв'язання доцільно створити універсальне інструментальне середовище, що підтримує весь потрібний функціонал. З огляду на те, що набір задач може розширюватися, а інформаційні потреби користувачів – ускладнюватись, доцільно передбачити можливості гнучкого розширення архітектури такої системи, яка дозволить поповнювати її новими модулями без змін вже існуючих.

У найбільш узагальненому вигляді ця задача має наступний вигляд: вхідними даними для аналізу є: 1) набір документів, що містить знання щодо ПрО, – як семантично формалізовані (тезауруси, онтології, бази знань), так і слабо формалізовані (природномовні описи, стандарти, різноманітні сирі дані, проаналізовані приклади тощо); 2) індивідуальні дані користувача, які конкретизують його інформаційні потреби, в тому числі – специфічні вимоги, правила оформлення контенту та термінології; 3) засоби визначення вимог до результуючих документів – формалізовані вимоги, природномовні описи, приклади.

В результаті обробки треба згенерувати документи, структура яких відповідає визначеним вимогам, а контент характеризує вказані користувачем об'єкти із застосуванням знань щодо ПрО. Відповідно до специфіки ПрО, результуючими документами можуть бути спеціалізовані профілі організацій або інформаційних систем; рекомендації, що узагальнюють відомості з сирих даних; аналітичні огляди технічної та наукової інформації, структуровані за певними правилами; оцінки діяльності підрозділів організацій тощо. Для цього необхідно розв'язати низку підзадач: виокремлення структури результуючого документа на основі наявних прикладів та описів, зв'язування на рівні семантики елементів контенту цього документа з відомостями ПрО, співставлення інформації від користувача з терміносистемою ПрО тощо. Найбільш складними елементами є коректне розпізнавання фрагментів документації, що відповідають слабо формалізованим вимогам до результату, та аналіз семантичної коректності отриманого результату.

Важливо розуміти, що використання лише логічного виведення та семантичних запитів є недостатнім для здобуття знань з ПМ-документів, а передача задачі в цілому до LLM (навіть з великою кількістю попередніх налаштувань та значною кількістю ітерацій) виявляється занадто складною для аналізу якості отриманого результату та виокремлення причин, що призвели до некоректних результатів. Тому доцільно інтегрувати в одній інформаційній системі обидві можливості, доповнивши їх гнучкими засобами менеджменту інформації – як на рівні документів, так і на рівні знань.

Але потрібно відзначити, що просте механічне поєднання одного технологічного середовища LLM та Semantic MediaWiki не вирішує поставлене завдання: потрібно чітко визначити етапи обробки інформації, формати збереження даних та моделі обміну між окремими модулям, передбачити засоби керування та зворотного зв'язку.

Така система має забезпечити функціонально повний набір сервісів для перетворення вхідної інформації на результуючий набір документів, що відповідають вимогам користувача. Технологія використання системи повинна визначати порядок застосування сервісів, коректні перетворення інформації та можливість контролю людиною всіх етапів таких перетворень з метою вчасного виявлення та запобігання семантичним неоднозначностям (рис.1).

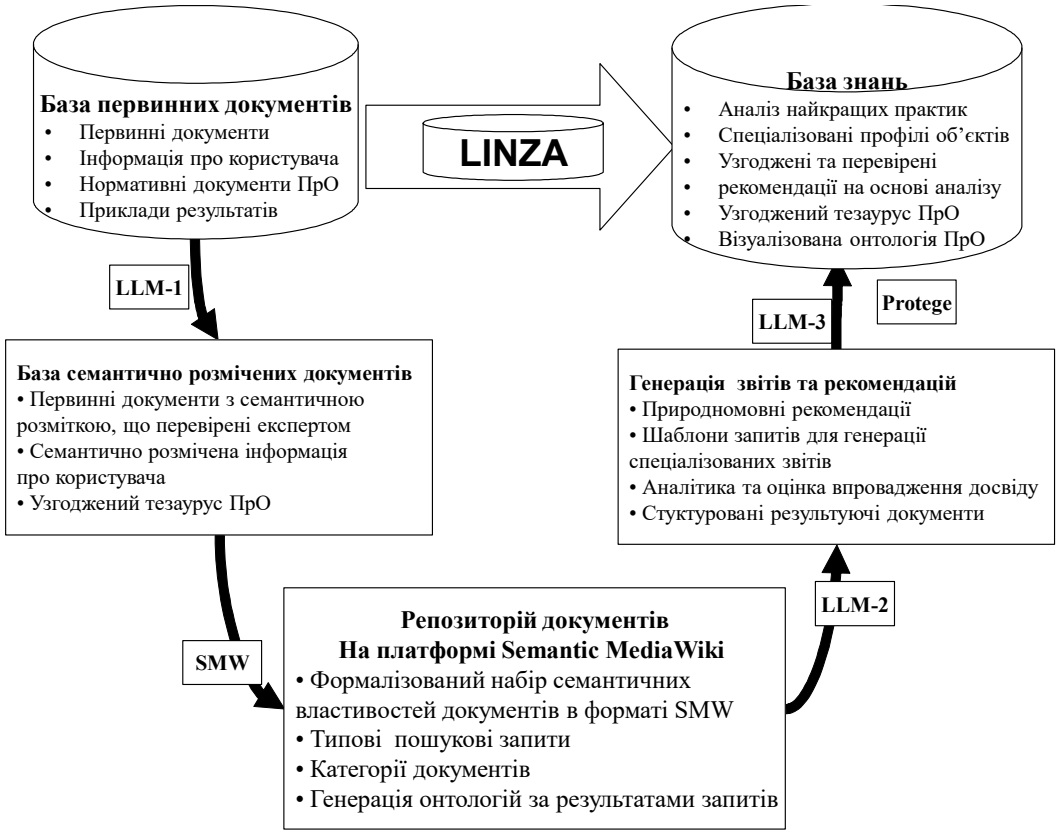


Рис. 1. Узагальнена схема технологічного середовища “Лінза”

На попередньому етапі розробки цього інтегрованого технологічного середовища потрібно чітко визначити його призначення та базовий функціонал, формалізувати основні види перетворень інформації в процесі обробки. Саме це дозволяє визначити як склад цього середовища, так і призначення окремих модулів і засобів взаємодії між ними та користувачами, охарактеризувати необхідні операції у життєвому циклі інформації (Рис.2).

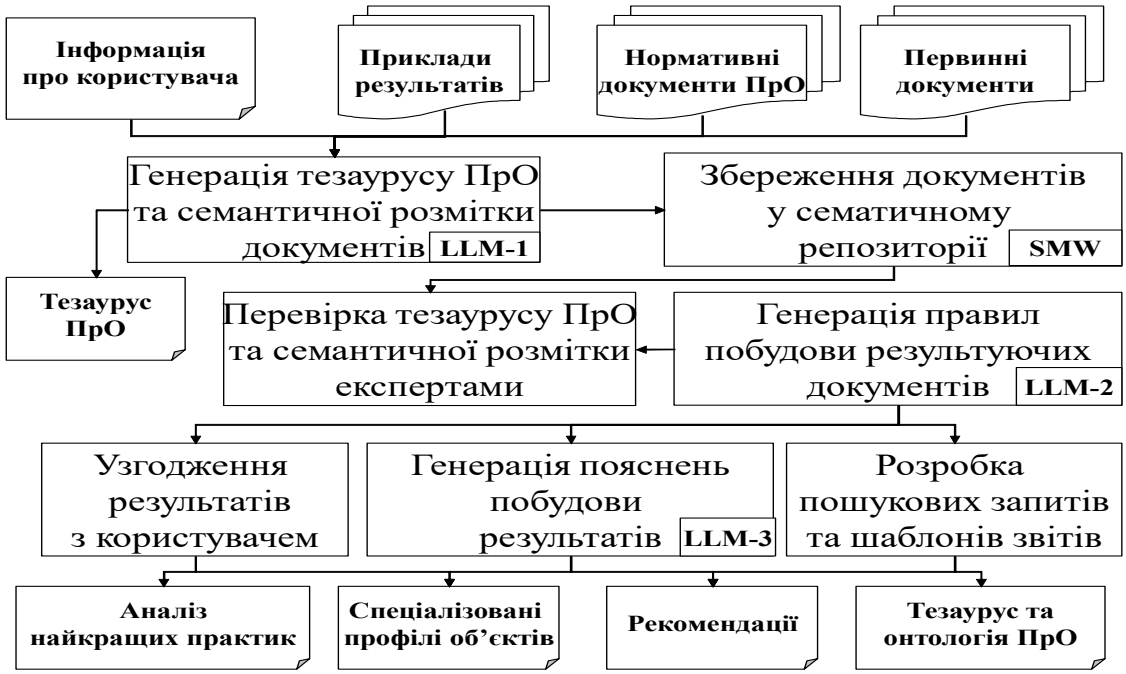


Рис. 2. Життєвий цикл інформації у технологічному середовищі “Лінза”

Для цього потрібно формалізувати вимоги до системи, яка має забезпечити автоматизацію процесу формування таких документів на основі поєднання генеративних можливостей LLM з контрольованістю та прозорістю семантичних платформ, а саме: визначити базові принципи побудови такої системи, її компоненти, механізми інтеграції джерел знань і способи забезпечення відповідності результату очікуваним формальним критеріям.

Теоретичні засади системи аналізу документів "Лінза"

Ключем до успішної реалізації гібридної системи є детермінація та структуризація вхідних і вихідних інформаційних потоків. Вхідні дані в таких задачах здебільшого представлені масивними колекціями неструктурованих чи напівструктурованих документів із ПрО. До них належать нормативні документи та стандарти, зокрема, у сферах з обмеженим або контрольованим доступом до інформації, та емпіричні описи конкретних систем. Ці джерела, як правило, представлені у вільному форматі та містять складні внутрішні структури, термінологію та взаємозв'язки, що вимагають глибокого семантичного аналізу. Обробка цих вхідних даних здійснюється за допомогою інтегрованих LLM та спеціалізованих сервісів, які виконують функції парсингу, ідентифікації ключових сутностей, виявлення реляційних зв'язків та первинної семантичної розмітки.

Вихідними даними системи є формалізовані та структуровані знання, що зберігаються у базі знань на платформі SMW, а також кінцеві згенеровані документи. Зокрема, до вихідних даних належать: семантично збагачені структури знань, де витягнута інформація трансформується у формат, сумісний з SMW (тріади, властивості та класифікації), забезпечуючи високий рівень інтероперабельності та можливість логічного виведення. Зокрема, система продукує верифіковані фрагменти або цілі документи складної структури, правдивість яких гарантується етапом валідації за участю експертів (human-in-the-loop). Також до вихідних даних належать ланцюги логічного виведення та детальні пояснення ухвалених рішень, включаючи ідентифікацію підмножини знань, використаної моделлю, що забезпечує прозорість та можливість аудиту в умовах роботи з конфіденційною інформацією. Таким чином, відбувається процес трансформації неструктурованих вхідних даних у структуровані, верифіковані та пояснювані вихідні знання, що є центральним аспектом функціонування системи, забезпечуючи її високу цінність для автоматизації складних процесів обробки інформації.

Не менш важливим є пошук ефективних технологічних рішень, зокрема, застосування агентів та сервісів для цілісного та безперервного перетворення інформації між різними компонентами системи та визначення надійних джерел отримання знань щодо ПрО. У контексті поточної роботи предметна область охоплює документи і стандарти, приклади конкретних систем та їхній опис. Критично важливим аспектом є розуміння специфіки задачі, зокрема питання щодо допустимості розміщення опису предметної області у відкритому доступі, оскільки передача чутливих відомостей до зовнішніх LLM є неприпустимою з огляду на інформаційну безпеку та конфіденційність. Це додатково доводить доцільність локального розгортання та інтеграції компонентів і застосування концепції "human-in-the-loop" для контролю та валідації.

Метою роботи є аналіз доцільності інтеграції семантичних технологій з LLM для автоматизованого створення комплексних природномовних документів, що включають текстові, графічні та табличні дані, відповідають формальним вимогам до структури та враховують специфічні запити замовника.

Аналіз конкретних прикладних задач та шляхів їх розв'язання спрямований на те, щоб дослідити вимоги до такої автоматизованої системи. Запропонований підхід має дозволити розв'язати проблему довіри та достовірності результатів, притаманні сучасному генеративному ШІ.

Дослідивши сучасний стан розробок у сфері інтелектуальної обробки інформації, ми виявили доцільність застосовувати у "Лінзі"**:**

- Агентні технології та сервіс-орієнтоване програмування [5] – для персоніфікованої динамічної обробки даних у веб-середовищі: інтелектуальні агенти дозволяють відображати цілі та задачі різних суб'єктів системи, щоб знаходити та активувати найбільш прийнятні сервіси для перетворення та аналізу інформації;
- Семантичні технології та онтологічний аналіз [6] – для семантичної інтерпретації контенту вхідних даних системи з використанням зовнішніх джерел знань (онтологій, тезаурусів, таксономій), їх структурування формалізованими мовами для інтероперабельності та автоматичної обробки;
- Великі мовні моделі – як засіб глибокого семантичного аналізу природномовних документів для співставлення їхнього контенту з онтологічними моделями ПрО класифікації та структурування (наприклад, LLM дозволяють автоматизовано генерувати семантичну розмітку сирих природномовних даних тегами, що відповідають поняттям та відношенням з обраної онтології).
- Семантичні вікі [7] – як платформи для зберігання та надання користувачам доступу до семантично структурованого контенту.

Сучасний стан досліджень у сфері інтеграції LLM із семантичними технологіями

У більшості сучасних інтелектуальних систем, що базуються на LLM, користувач позбавлений доступу до процесів структурування знань, джерел первинного контенту та логіки формування рекомендацій. Алгоритми обробки залишаються непрозорими, що обмежує довіру до системи, ускладнює валідацію результатів та викликає труднощі у експертній - перевірці.[4]

У цьому контексті особливий інтерес становить застосування вікі-технологій, зокрема, таких, як Semantic MediaWiki та WikiData, у поєднанні з LLM. Саме вікі-підхід дозволяє зробити структуру знань відкритою, зрозумілою та доступною для редагування, що, своєю чергою, пояснює причини формування системних висновків і підвищує прозорість ухвалення рішень.

У науковій літературі представлено низку проєктів, у яких здійснюється інтеграція LLM із WikiData або Wikipedia [8, 9]. Їхній аналіз виділяє ряд закономірностей, характерних для мультиагентних систем із залученням LLM.

Більшість описаних рішень реалізують мультиагентну архітектуру, де кожен агент виконує окрему спеціалізовану функцію — від семантичного аналізу . до генерації тексту [3] та перевірки правдивості даних [4] Такий підхід дозволяє формалізувати робочий процес і значно знижує ризики помилок, типових для генеративних моделей. У всіх системах застосовується *Retrieval-Augmented Generation* (RAG) [10] — техніка, що дозволяє LLM поєднувати генеративні можливості з релевантною фактологічною інформацією із семантичних структур або зовнішніх джерел. У цих підходах активно використовують техніки покрокового уточнення запитів (workflow orchestration, prompt chaining), що наближає модель до логічного міркування й багатоетапної побудови контенту та забезпечує структуровану генерацію документів на основі формалізованих знань, зниженні частоти «галюцинацій» моделей, високій гнучкості в розподілі обов'язків між агентами і можливості адаптації під різні доменні задачі. Зокрема, використання семантичних шаблонів дозволяє LLM автоматично виводити інформаційні вимоги до доку-

мента, тоді як у CLAIR граф знань забезпечує багатоетапне логічне виведення фактів для підготовки технічної документації. DocAgent використовує багатогранну систему оцінювання згенерованого результату, що дозволяє об'єктивно оцінювати повноту, корисність та фактичність документації [9].

Водночас, існують і певні обмеження. Системи залишаються чутливими до обмежень контексту моделей — навіть при використанні довгих вікон (16K+ токенів) генерація в межах великих кодових баз або графів знань може втрачати релевантність. Деякі системи, зокрема, DocAgent, орієнтовані переважно на статичний аналіз, що обмежує обробку динамічних аспектів програмних систем. Ефективність генерації значною мірою залежить від якості семантичного опису — неповні або суперечливі шаблони можуть призводити до помилок або втрати важливої інформації. Також слід зазначити, що інтеграція з платформами на кшталт SMW потребує додаткових зусиль у побудові запитів та формалізації знань у придатному для LLM вигляді.

Крім того, у сучасних дослідженнях активно вивчаються підходи до поєднання великих мовних моделей із семантичними базами знань, зокрема, у форматі Wikidata. Одним з таких прикладів є LLM Store [11], що виступає як проміжне середовище між LLM та структурованими даними, дозволяючи трансформувати природномовні запити у твердження у форматі RDF-триад [12]. Архітектурно система реалізована як плагін до KIF (Knowledge Integration Framework), що забезпечує трансляцію відповідей моделі у формат, придатний до розміщення у Wikidata. Зокрема, її ефективність була продемонстрована в задачах LM-KBC (Language Model-based Knowledge Base Construction), де LLM Store показав високі результати точності генерації. Найвищих показників F1-score (до 91%) вдалося human-in-the-loop досягти шляхом додавання контексту до запитів, що підтверджує ключову роль модуля генерації контексту. Попри це, автори підкреслюють слабке місце системи — невисоку точність для "вузьких" специфічних відносин та помилки в ідентифікації сутностей.

В іншій системі — Scholarly Wikidata, LLM використовується для напівавтоматизованого вилучення метаданих наукових конференцій із веб-сайтів та текстів матеріалів. Результатом стало суттєве збагачення бази Wikidata: додано тисячі сутностей та нових властивостей, що охоплюють організаційні ролі, прийняті статті, доповіді, тощо. Перевагою цього підходу є висока ефективність витягування структурованої інформації, однак автори вказують на схильність LLM до помилок під час роботи з датами, назвами треків або складними залежностями між сутностями. Для таких випадків необхідне втручання людини (human-in-the-loop), що підвищує загальні витрати на підтримку системи [8].

Дослідження Каверинського, Літвіна та Палагіна [13] пропонують інноваційний підхід до керованої генерації природної мови. Основна ідея роботи полягає в концепції "зворотного синтезу", яка, на відміну від традиційного аналізу природної мови, зосереджена на продукуванні текстових висловлювань з наявних формалізованих онтологічних представлень. Ці онтологічні представлення, які є центральною ланкою методології, автоматично будуються на основі аналізу речень науково-технічних текстів за допомогою раніше розроблених програмних засобів. Вони інкорпорують сутності, ідентифіковані в тексті, та типізовані семантичні зв'язки між ними, формуючи таким чином структуровану мережу знань. Для взаємодії з великою мовною моделлю, зокрема ChatGPT, автори розробили спеціально структуровані інструкції-підказки (prompts), які направляють модель для генерації тексту, що точно відповідає заданій семантичній структурі. Проведена серія експериментів із синтезу природномовних висловлювань підтвердила ефективність запропонованого підходу. Ця методологія є розв'язанням проблем, пов'язаних із неконтрольованою генерацією та низькою прозорістю LLM, забезпечуючи значно вищий рівень правдивості та прозорості згенерованих висловлювань через

їхню опору на верифіковані онтологічні структури та керовані підказки.

У контексті інтеграції великих мовних моделей із семантичними технологіями, значний інтерес становить досвід застосування семантичних вікі-систем для управління знаннями. Зокрема, [14] описує використання MediaWiki та Semantic MediaWiki для створення онтологічно-орієнтованих репозиторіїв знань, що підтверджує доцільність застосування таких платформ для формалізації предметних областей, забезпечення однозначної інтерпретації термінології та підтримки семантичного пошуку у складних, динамічних інформаційних масивах. Запропонована архітектура, що інтегрує різні джерела знань та сервіси, слугує вагомим підґрунтям для розробки гібридних систем, які прагнуть поєднати генеративні можливості LLM із верифікованими семантичними структурами.

Отже, вище зазначені системи демонструють спільну тенденцію: використання великих мовних моделей як джерела знань, які потім зберігаються у формалізованій структурі типу Wiki. Проте ці моделі мають спільне обмеження — дані, які були згенеровані великими мовними моделями, стають частиною бази знань без механізмів прозорої верифікації. Крім того, наразі, запит у пошуковій системі не дає жодного релевантного посилання, що свідчить про новизну запропонованого напряму (“LLM and SMW”). Тож можна стверджувати, що поєднання нейронних моделей із SMW утворює синергетичну модель, де LLM виступає як інтерпретатор природної мови та генератор контенту, а SMW забезпечує структуроване середовище для верифікації, представлення та розширення знань. Такий підхід дозволяє забезпечити покращення для традиційних систем, підвищуючи точність і надійність згенерованих даних завдяки формалізованим семантичним шаблонам і механізмам перевірки.

Таким чином, системи на основі мовних моделей і семантичних структур демонструють значний потенціал для автоматизованої побудови та структурування даних у різних галузях — від публічного адміністрування до програмної інженерії. Подальший розвиток доцільно спрямувати на вдосконалення механізмів інтеграції з джерелами знань, покращення інтерпретованості результатів і розширення підтримки динамічних сценаріїв використання.

Система «ЛІНЗА»: архітектура, призначення та інтелектуальні механізми

Інформаційна система лінгвістичної обробки нормативних документів «ЛІНЗА» призначена для автоматизації створення та обробки складних природномовних документів, що інтегрують розрізнені дані та відповідають жорстким формальним вимогам.

Система є розробкою Інституту програмних систем НАН України та має значний потенціал застосування в тих сферах, де критично важлива точність, структура та правдивість інформації.

“ЛІНЗУ” доцільно застосовувати в тих галузях, де необхідно забезпечити семантичне структрування великих обсягів інформації з їх подальшим аналізом та перетворенням на документи зі складною, наперед визначеною структурою, а також генерації пояснюваної інформації щодо створених документів та забезпеченні високої правдивості даних. Водночас значна частина правил перетворення та принципів сруктурування подається неявно, тобто ці знання потрібно здобувати з відповідних документів, пов’язаних між собою. Сферами застосування системи можуть бути: державне управління, де вона може значно спростити підготовку нормативно-правових актів, звітів, протоколів та іншої офіційної документації; юриспруденція – для аналізу величезних обсягів юридичних документів, ефективного вилучення ключової інформації та автоматичного формування декларацій; сертифікація та стандартизація – як іструмент автоматизованої підготовки звітної документації на вимогу сертифікаційниих органів.

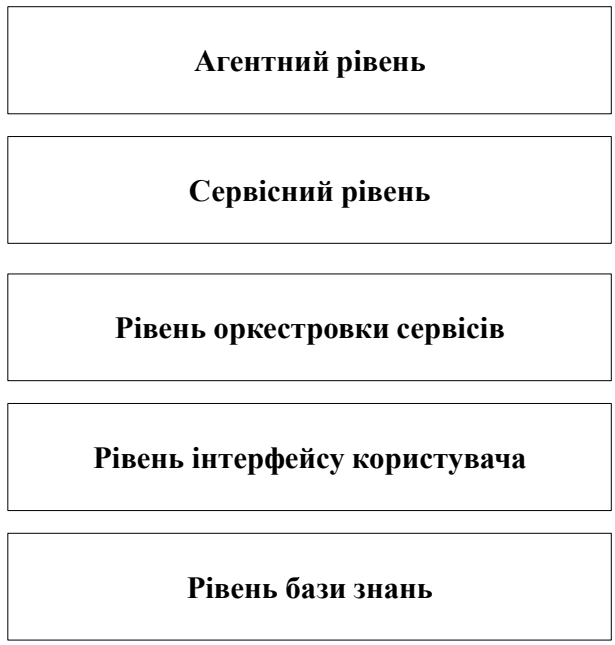


Рис. 3. Багаторівнева архітектура технологічного середовища “Лінза”

Можливість аналізу та реалізації досвіду для таких інтелектуальних систем, як "ЛІНЗА", має ключове знання. Це є основою для вдосконалення функціоналу системи: регулярний аналіз результатів її роботи, зворотний зв'язок від користувачів, експертів, що дозволяє виявляти недоліки та оптимізувати процеси. Такий підхід сприяє ефективному накопиченню знань, оскільки система дозволяє структурувати та зберігати отриманий досвід, перетворюючи його на структуровані онтології та інші формати, доступні для подальшого використання та аналізу. Крім того, у динамічних сферах, таких як захист інформації, де постійно виникають нові виклики та вимоги, постійні адаптації до змін є життєво необхідними.

“Лінза” є складною системою, що поєднує різні технології та методи обробки інформації. Тому для її розробки застосовується багаторівнева архітектура, яка дозволяє перетворити задачу розробки системи на набір простіших підзадач (Рис.3).

Взаємодія складових системи "ЛІНЗА" координується через сервіс оркестровки [15] (Orchestration Service), який координує інтелектуальну обробку документації за участі агентів і сервісів, що виконують спеціалізовані функції збору, аналізу, перетворення та перевірки даних. Такий підхід забезпечує гнучкість, масштабованість та адаптивність системи до зміни вхідних даних і вимог.

На рівні взаємодії з користувачем система реалізує веб-інтерфейс (UI - user interface), який уможливлює завантаження документів, формулювання запитів природною мовою та перегляд аналітичних результатів і згенерованих - декларацій.

Сервісний рівень виконує наступні функції: завантаження та попередню обробку документів, семантичне анотування, онтологічне моделювання, генерацію відповідей за допомогою LLM, та інтеграцію знань за підходом Retrieval-Augmented Generation (RAG). Для збереження, оновлення та валідації знань використовується Semantic MediaWiki, що дає можливість семантичного доступу через SPARQL-запити.

Такий архітектурний підхід дозволяє ефективно інтегрувати експертні знання з мовними моделями, що суттєво покращує якість аналізу, обґрунтування рішень та пояснення складних нормативних понять.

Функціональні можливості “Лінзи”

Функціональна архітектура системи "ЛІНЗА" інтегрує низку спеціалізованих модулів, кожен з яких виконує критично важливі операції в процесі інтелектуальної обробки інформації, забезпечуючи її перетворення від неструктурованих вхідних даних до формалізованих та пояснюваних знань.

Обробка вхідних документів: Відповідає за ініціальний етап життєвого циклу інформації в системі. Він здійснює інжест неструктурованих документів, представлених у типових офісних форматах (наприклад, PDF, DOCX). Основною функцією є не лише ідентифікація формату, а і їхня трансформація у стандартизований текстовий формат, придатний для подальшого автоматизованого аналізу. Додатково цей модуль інтегрує засоби оптичного розпізнавання символів (OCR) для конвертації графічних представлень тексту, а також алгоритми попереднього очищення даних, спрямовані на усунення аберацій та нормалізацію текстового контенту для забезпечення високої якості вхідної інформації для подальших етапів обробки.

Семантичне анотування (Semantic Annotation): Центральний елемент інтелектуальної обробки, що реалізує парадигму семантичного збагачення даних. Шляхом інтеграції передових LLM та семантичних технологій, цей модуль здійснює не лише ідентифікацію номенклатурних сутностей у тексті, а й виявляє складні когнітивні та функціональні зв'язки між ними. Результатом є побудова деталізованої семантичної моделі документа, що включає контекстуалізовані відносини між об'єктами знань. Ця структура зберігається у структурованому сховищі знань, реалізованому на базі Semantic MediaWiki, формуючи онтологічно збагачений репозиторій для подальшого аналізу та запитів.

Перетворення природномовних запитів (Natural Language Query Transformation): Виконує функцію інтерфейсу між природномовним запитом користувача та формалізованою базою знань. Його призначення полягає у трансляції неформальних запитів, висловлених природною мовою, у структуровані запити мовою SPARQL або ASK – мовою для запитів до SMW. Цей процес вимагає глибокого семантичного розуміння запиту користувача та його відображення на онтологічну структуру сховища, забезпечуючи високу релевантність та точність отриманих результатів.

Генерації декларацій та аналітичних звітів (Declaration and Analytical Report Generation): Даний процес синтезує отримані знання у формі, зручній для кінцевого користувача. Він використовує не лише безпосередньо витягнуті факти, а й складні механізми логічного виведення (reasoning) для формування обґрунтованих та когерентних відповідей. Застосування підходу Retrieval-Augmented Generation (RAG) дозволяє інтегрувати можливості генерації тексту LLM з доступом до верифікованих знань з внутрішньої бази даних. Це забезпечує продукування не просто відповідей, а повноцінних аналітичних звітів та декларацій, які є прозорими, посилаються на джерела та відповідають встановленим стандартам звітності.

Управління знаннями (Knowledge Management): Є фундаментом для забезпечення життєвого циклу знань у системі. Він надає функціональність для створення, персистенції, модифікації та версифікації знань, представлених у вигляді онтологій та статей Semantic MediaWiki. Онтології забезпечують формалізоване представлення предметної області, тоді як статті Semantic MediaWiki слугують зручним інтерфейсом для взаємодії зі знаннями. Система версифікації критично важлива для відстеження еволюції знань у динамічних доменах, гарантуючи їхню актуальність та цілісність.

Експертна валідація (Expert Validation Module): Ця функція впроваджує парадигму Human-in-the-Loop, забезпечуючи високий рівень достовірності результатів. На відміну від автоматизованої обробки, система надає інтерфейс для ручної верифікації та коригування витягнутих сутностей та згенерованих декларацій.

Можливість гнучкої корекції та аналізу експертами дозволяє мінімізувати потенційні помилки, підвищити точність інформації та забезпечити відповідність високим галузевим стандартам, особливо у сферах, де помилки неприпустимі.

Експорт результатів (Results Expor): Завершальний етап функціонального ланцюга, що забезпечує інтеграцію системи із зовнішніми середовищами та бізнес-процесами. Даний модуль відповідає за форматування та експорт вихідної звітної документації у стандартизованих, загальноприйнятих форматах (наприклад, PDF, DOCX). Це дозволяє безперешкодно інтегрувати результати роботи "ЛІНЗИ" в існуючий документообіг, що забезпечує можливість передачі сформованих документів до відповідних сертифікаційних або регуляторних органів.

Основні модулі системи

Сервіс-орієнтована архітектура системи "ЛІНЗА" забезпечує її модульність та гнучкість. Вона складається з чотирьох основних функціональних груп сервісів: модулі взаємодії з користувачем, сервіси обробки документів та управління LLM, сервіси управління знаннями, сервіси валідації та експорту, процеси взаємодії яких узагальнено представлено на рис. 4.

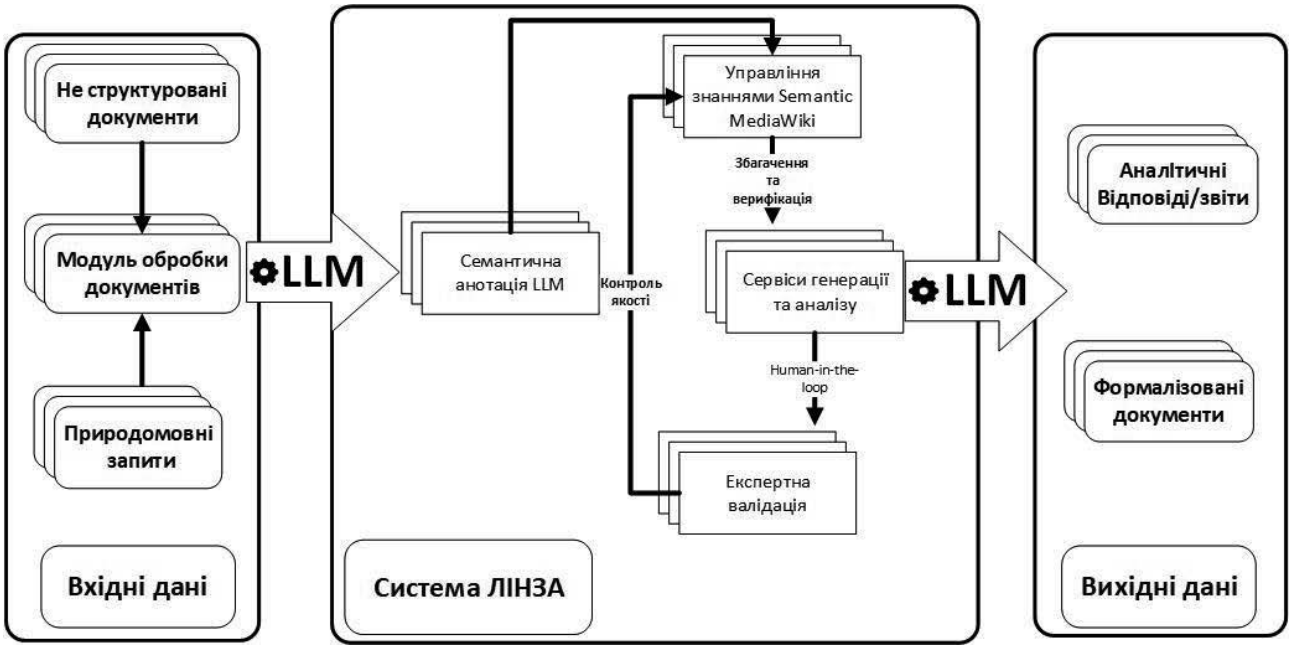


Рис. 4. Узагальнена схема функціонування системи "ЛІНЗА"

Модулі взаємодії з користувачем

- **UI Service** — це графічний інтерфейс користувача, через який відбувається завантаження документів, подання запитів, перегляд результатів та декларацій.
- **API Gateway Service** — єдина точка входу до системи, що виконує функції маршрутизації запитів, автентифікації та авторизації.

Сервіси обробки документів та управління LLM

- **Document Parsing Service** — відповідає за трансформацію вхідних файлів у структурований текст із попереднім очищенням і OCR.
- **LLM Orchestration Service** — головний координаційний модуль, що розподіляє завдання між агентами та LLM, виконує reasoning і генерацію за ReAct-підходом.
- **NLQ-to-SPARQL** — трансформує природномовні запити у формальні запити до бази знань.

- **Response Generation** — виконує генерацію відповідей на основі результатів SPARQL-запитів.

Сервіси управління знаннями

- **Knowledge Base Service (Semantic MediaWiki Core)** — реалізує сховище структурованих знань та підтримує SPARQL-запити.
- **Persistence Service** — забезпечує довготривале зберігання вхідних документів, онтологій, логів та інших службових даних.
- **Vector Database Service** — векторне сховище embedding-представлень знань для семантичного пошуку в задачах типу RAG.

Сервіси валідації та експорту

- **Validation Service** — надає інтерфейс для експертної перевірки витягнутих сутностей та декларацій, дозволяє ручну корекцію та затвердження.
- **Export Service** — відповідає за експорт результатів у відповідні стандартизовані формати (PDF, DOCX), з можливістю подальшої передачі сертифікаційним органам.

Вхідні та вихідні дані системи

Вхідні дані:

- **Неструктуровані текстові документи**: Завантажуються користувачами у форматах PDF, DOCX.
- **Природномовні запити**: Формулюються користувачами для отримання пояснень чи витягів з бази знань.
- **Редагування експертів**: Валідовані уточнення, зауваження, вручну додані знання.

Вихідні дані:

- **Аналітичні відповіді**: Сформовані системою результати запитів, представлені у зрозумілій для користувача формі.
- **Семантичні сутності**: Витягнуті та семантично пов'язані поняття, збережені у базі знань.
- **Декларації та звіти**: Автоматично згенеровані документи у форматах PDF, DOCX, придатні для подальшого використання.
- **Embedding-представлення**: Векторні форми знань для швидкого пошуку релевантної інформації.

Висновки

У роботі досліджено потенціал синергетичної інтеграції великих мовних моделей та семантичних технологій для автоматизованого створення й обробки складних природномовних документів. Розглядається концепція побудови гібридної інформаційної системи лінгвістичної обробки документів «ЛІНЗА», яка поєднує великі мовні моделі (LLM), семантичні технології та мультиагентні архітектурні рішення. Основна мета дослідження полягає в розробці автоматизованої платформи для обробки складних природномовних документів, що потребують високого ступеня точності, структурного узгодження та пояснювання.

Запропонована система має багаторівневу архітектуру, яка містить модулі попередньої обробки документів, семантичного анотування, генерації документів і аналітичних звітів, управління знаннями та експертної валідації. Особливу увагу приділено використанню Retrieval-Augmented Generation як механізму інтеграції генеративних можливостей LLM із формалізованими базами знань на основі Semantic MediaWiki.

На етапі архітектурного проєктування авторами запропоновано багаторівневу модель інформаційної системи «ЛІНЗА», що поєднує агентні компоненти з використанням LLM для глибокого семантичного аналізу, набір сервісів для динамічної обробки даних і платформу Semantic MediaWiki для структурованого зберігання та верифікації контенту.

Запропонована архітектура демонструє високий потенціал ефективного застосування в критичних доменах, де особливо важливими є достовірність, прозорість і безпека інформаційної обробки. Залучення формалізованих семантичних шаблонів, механізмів перевірки та підходу

human-in-the-loop забезпечує підвищення точності, надійності та контрольованості результатів.

Застосування системи «ЛІНЗА» охоплює сфери публічного управління, юриспруденції, сертифікації, стандартизації та спрямоване на забезпечення прозорої, контрольованої і формалізованої обробки документів, що відкриває перспективи масштабованої автоматизації складних інформаційних процесів з явним використанням знань предметної - області.

Подальший розвиток системи доцільно спрямувати на вдосконалення механізмів інтеграції з джерелами знань, покращення інтерпретованості результатів і розширення підтримки динамічних сценаріїв використання, а також ширшу інтеграцію зі стандартами проєкту Semantic Web [16], що стосуються подання сервісів, програмних агентів та онтологій.

Література

1. Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Mian, A. A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology*. 2023. URL: https://dl.acm.org/doi/pdf/10.1145/3744746.

2. 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. P. 1–24. DOI: https://doi.org/10.1080/09540091.2023.2168253.

3. Musumeci, E., Brienza, M., Suriani, V., Nardi, D., Bloisi, D. D. LLM-based multi-agent generation of semi-structured documents from semantic templates in the public administration domain. In: *Proceedings of the International Conference on Human-Computer Interaction (HCII 2024)*. Cham: Springer, 2024. P. 98–117.

4. Eichhorn, T. CLAIR: Generating on-demand low-code application documentation through knowledge graph and LLM-based multi-agent system integration. Master's thesis. University of Twente, 2025.

5. Плескач В. Л., Рогушина Ю. В. Агентні технології: Монографія. Київ : Київ. нац. торг.–екон. ун–т, 2005.

6. Рогушина Ю. В., Гладун А. Я., Осадчий В. В., Прийма С. М. Онтологічний аналіз у Web: Монографія. Мелітополь : МДПУ ім. Богдана Хмельницького, 2015. 407 с. URL: http://www.dut.edu.ua/uploads/l_2148_67675988.pdf. ISBN 978-617-7346-27-1.

7. Vrandečić, D., Krötzsch, M. Semantic MediaWiki. In: *Semantic Knowledge Management: Integrating Ontology Management, Knowledge Discovery, and Human Language Technologies*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. P. 171–179.

8. Chen, J., Lu, X., Du, Y., Rejtig, M., Bagley, R., Horn, M., Wilensky, U. Learning agent-based modeling with LLM companions: Experiences of novices and experts using ChatGPT & NetLogo chat. In: *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 2024. P. 1–18.

9. Yang, D., Simoulin, A., Qian, X., Liu, X., Cao, Y., Teng, Z., Yang, G. DocAgent: A multi-agent system for automated code documentation generation. *arXiv preprint arXiv:2504.08725*. 2025.

10. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Kiela, D. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*. 2020. Vol. 33. P. 9459–9474.

11. Machado, M., Rodrigues, J. M., Lima, G., Fiorini, S. R., da Silva, V. T. LLM Store: Leveraging large language models as sources of Wikidata-structured knowledge. In: *Proceedings of the International Semantic Web Conference (ISWC 2024)*. Cham: Springer, 2024 (to appear).

12. Mihindukulasooriya, N., Tiwari, S., Dobriy, D., Nielsen, F. Å., Chhetri, T. R., Polleres, A. Scholarly Wikidata: Population and exploration of conference data in Wikidata using LLMs. In: *Proceedings of the International Conference on Knowledge Engineering and Knowledge Management (EKAW 2024)*. Cham: Springer, 2024. P. 243–259.

13. Каверинський В. В., Літвін А. А., Палагін О. В. Зворотний синтез природномовних висловлювань на основі їх онтологічного представлення з використанням великої мовної моделі. *Проблеми програмування*. 2024. № 2-3. С. 359. DOI: https://doi.org/10.15407/pp2024.02-03.359.

14. Рогушина Ю. В., Гладун А. Я., Аніщенко О. В., Прийма С. М. Семантичні технології як інструмент інформаційного забезпечення професіоналізації андрагогів. *Проблеми програмування*. 2024. № 2-3. С. 441. DOI: https://doi.org/10.15407/pp2024.02-03.441.

15. Rotsos, C., King, D., Farshad, A., Bird, J., Fawcett, L., Georgalas, N., Hutchison, D. Network service orchestration standardization: A technology survey. *Computer Standards & Interfaces*. 2017. Vol. 54. P. 203–215.

16. Patel, A., Jain, S. Present and future of semantic web technologies: a research statement. *International Journal of Computers and Applications*. 2021. Vol. 43, Iss. 5. P. 413–422.

Одержано: 19.07.2025
Внутрішня рецензія отримана:26.07.2025
Зовнішня рецензія отримана: 28.07.2025

Про авторів:

Сініцин Ігор Петрович,
доктор технічних наук, професор,
член-кореспондент НАН України,
https://orcid.org/0000-0002-4120-0784,
ips@nas.gov.ua

Рогушина Юлія Віталіївна,
кандидат фіз.-мат.наук, с.н.с., доцент,
http://orcid.org/0000-0001-7958-2557,
ladamandraka2010@gmail.com

Юрченко Костянтин Юрійович,
аспірант, м.н.с.
https://orcid.org/0000-0003-3150-0027,
urchikak8@gmail.com

Місце роботи авторів:

Інститут програмних систем
Національної академії наук України,
03187, м. Київ-187,
просп. Академіка Глушкова, 40,
корпус 5.

УДК 004.89:004.51 https://doi.org/10.15407/pp2025.02.077

B. O. Kuzikov, O. A. Shovkoplias, P. O. Tytov, S. R. Shovkoplias

APPLICATION OF SMALL LANGUAGE MODELS FOR SEMANTIC ANALYSIS OF WEB INTERFACE ACCESSIBILITY

Web accessibility remains a critical aspect of ensuring equal opportunities for internet resource usage, especially for people with disabilities. The Web Content Accessibility Guidelines 2.5.3 criterion “Label in Name” requires that the accessible name of an interface component include text that is visually presented. Existing automated verification methods for this criterion are predominantly based on primitive string comparison, which does not account for semantic context. Objective: investigate the possibilities of using small language models with up to 1 billion parameters for automated semantic analysis of compliance with Web Content Accessibility Guidelines 2.5.3, as an alternative to resource-intensive large language models and limited algorithmic methods. Methodology: the research involved creating synthetic datasets (7,200 English-language and 5,615 Ukrainian-language samples) and using real-world datasets (Top500 – 380 samples, UaUniv – 319). Sentence Bidirectional Encoder Representations from Transformers models were tested for computing semantic similarity, and fine-tuning of the google/electra-base-discriminator model was performed for 3-class classification of semantic relationships (“similar”, “unrelated”, “opposite”). Results: the trained model of 437 MB demonstrated high accuracy on synthetic data (0.96) and sufficient accuracy on real datasets (Top500: 0.77, UaUniv: 0.73). The model effectively identifies all three classes of semantic relationships with an accuracy of 95.1 % for “opposite”, 92.7 % for “unrelated”, and 97.4 % for “similar” texts in the validation sample. Conclusions: the research confirmed the feasibility of using small language models for automated verification of semantic compliance according to Web Content Accessibility Guidelines 2.5.3. The proposed approach provides acceptable classification accuracy with significantly lower computational costs compared to large language models, allowing for the integration of semantic analysis into standard development and testing processes. Despite certain limitations, the developed solution can significantly improve web accessibility testing.

Keywords: Small Language Models, Semantic Analysis, Text Classification, Web Accessibility, Web Content Accessibility Guidelines, Model Fine-Tuning, Natural Language Processing

Б. О. Кузіков, О. А. Шовкопляс, П. О. Титов, С. Р. Шовкопляс

ЗАСТОСУВАННЯ МАЛИХ МОВНИХ МОДЕЛЕЙ ДЛЯ СЕМАНТИЧНОГО АНАЛІЗУ ДОСТУПНОСТІ ВЕБІНТЕРФЕЙСІВ

Вебдоступність залишається важливим аспектом для забезпечення рівних можливостей користування інтернет-ресурсами, особливо для людей з інвалідністю. Критерій Настанов з доступності вебвмісту 2.5.3 «Мітка в імені» вимагає, щоб доступне ім’я компонента інтерфейсу включало текст, представлений візуально. Існуючі методи автоматизованої перевірки цього критерію базуються переважно на примітивному порівнянні рядків, не враховуючи семантичний контекст. Мета. Дослідити можливості застосування малих мовних моделей із кількістю параметрів до 1 мільярда для автоматизованого семантичного аналізу відповідності критерію Настанов із доступністю вебвмісту 2.5.3 як альтернативи ресурсомістким великим мовним моделям і обмеженим алгоритмічним методам. Методологія. Дослідження передбачало створення синтетичних (англомовних – 7200, україномовних – 5615 прикладів) та використання реальних наборів даних (380 прикладів із 500 найвідвідуваніших вебсайтів, 319 прикладів із вебсайтів українських університетів). Здійснено тестування моделей двоспрямованих кодувальних представлень речень із трансформерів для обчислення семантичної схожості та використано тонке налаштування базової дискримінаторної моделі ELECTRA від Google: для 3-класової класифікації семантичних відношень («схожі», «непов’язані», «протилежні»). Результати. Навчена модель розміром 437 МБ продемонструвала високу точність на синтетичних даних (0,96) та достатню точність на реальних наборах даних (0,77 для найвідвідуваніших вебсайтів та 0,73 для вебсайтів українських університетів). Модель здатна ефективно ідентифікувати всі три класи семантичних відношень із точністю 95,1 % для «протилежних», 92,7% для «непов’язаних» та 97,4% для «схожих» текстів на валідаційній вибірці. Висновки. Дослідження підтвердило доцільність застосування малих мовних моделей для автоматизованої перевірки семантичної відповідності згідно з критерієм Настанов із доступністю вебвмісту 2.5.3. Запропонований підхід забезпечує при-

йнятну точність класифікації при значно менших обчислювальних витратах порівняно з великими мовними моделями, що дозволяє інтегрувати семантичний аналіз у стандартні процеси розроблення та тестування. Незважаючи на певні обмеження, розроблене рішення може істотно покращити процес тестування вебдоступності.
Ключові слова: малі мовні моделі, семантичний аналіз, класифікація тексту, вебдоступність, Настанови з доступності вебвмісту, тонке налаштування моделей, обробка природної мови

Introduction

Motivation. Web accessibility is critically important for ensuring equal access to information and services for all users. The Web Content Accessibility Guidelines (WCAG) define relevant standards. One important criterion is WCAG 2.5.3 “Label in Name”, which specifies that the accessible name of an interface component must include text that is visually presented. This allows users with disabilities to rely on visible labels as a means of interaction: individuals using voice control can activate elements by speaking their visible names, and users of text-to-speech technologies gain a better experience due to consistency between seen and heard text [1]. Developers have ethical and legal obligations to comply with accessibility standards. Conducting accessibility testing is fundamental to improving application usability for people with disabilities and generally enhances usability for all users [2].

Automated accessibility testing is recognized as an effective tool for quickly identifying a significant portion of problems. It allows for systematic evaluation of the user interface and code for compliance with numerous rules and recommendations [2]. However, despite its advantages, automated testing is not an exhaustive solution and has significant limitations [3]. In particular, automated tools cannot fully evaluate the context of use, complex interactions, and subjective aspects of user experience, which are critically important for users with different needs [4, 5]. Passing automated tests does not guarantee full application accessibility, and results may contain false positives that require human verification. Thus, a comprehensive approach to ensuring accessibility requires a combination of automated testing with manual testing and involvement of users with disabilities.

The task of verifying WCAG 2.5.3 criterion essentially comes down to fuzzy string comparison, with or without consideration of semantics and structure, depending on the quality of the tool. Existing automated tools may take into account Best Practice recommendations – “The label should begin with visible text”. Classical, deterministic string comparison algorithms can be divided into several categories: fuzzy text comparison (Levenshtein distance, longest common subsequence), phonetic algorithms (Soundex, Metaphone, New York State Identification Intelligence System), and token-based methods (Jaccard coefficient, cosine similarity, BM25). It is important to note that from those listed, only the last category can account for semantic proximity of texts, but its capabilities are limited without using.

Recently, there has been significant interest in artificial intelligence capabilities. Despite impressive results, tools such as ChatGPT have limitations in specific tasks, particularly in accessibility testing, as they use training data that does not always cover the specifics and requirements of modern accessibility standards [6]. In the context of artificial intelligence development, particularly in the field of natural language processing, language models are often classified by their size and computational requirements. Large language models (LLMs), such as GPT-4 or GLaM [7], contain hundreds of billions or even trillions of parameters and require significant computational resources for training and execution. Due to their ability to consider semantic relationships and context, LLMs can understand and evaluate headings and labels with a high degree of accuracy. However, deploying LLMs for real-time accessibility testing faces significant challenges, including high computational requirements, high latency, and potential instability of provider APIs. This creates a need for efficient, reliable, and context-oriented solutions that can operate with limited computational resources.

Small language models (SLMs) are significantly more compact – they typically contain from a few hundred million to a few billion parameters, ensuring their availability and efficient deployment on standard equipment and facilitating integration into everyday tools and development processes [8]. Among SLMs, micro- and nano-language models stand out, with parameter counts typically not exceeding one billion. Despite their smaller size, these models can achieve performance comparable to larger models in specific tasks after appropriate fine-tuning [8, 9]. For example, the Phi-3-mini model with only 3.8 billion parameters demonstrated performance corresponding to models twice its size in various natural language understanding tasks [10].

The aim of the research is to analyze the capabilities of SLMs (up to 1 billion parameters) for automated verification of compliance with WCAG 2.5.3 criterion. We investigate the use of Sentence Transformers (SBERT) models, ready-made classification tasks from the Hugging Face platform, and fine-tuning of a selected SLM. The hypothesis is that properly configured SLMs can provide accurate semantic analysis of relationships between labels and names through 3-class classification ("similar", "unrelated", "opposite"), while maintaining the performance characteristics necessary for integration into existing development processes. This could fill the gap between simple string matching and resource-intensive LLM-based solutions.

The main research objectives:

– Develop and adapt a methodology for applying SLMs for automated verification of compliance with WCAG 2.5.3 "Label in Name" criterion in web interfaces.

– Evaluate the effectiveness of SBERT models for semantic analysis of relationships between visible text labels and accessible names in the context of WCAG 2.5.3 criterion.

– Conduct fine-tuning of a selected SLM for 3-class classification of relationships between visible text labels and accessible names ("similar", "unrelated", "opposite").

– Validate the developed solution by evaluating its performance on synthetic datasets, as well as on real data from the Top 500 most visited websites (Top500) and Ukrainian university websites (UaUniv).

Methodology

Dataset Preparation. The research involved both synthetic and real datasets to evaluate and train SLMs. Specifically, synthetic datasets (English – 7,200 samples, Ukrainian – 5,615 samples) were created using leading LLMs (Anthropic Claude, OpenAI ChatGPT, Google Gemini, Grok 3). A diverse range of models was used to increase input data variety and minimize potential biases. To ensure meaningful control over generated samples, a taxonomy of semantic changes was developed beforehand, describing typical text modifications in web content with Accessible Rich Internet Applications (ARIA) attributes. Its application allowed for systematizing change types and ensuring the relevance of synthetic samples.

The taxonomy classifies differences between visible text and its ARIA description, considering both the nature of changes and their potential impact on web resource accessibility and security. Categories include context expansion ("Submit" → "Submit registration form"), action object changes ("Submit payment" → "Submit order"), action type changes ("Save" → "Save and delete"), negation ("Submit" → "Do not submit"), technical modifications ("Submit form" → "SUBMIT FORM"), etc.

In addition to synthetic datasets, the study also utilized real datasets that reflect practical samples of semantic discrepancies in web content:

– Top500 – contains 380 samples of differences between visible text and its representation for assistive technologies. Data was collected from pages of the 500 most popular websites according to the Moz ranking [11], ensuring representation of contemporary publicly accessible internet content.

– UaUniv – includes 319 samples collected from the main pages of official websites of Ukrainian higher education institutions. Data collection was conducted as part of an accessibility study in January 2024 [12], allowing assessment of text information presentation specifics in the educational segment of the Ukrainian internet space.

For training classification models based on SLMs, input data was pre-annotated

using the LLM-as-Judge approach [13]. In this approach, large language models serve as expert evaluators, enabling the scaling of the annotation process without requiring extensive human resources. Queries to LLMs consisted of instructions (system prompt) and user data – pairs of visible text and ARIA labels. The model evaluated semantic similarity between these elements on a scale from –1.0 to 1.0, where –1.0 indicates complete opposition or content contradiction, 0 means no connection, and 1.0 represents complete semantic correspondence. Intermediate values reflected partial correspondence, including cases of context expansion, changes in object or action type. To ensure annotation reliability, consistency checks were performed on ratings generated by different LLMs. The numerical ratings obtained from LLMs in the range [–1.0, 1.0] were quantized into three semantic correspondence categories:

1. "Similar" (same/similar) – texts have identical or very close content (LLM ratings within [0.65, 1.0]).

2. "Not related" (not related) – texts have no semantic connection (ratings near zero).

3. "Opposite" (opposite) – texts have opposite or contradictory meanings (ratings within [–1.0, –0.15]).

Results obtained in previous research stages allowed us to create a high-quality annotated dataset that can be used for model training and evaluation. This is an important resource, especially considering the task complexity and the need for high-quality annotations for effective machine learning in the field of semantic text analysis.

As part of the research, more economical models were tested for their suitability for semantic text comparison tasks. The following models were tested: mistral/ministral-8b, qwen/qwen2.5-coder-7b-instruct, meta-llama/llama-3.1-8b-instruct, amazon/nova-micro-v1, liquid/lfm-3b, openai/gpt-4.1-nano, google/gemini-2.5-pro-exp-03-25, liquid/lfm-7b. However, none of these models demonstrated sufficient effectiveness for accurate analysis of semantic relationships between visible text labels and accessible names. This indicates the need to use more powerful models or additional fine-tuning to achieve acceptable results in this domain.

Using SBERT for Semantic Similarity. One promising approach to solving the semantic text comparison problem is using SBERT [14] – a specialized Python module for working with modern vector representation models and rerankers. This framework provides access to creating, using, and training state-of-the-art models for computing text embeddings. Sentence Transformers can be used both for computing vector representations of texts using Sentence Transformer models and for calculating text similarity metrics using Cross-Encoder models. This opens a wide range of applications, including semantic search, determining semantic textual similarity, and paraphrase detection.

Over 10,000 pre-trained Sentence Transformer models are available on the Hugging Face platform for immediate application, including many modern models from the Massive Text Embeddings Benchmark (MTEB) [15] ranking. Additionally, the framework makes it easy to train or fine-tune custom embedding models or rerankers, enabling the creation of specialized models for specific use cases. Particularly promising is the application of this toolkit for Semantic Textual Similarity (STS) tasks. In this approach, vector representations are created for all analyzed texts, after which similarity metrics between them are calculated. Text pairs with the highest similarity score are considered semantically closest.

Testing on a synthetic dataset showed that baseline SBERT models rank text similarity well. However, standard distance metrics (Euclidean, cosine similarity) do not effectively distinguish texts with opposite content, often classifying them in the same category as unrelated texts. This is a significant limitation for our task. Figure 1 shows classification accuracy by category for baseline SBERT models, illustrating this problem.

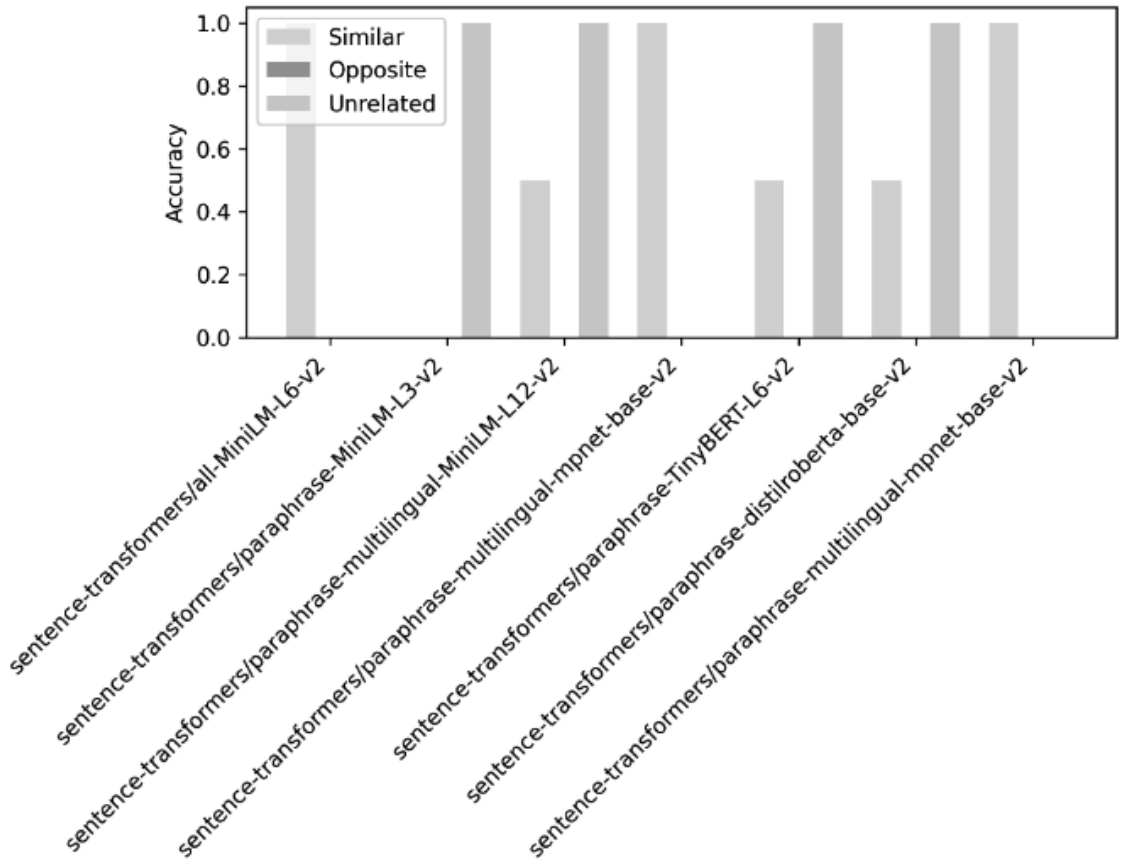


Fig. 1. Per-Category Classification Accuracy for basic SBERT

Hugging Face Tasks and Model Selection for Fine-Tuning. To find a compromise between computational efficiency and result quality, we turned to the Hugging Face ecosystem, which provides ready-made pipelines for natural language processing tasks. The key advantage of this approach is that most pre-trained models are relatively small and can be run locally, significantly reducing their usage cost compared to APIs of large models.

In our research, we considered several types of tasks available in Hugging Face:

– zero-shot-classification – a method that allows classifying texts by categories without seeing examples of these categories during training;

– text-classification – the traditional approach to assigning text to predefined categories;

– fill-mask – a task where the model fills in missing words in text, which can be used to evaluate semantic proximity;

– question-answering – the model answers questions based on context, which can potentially be adapted for text comparison;

– text-generation – creating new text that can be used for paraphrasing and subsequent comparison.

For these tasks, we tested a wide range of models of various architectures and sizes: “google/flan-t5-large”, “google/electra-large-discriminator”, “facebook/bart-large-mnli”, “roberta-large-mnli”, “l-yohai/bigbird-roberta-base-mnli”, “cross-encoder/nli-distilroberta-base”, “distilbert-base-uncased-finetuned-sst-2-english”, “bert-base-uncased”, “albert-base-v2”, “roberta-base”, “distilbert-base-cased-distilled-squad”, “deepset/roberta-base-squad2”, “google/electra-large-discriminator”, “EleutherAI/gpt-neo-125M”, “gpt2”, “distilgpt2” and “facebook/opt-350m”. Testing was conducted taking into account the architectural features of each model and the specifics of the corresponding pipelines. Figure 2 presents a comparison of different models by size (in megabytes) and achieved accuracy on the synthetic dataset.

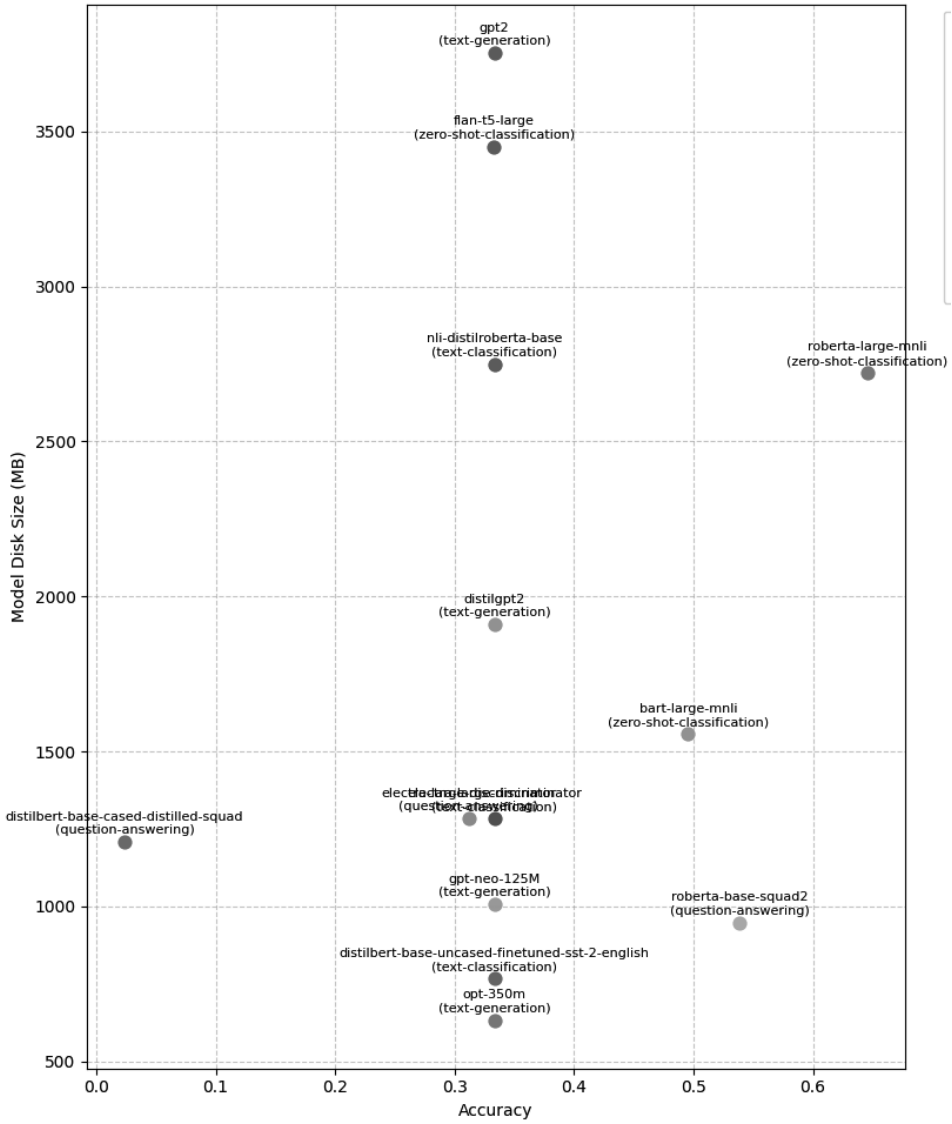


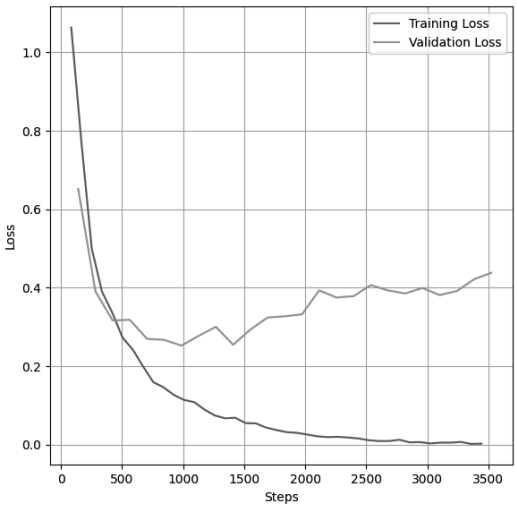
Fig. 2. Model Performance
(Accuracy vs. Size)

Analysis showed that none of the tested combinations of ready-made models and pipelines fully met the task requirements: models either demonstrated insufficient accuracy or were too large for efficient local use. Therefore, a decision was made to fine-tune a model. For this purpose, google/electra-base-discriminator [16] was chosen in combination with the text-classification pipeline as the most promising in terms of balance between size, potential accuracy, and computational requirements.

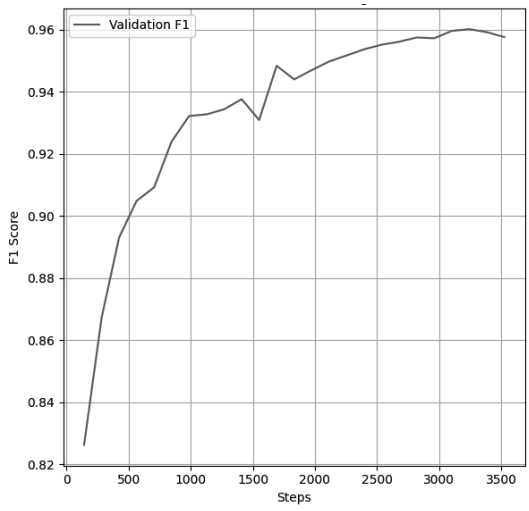
Fine-tuning of the google/electra-base-discriminator model was conducted on a mixed dataset that included English and Ukrainian synthetic samples. The dataset was augmented by swapping texts in pairs, considering the symmetry of the similarity function. Thus, the training set contained 17,941 samples, and the validation set – 7,689 samples.

Results

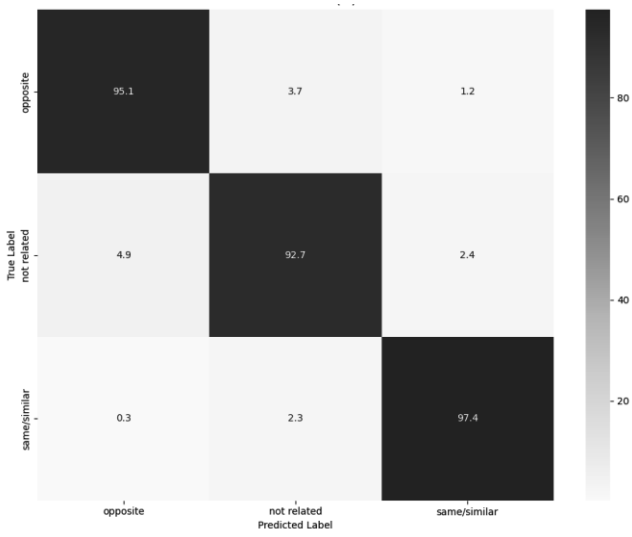
As a result of fine-tuning the google/electra-base-discriminator model, we obtained a model with a size of 437 MB. On the validation set (from synthetic data), the model achieved the following metrics: F1-score = 0.96, Recall = 0.96. The results of model fine-tuning are shown in Figure 3.



a



b



c

Fig. 3. Loss (a), F1-Score (b), and Confusion Matrix (c)

The Loss graph demonstrates a stable decrease in both training and validation losses to a level of ≈0.2–0.3 after 2000–2500 steps, indicating effective optimization. The Validation F1-Score increased to 0.96, reflecting high classification accuracy. The confusion matrix confirms accuracy for the classes “opposite” (95.1 %), “not related” (92.7 %), and “same/similar” (97.4 %).

Testing of the fine-tuned model on real data from the Top500 and UaUniv datasets showed acceptable accuracy. The results are presented in Table 1.

Table 1.

Results of testing the fine-tuned model on real data

	Top500	**UaUniv**
Accuracy	0.76	0.72
Precision	0.79	0.74
Recall	0.76	0.72
F1-score	0.77	0.73

Discussion

The article addresses a fundamental contradiction in the comparison of visible text and text for assistive technologies, where attention is often focused solely on formal compliance, such as “The aria-label text begins with the text of the visible label”. This compliance is easily verifiable algorithmically but fails to account for the semantic content of the texts. Basic methods, such as fuzzy string comparison (e.g., Levenshtein distance), are incapable of considering the semantic content of texts. This makes them unsuitable for the task of evaluating semantic similarity, as they may ignore significant differences or incorrectly classify texts as similar when their content differs. As evidence, in the Top500 dataset of 382 samples, 287 (75 %) were identified by LLM as semantically similar but formally violate the WCAG 2.5.3 criterion for algorithmic methods. Similarly, in the UaUniv dataset, 149 out of 320 samples (46 %) were classified by LLM as similar but did not meet the criterion for basic methods. These results demonstrate that basic methods cannot provide adequate semantic analysis, therefore comparison with them was not conducted in this study.

The research results demonstrate that micro- and nano-language models, particularly

the fine-tuned google/electra-base-discriminator model, can be effectively used for automated verification of compliance with the WCAG 2.5.3 criterion. The transition from a detailed continuous scale of semantic similarity assessment from –1.0 to 1.0, which was used for data annotation by large language models (where, recall, 1.0 meant complete semantic correspondence, and –1.0 meant opposition), to a more generalized 3-class classification (“similar”, “unrelated”, “opposite”) proved to be a successful approach for training SML. This allowed for high accuracy on synthetic data (F1 = 0.96).

The initial investigation of SBERT models confirmed their ability to rank similarity but also revealed limitations in clearly distinguishing semantically opposite texts using standard distance metrics. This highlighted the need for more specialized approaches, such as fine-tuning for a specific classification task.

The performance of the fine-tuned model on real datasets Top500 (F1 = 0.77) and UaUniv (F1 = 0.73) is somewhat lower than on synthetic data. This is expected, as real data often contains greater diversity and complexity of samples than synthetic data. However, the achieved indicators are still sufficiently high for practical application, especially considering the significantly lower computational resources required for SLM compared to LLM. Analysis of real websites showed that the vast majority of detected errors were related not so much to subtle semantic nuances between visible text and accessible name, but to fundamentally incorrect markup. This often made the use of assistive technologies extremely difficult or even impossible, rather than merely causing confusion due to semantic discrepancies. Among unexpected patterns, it is worth highlighting the contextual sensitivity and certain language independence of SLM, which are positive aspects.

Research Limitations. The analysis of real data was limited to the Top500 and UaUniv datasets, which, although representative, do not cover the entire spectrum of websites. The effectiveness of SLM largely depends on the quality of data for fine-tuning and the fine-tuning process itself.

Despite the achieved results, it is important to remember that automated accessibility testing, even using advanced models similar to those proposed, is not a "silver bullet." It effectively identifies technical compliance with standards but cannot fully replace human verification for evaluating context, complex interactions, and overall user experience [3]. Thus, the presented research contributes to this important field by offering a solution that simplifies accessibility testing and reduces barriers to its implementation, but the best results are achieved when combining automated methods with manual expertise.

Practical Significance. The obtained results confirm that SLMs can serve as a foundation for developing new, more accessible, and efficient tools for automated web accessibility verification. This allows for the integration of semantic analysis into development processes without excessive resource expenditure.

The application of specialized AI tools developed by accessibility experts can significantly improve the testing process and expand its coverage [6]. Thus, the presented study of micro- and nano-language models contributes to this important field by offering a solution that simplifies accessibility testing and reduces barriers to its implementation.

Conclusion

The research demonstrated the effectiveness of using micro- and nano-language models for automating the verification of semantic compliance according to the WCAG 2.5.3 criterion “Label in Name”.

Key findings:

1. SBERT models are useful for obtaining vector representations of texts and initial similarity ranking; however, standard metrics do not reliably distinguish semantically opposite texts.

2. Fine-tuning a relatively small model (google/electra-base-discriminator, 110M parameters) for the task of 3-class classification (“similar”, “unrelated”, “opposite”) allowed for high accuracy (F1 = 0.96) on synthetic data and sufficient accuracy (F1 up to 0.77) on real data (Top500, UaUniv).

3. SLMs are significantly more compact and less resource-intensive compared to LLMs, making them suitable for local deploy-

ment and integration into various development tools.

4. The developed approach offers a practical solution for improving automated accessibility testing, complementing existing tools with semantic analysis capabilities.

Despite the achieved results, it is important to remember the limitations of SLMs and the necessity of human verification in complex cases. Further research may be directed toward expanding training datasets, investigating other SLM architectures and fine-tuning methods, as well as integrating the developed models into comprehensive accessibility testing systems. This will contribute to creating a more accessible web environment for all users.

To ensure the reproducibility of our research, we publish our artifacts on Kaggle [17].

References

1. Web Content Accessibility Guidelines (WCAG) 2.1 [Electronic resource]. 2025. URL: https://www.w3.org/TR/WCAG21/ (accessed: 01.06.2025).
2. Suarez C. Comprehensive Guide to Automated Accessibility Testing [Electronic resource]. 2024. URL: https://kobiton.com/blog/comprehensive-guide-to-automated-accessibility-testing/ (accessed: 01.06.2025).
3. Prasad M. DigitalA11Y. Automated Accessibility Testing Is Not a Sil-ver Bullet [Electronic resource]. 2025. URL: https://www.digitala11y.com/automated-accessibility-testing-is-not-a-silver-bullet/ (accessed: 01.06.2025).
4. Wieland R. Limitations of an Automated-Only Web Accessibility Plan [Electronic resource]. 2024. URL: https://allyant.com/blog/limitations-of-an-automated-only-web-accessibility-plan/ (accessed: 01.06.2025).
5. Intelligence Community Design System. Limitations of automated testing [Electronic resource]. URL: https://design.sis.gov.uk/accessibility/testing/automated-testing-limitation (accessed: 01.06.2025).
6. Barrell N. Enhancing Accessibility with AI and ML [Electronic resource]. 2023. URL: https://www.deque.com/blog/enhancing-accessibility-with-ai-and-ml/ (accessed: 01.06.2025).
7. Du N. et al. GLaM: Efficient Scaling of Language Models with Mixture-of-Experts // Proc Mach Learn Res. ML Research Press, 2021. Vol. 162. P. 5547–5569.
8. Achary S. The Rise of Small Language Models: A New Era of AI Accessibility and Efficiency [Electronic resource]. 2024. URL: https://medium.com/small-language-models/the-rise-of-small-language-models-a-new-era-of-ai-accessibility-and-efficiency-752322d82656 (accessed: 01.06.2025).
9. Aralimatti R. et al. Fine-Tuning Small Language Models for Domain-Specific AI: An Edge AI Perspective. Preprints, 2025.
10. Abdin M. et al. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. 2024.
11. Moz. Top 500 Most Popular Websites [Electronic resource]. URL: https://moz.com/top500 (accessed: 01.06.2025).
12. Титов П.О., Шовкопляс О.А., Кузіков Б.О. Аналіз вебдоступності сайтів українських закладів вищої освіти // Системні дослідження та інформаційні технології. 2025. № 2.
13. Gu J. et al. A Survey on LLM-as-a-Judge. 2024. Vol. 1.
14. Reimers N., Gurevych I. Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation // EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference. Association for Computational Linguistics (ACL), 2020. P. 4512–4525.
15. Muennighoff N. et al. MTEB: Massive Text Embedding Benchmark // EACL 2023 - 17th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference. Association for Computational Linguistics (ACL), 2022. P. 2006–2029.
16. Clark K. et al. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators // 8th International Conference on Learning Representations, ICLR 2020. International Conference on Learning Representations, ICLR, 2020.
17. Tytov P. Semantic Language Models for WCAG [Electronic resource]. URL: https://www.kaggle.com/datasets/tytovpavel/semantic-language-models-for-wcag (accessed: 01.06.2025).

Одержано: 06.06.2025
Внутрішня рецензія отримана: 13.06.2025
Зовнішня рецензія отримана: 16.06.2025

Про авторів:

Кузіков Борис Олегович,
к.т.н., доцент
https://orcid.org/0000-0002-9511-5665
b.kuzikov@cs.sumdu.edu.ua

Шовкопляс Оксана Анатоліївна
к.ф.-м.н., доцент
https://orcid 0000-0002-4596-2524
o.shovkoplyas@mss.sumdu.edu.ua

Титов Павло Олегович
здобувач ступеня доктора філософії
https://orcid.org/0009-0003-6911-5463
stegaspasha@gmail.com

Шовкопляс Сергій Ростиславович
здобувач ступеня доктора філософії
https://orcid.org/0000-0003-1837-0213
s.shovkoplyas@student.sumdu.edu.ua

Місце роботи та навчання авторів:

Сумський державний університет,
кафедра комп'ютерних наук,
40007, м. Суми, вул. Харківська, 116,
тел. +38(0542)687776

УДК 004.62, 004.67 https://doi.org/10.15407/pp2025.02.087

О.В. Царинюк, А.М. Глибовець

СЕГМЕНТАЦІЯ ГЕОПРОСТОРОВИХ РАСТРІВ: АНАЛІЗ ЧАСОВИХ ХАРАКТЕРИСТИК АЛГОРИТМУ AREAONAREAOVERLAYER

У статті досліджується продуктивність алгоритму AreaOnAreaOverlayer, що використовується для сегментації геопросторових растрів за ознаками висоти в середовищі FME. Основну увагу приділено аналізу часових характеристик алгоритму під час обробки великих обсягів даних, зокрема, в задачах класифікації рослинного покриву. Описано експериментальне середовище, типові вхідні дані та вплив геометричних параметрів полігонів на час виконання операції. Отримані результати дають змогу оцінити межі застосування алгоритму та виявити залежності між структурою вхідних даних і обчислювальною складністю.

Ключові слова: просторовий аналіз, сегментація растрів, геоінформаційні системи, часові характеристики, продуктивність алгоритму

O.V. Tsaryniuk, A.M. Hlybovets

SEGMENTATION OF GEOSPATIAL RASTERS: ANALYSIS OF THE TEMPORAL CHARACTERISTICS OF THE AREAONAREAOVERLAYER ALGORITHM

This paper investigates the performance of the AreaOnAreaOverlayer algorithm used for segmenting geospatial rasters based on elevation features within the FME environment. The main focus is on analyzing the algorithm's temporal characteristics when processing large volumes of data, particularly in vegetation cover classification tasks. The study describes the experimental setup, typical input data, and the impact of polygon geometric parameters on execution time. The results provide insight into the algorithm's application limits and reveal dependencies between the structure of input data and computational complexity.

Keywords: spatial analysis, raster segmentation, geographic information systems, temporal characteristics, algorithm performance

.

Вступ

В останні десятиліття геопросторові дані все більше інтегруються у різні сфери людської діяльності. Якщо раніше геоінформаційні системи знаходили застосування лише в окремих галузях, на кшталт логістики чи військової справи, то зараз складно знайти галузь, яка не використовує карти чи геопросторові дані. Перед виробництвом геопросторових даних постають виклики, коли потрібно створювати більш точні геодані, у стисліші терміни та за меншу собівартість.

Основним джерелом геопросторових даних є засоби дистанційного зондування Землі (ДЗЗ). До цих засобів належать супутникові знімки, аерофотозйомка, лідарна зйомка, радіолокаційна зйомка та інші. Кожен із цих засобів має свої обмеження до застосування та специфіку обробки отриманих даних. Наприклад, супутниковий знімок несе в собі спектральні характеристики об'єктів, але на ньому відсутня інформація про висотну складову, а на 3D-моделі радіолокаційної зйомки відсутня інформація про типи об'єктів, розташовані на цій моделі. Лідарна зйомка дозволяє отримати інформацію про земну поверхню у тривимірному просторі, включно із типами об'єктів. Але вона має певні обмеження, зокрема, через необхідність використання літальних апаратів із відносно вузькою смугою охоплення (до 2 км за проліт) та значний обсяг вихідних даних, який коливається в межах 100–10 000 МБ/км².

Огляд літератури

Стрімкий розвиток технологій машинного навчання уможливив вдосконалення існуючих технологій отримання геоданих. У дешифруванні супутникових знімків широко застосовуються згорткові моделі, U-Net [1], ResNet [2], що дозволило значно заощадити ресурси на ручній векторизації супутникових зображень. Математичні моделі, описані у працях Liu, et al [3,4] дозволили створювати висотні моделі місцевості (DHM) з глобальним покриттям завдяки поєднанню відкритих даних лідарної зйомки та супутникових знімків високої роздільної здатності.

Сегментація зображень є однією з найскладніших задач обробки зображень. На сьогодні існує багато підходів і методів сегментації, зокрема, методи сегментації описані в роботах Hofmann & Tiede [5] та Mueller & Corcoran [6]. Більшість досліджень у сфері сегментації рослинності зосереджені на визначенні індивідуальних крон дерев. Цей напрям відіграє ключову роль у детальному вивченні лісових екосистем, що підтверджується роботами Douss, et al [7], Li, et al [8], Lindberg, et al [9], а також Jakubowski, et al [10]. Завдяки цим дослідженням було значно поглиблено наше розуміння характеристик окремих дерев, структури лісів та розподілу біомаси.

На відміну від детального аналізу окремих крон дерев, наше дослідження спрямоване на розробку методів узагальненої сегментації, що дозволяють виділяти великі масиви рослинності з подібними (або майже однаковими) висотними характеристиками. Такі підходи є ефективними для сегментації рослинного покриву на великих територіях, наприклад, на рівні цілих країн. Це особливо важливо для аналізу рослинності, необхідного для регіональних і національних екологічних оцінок, планування землекористування та реалізації масштабних природоохоронних - заходів.

Попередня робота

У попередній частині дослідження Tsaryniuk, et al [11] було розроблено три різні підходи до сегментації геопросторових растрів: методом згорткових фільтрів, методом рандомних точок та методом гексагонів. Кожен із методів використовує різні підходи до сегментації та має свої особливості.

Метод згорткових фільтрів має перевагу у вигляді можливості обробляти великі растри цілісно, без розділення на частини. Проте його використання ускладнюється у випадках, коли потрібно працювати з окремими тайлами: на стиках виникають відмінності в даних, що унеможливлює подальше об’єднання результатів в єдиний набір. Крім того, перетворення растрового зображення у векторну форму є складним під час обробки великих об’ємів даних, а також цей метод схильний до створення полігонів малої площі, які потребують додаткової фільтрації або узагальнення.

Метод рандомних точок дозволяє паралельно обробляти окремі полігони, на яких генеруються точки, що сприяє підвищенню продуктивності. Водночас він вимагає наявності додаткового векторного шару, який підлягає сегментації, тоді як інші методи працюють безпосередньо з растровими даними. У деяких випадках полігони, які використовуються для генерації точок, можуть бути надто великими, що ускладнює та уповільнює процес - обробки.

Метод гексагонів також забезпечує можливість паралельної обробки як на рівні растрових даних, так і на рівні векторної сітки гексагонів. Він не потребує додаткових векторних шарів, на противагу методу рандомних точок. Водночас точність результатів методу гексагонів дещо нижча порівняно з методом згорткових фільтрів.

Оцінка точності цих методів показала, що всі три методи мають достатньо високий показник точності. Для подальшої роботи було обрано метод гексагонів як найперспективніший в плані обробки великих масивів даних, оскільки цей метод має низку переваг у порівнянні з іншими методами: відносно невисока складність обчислень, можливість обробки даних частинами та подальше комбінування частин в єдиний набір даних.

Оскільки нашою основною задачею є побудова алгоритму, що зможе працювати з великими даними, в цій роботі ми зосередилися на визначенні характеристик даних, що впливають на час обробки та ресурси, необхідні для успішного завершення роботи алгоритму.

Опис методу сегментації гексагонами

У нашій роботі ми вирішуємо задачу поєднання геопросторових даних за типами об'єктів з висотною складовою у великих масштабах. Одним із прикладів застосування є сегментація рослинного покриву. Для реалізації запропонованого методу сегментації геопросторових растрів було обрано програмне середовище FME (Feature Manipulation Engine) [12].

FME — це програмна платформа, розроблена компанією Safe Software, призначена для інтеграції, трансформації та конвертації просторових і непросторових даних між різними форматами та структурами. FME підтримує сотні форматів ГІС, CAD, баз даних, растрових зображень, 3D-моделей, а також різноманітні хмарні сервіси. Особливістю FME є можливість створення робочих процесів (workflow) без необхідності програмування, завдяки використанню трансформерів — спеціалізованих інструментів для обробки даних.

Трансформер в FME — це окремий функціональний блок, який виконує певну операцію над даними у процесі трансформації. Наприклад, трансформери можуть змінювати геометрію, фільтрувати об'єкти, змінювати атрибути, об'єднувати дані, аналізувати просторові зв'язки тощо. Кожен трансформер має свою спеціалізацію та набір параметрів для налаштування.

Кастомний трансформер (Custom Transformer) — це користувацький набір з одного або кількох трансформерів, згрупованих в єдиний логічний блок. Він дозволяє створювати повторно використовувані компоненти з власними вхідними та вихідними портами, які можуть бути інтегровані в інші робочі процеси FME. Кастомні трансформери часто використовуються для спрощення складних робочих процесів, підвищення читабельності схем або стандартизації окремих частин трансформації.

Наш метод сегментації гексагонами (Рис. 1) передбачає створення послідовної сітки шестикутників із заданою стороною SIDE_LENGTH, поєднання цієї сітки з матрицею DHM (Digital Height Model) та векторним шаром рослинного покриву. Для генерації гексагонів було використано кастомний трансформер HexagonSampler (Рис. 2), розроблений та опублікований Gauthier [13]. Він створює сітку шестикутників по заданій рамці (у нашому випадку в ролі рамки виступає описуючий прямокутник вхідного набору рослинності). Створення сітки шестикутників складається з наступних етапів:

- ExpressionEvaluator та ExpressionEvaluator_2 обчислюють атрибути _hoffset та _voffset на базі заданого користувачем параметра довжини сторони шестикутника — SIDE_LENGTH:

$$hoffset = SIDE\_LENGTH * 3$$

$$voffset = \cos(30°) SIDE\_LENGTH * 2$$

- 2DgridAccumulator генерує сітку точок з інтервалом _hoffset та _voffset;
- Offsetter, підключений паралельною гілкою, створює копію сітки точок і зміщує її відносно першої сітки на $\frac{\_hoffset}{2}$ та $\frac{\_voffset}{2}$;
- 2DArcReplacer перетворює точки на кола з радіусом SIDE_LENG;

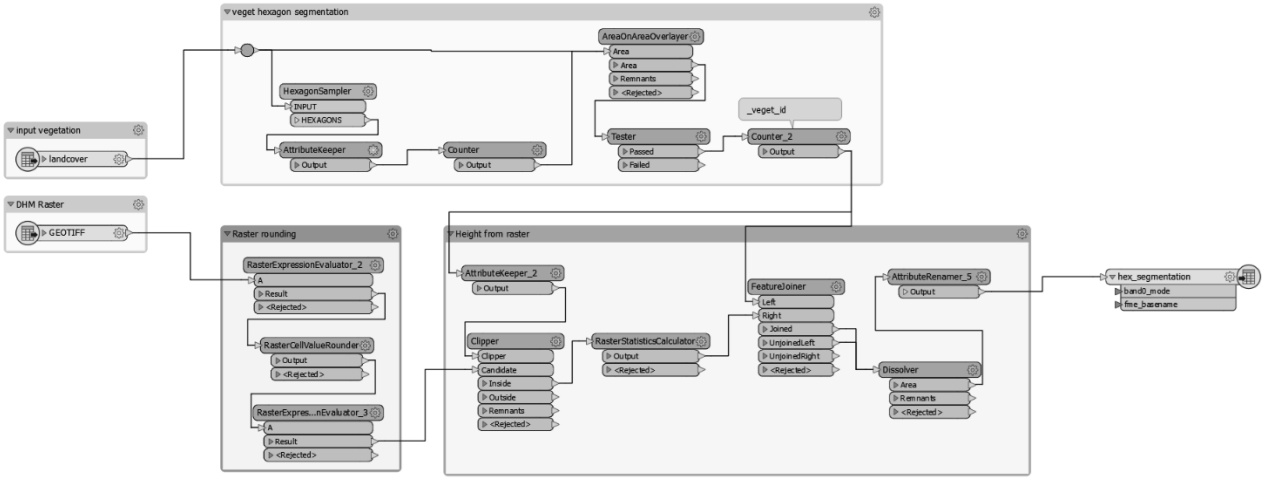


Рис. 1. Реалізація сегментації векторного набору рослинності за висотною характеристикою методом гексагонів у середовищі FME

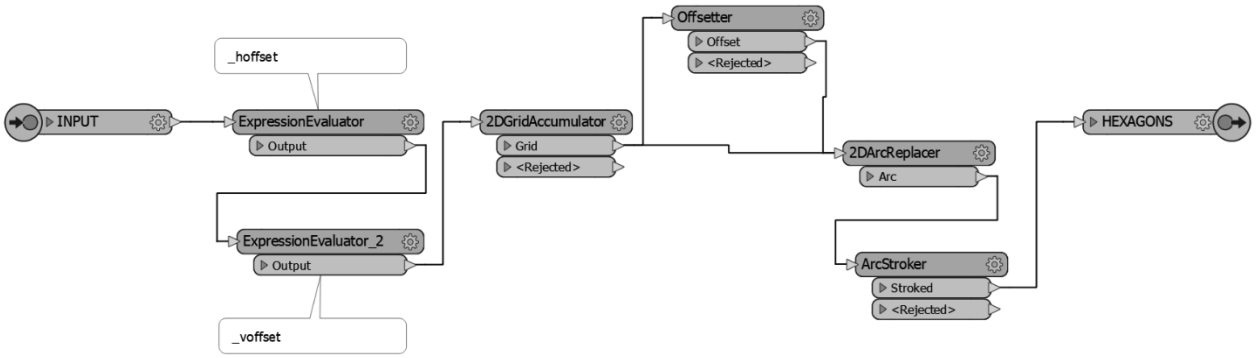


Рис. 2. Загальний вигляд кастомного модуля HexagonSampler

- ArcStroker генералізує коло до шестикутника за заданим числом інтерпольованих граней, яке дорівнює 6.

Результатом роботи HexagonSampler є послідовна нерозривна сітка правильних шестикутників з фіксованою довжиною ребра. AttributeKeeper видаляє непотрібні атрибути, створені HexagonSampler, а Counter створює унікальний id шестикутників, який надалі буде використаний для обробки великих масивів даних окремими частинами. AreaOnAreaOverlayer накладає полігони на полігони, перетинаючи геометрію та спільні атрибути. У нашій моделі він приймає на вхід два набори полігонів: вхідний вектор рослинності та сітку шестикутників. Tester відсіює непотрібні об'єкти, залишаючи лише частини гексагонів, які перетнулися з набором рослинності. Counter_2 створює унікальні id в атрибуті _veget_id, які будуть використані для перенесення значень з растра. Модуль Raster rounding (RasterExpressionEvaluator_2, RasterCellValueRounder, RasterExpressionEvaluator_3) виконує округлення значень пікселів вхідного растра до 3. Оскільки у задачі прописування висот рослинності 3 метри є допустимою похибкою, ця операція допомагає прибрати зайвий «шум» у значеннях. Clipper нарізає вхідний растр по сегментованим полігонам векторної рослинності. На цьому етапі відбувається передача значень _veget_id до растрів. RasterStatisticsCalculator обчислює мінімальне\максимальне\середнє значення пікселів кожного фрагменту растра.

FeatureJoiner відповідає за перенесення мінімального\максимального\середнього значення пікселів із растра до сегментованих полігонів рослинності. Dissolver об'єднує сегменти рослинності з однаковими значеннями висоти, а AttributeRenamer_5 перейменовує атрибути перед записом даних у вихідний файл.

Результатом роботи алгоритму hexagon_segmentation.fmw є векторний шар рослинності, представлений сегментами з різною висотою (Рис. 3).

Рис. 3. Фрагмент векторного шару рослинного покриву. До сегментації – ліворуч, сегментований методом гексагонів з довжиною сторони гексагону 20 метрів та кроком висоти 3 метри – праворуч

Аналіз складності алгоритму AreaonAreaOverlayer

Важливо зазначити, що FME є комерційним програмним забезпеченням із закритим кодом. Внутрішня реалізація алгоритмів, зокрема, методів оптимізації та індексування, залишається недоступною для користувача.

Через це обмеження ми не можемо безпосередньо проаналізувати алгоритмічну складність, переглянувши вихідний код чи документацію, тому було ухвалено рішення дослідити поведінку алгоритму експериментальним шляхом. Зокрема, ми дослідили вплив характеристик вхідних даних (кількість полігонів, їхня форма та кількість вершин) на час виконання алгоритму та використання оперативної пам'яті.

Були створені контрольовані тестові набори даних із чітко визначеними характеристиками — кількістю полігонів, кількістю перетинів та складністю геометрії (кількістю вершин). Це дозволило нам емпірично оцінити залежність часу обробки та витрат ресурсів від параметрів вхідних даних.

У дослідженні ми також детально проаналізували роботу трансформера AreaOnAreaOverlayer, який відіграє ключову роль у процесі сегментації. AreaOnAreaOverlayer виконує просторовий аналіз та створює нові об'єкти на перетинах вхідних полігонів. Немає інформації, які саме методи оптимізації використовуються у FME, але припускаємо, що там може бути застосоване просторове індексування, наприклад, R-дерева. Аналіз часу виконання алгоритму AreaOnAreaOverlayer дозволяє наблизитись до розуміння його ефективності та окреслити межі продуктивності методу гексагонів для сегментації великих масивів геопросторових даних.

Особлива увага приділялася аналізу часу виконання, враховуючи потенційне просторове індексування та геометричні операції. Для цього було створено тестові набори полігональних даних з різними

характеристиками та зроблено вимірювання продуктивності алгоритму на наборах даних розміром від 20 000 до 10 000 000 полігонів. Тестові набори даних генерувались із відомими кількостями полігонів, перетинами полігонів, вершинами полігонів. Було експериментально виміряно час роботи алгоритму AreaOnAreaOverlayer та кількість ресурсів (оперативної пам'яті), необхідних для обробки конкретного датасету.

Для перевірки гіпотези, що загальна кількість вершин полігонів має вплив на час роботи алгоритму, ми створили 2 типи наборів тестових даних: 'round' і 'squere'. Перший тип (round) – це сферичні полігони, які утворені з кола з кутом інтерполяції вершин 22,5гр. та зміщенням по осях X та Y (Рис. 4а). Другий тип (squere) – це полігони у формі квадратів, які зміщені один від одного на 1\4 довжини сторони квадрата по осях X та Y (Рис. 4б).

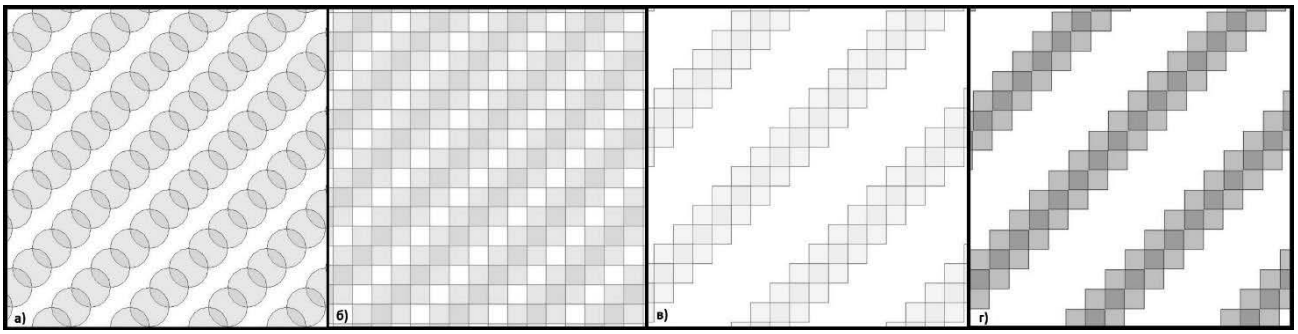


Рис. 4. Тестові набори для оцінки складності алгоритму AreaOnAreaOverlayer:
а) round_data, б) squere_data, в) v2_squere_data, г) v3_squere_data

Заміри часу та ресурсів проведені на 12 наборах кожного типу даних з загальною кількістю полігонів від 20 000 до 10 000 000. Результати замірів наведені у Таблицях 1, 2, 3, 4:

Таблиця 1

Результати замірів часу роботи та обсягу оперативної пам'яті на тестових даних (round_data)

Features_Count	Memory_KB	time_fin
20 000	252 872	2.8
100 000	911 008	13.1
1 000 000	8 323 876	151.6
2 000 000	17 502 468	316.2
3 000 000	26 613 348	541.2
4 000 000	30 941 884	804.8
5 000 000	41 899 008	1 131.4
6 000 000	46 550 676	1 547.4
7 000 000	61 031 896	2 090.4
8 000 000	65 557 424	2 691.9
9 000 000	70 041 100	3 256.1
10 000 000	74 203 936	4 191.9

Таблиця 2

Результати замірів часу роботи та обсягу оперативної пам’яті на тестових даних (squere_data)

Features_Count	Memory_KB	time_fin
20 000	334 408	3.6
100 000	1 277 492	16.5
1 000 000	12 935 884	205.8
2 000 000	22 070 072	500.4
3 000 000	32 607 576	861.8
4 000 000	47 216 188	1 373.4
5 000 000	51 416 280	2 011.2
6 000 000	71 398 116	2 861.8
7 000 000	74 950 820	3 747.1
8 000 000	104 255 576	6 227.3
9 000 000	104 426 492	9 888.2
10 000 000	104 196 532	10 370.2

Таблиця 3

Результати замірів часу роботи та обсягу оперативної пам’яті на тестових даних (v2_squere_data)

Features_Count	Memory_KB	time_fin
20 000	272 340	3.1
100 000	967 692	15.6
1 000 000	9 790 160	190.6
2 000 000	21 019 932	398.3
3 000 000	31 670 228	677.2
4 000 000	35 156 372	1 008.9
5 000 000	48 899 756	1 523.6
6 000 000	53 473 136	2 065
7 000 000	72 671 100	2 782.3
8 000 000	76 267 516	3 762.8
9 000 000	79 681 916	5 350.1
10 000 000	82 591 852	6 639.2

Таблиця 4

Результати замірів часу роботи та обсягу оперативної пам’яті на тестових даних (v3_squere_data)

Features_Count	Memory_KB	time_fin
20 000	269 380	2.2
100 000	964 524	13.1
1 000 000	7 535 360	166.5
2 000 000	16 001 076	376.4

3 000 000	23 667 180	644.8
4 000 000	34 240 536	838
5 000 000	37 579 208	1 057.7
6 000 000	51 428 532	1 565.8
7 000 000	54 452 632	2 199.7
8 000 000	58 342 284	2 874.3
9 000 000	77 111 808	3 633.3
10 000 000	80 141 216	3 421.1

Після отримання результатів було виявлено, що перший тип даних (квадрати) обробляється суттєво довше, та потребує більше оперативної пам'яті ніж другий тип, хоча має меншу загальну кількість вершин. Ми висунули припущення, що причиною такої роботи алгоритму була кількість взаємних перетинів полігонів.

Якщо вважати a,b,c,d…рядами полігонів у тестовому наборі, а 1,2,3,4,5… їхнім порядковим номером у напрямку зміщення, то, наприклад, у першому наборі полігонів round_data (Рис. 4а), об'єкт b5 має перетин лише з двома сусідніми об'єктами свого ряду b4 та b6. А полігон b5 з набору squere_data (Рис. 4б), має шість перетинів: b3,b4,b6,b7,a5,c5.

Для підтвердження гіпотези, що кількість перетинів суттєво впливає на час роботи та необхідну кількість ресурсів, ми ухвалили рішення створити ще 2 типи наборів: squere_v2 та squere_v3. У squere_v2 кількість взаємних перетинів була зменшена з 6 до 4, а у squere_v3 до двох.

У ході експерименту було виявлено, що кількість пар об'єктів, які перетинаються, мають значно більший вплив на час роботи, ніж кількість вершин у цих об'єктах. На графіках (Рис. 5) зображено відношення швидкості обробки даних до кількості об'єктів та кількості вершин. З цих графіків можна зробити висновок, що швидкість роботи алгоритму AreaOnAreaOverlayer більше залежить від кількості взаємних перетинів полігонів, ніж від кількості вершин у цих полігонах. Також швидкість обробки даних за одиницю часу суттєво зменшується зі збільшенням обсягу даних. Тому для подальшої адаптації архітектури методу сегментації для обробки великих масивів даних потрібно враховувати ці фактори.

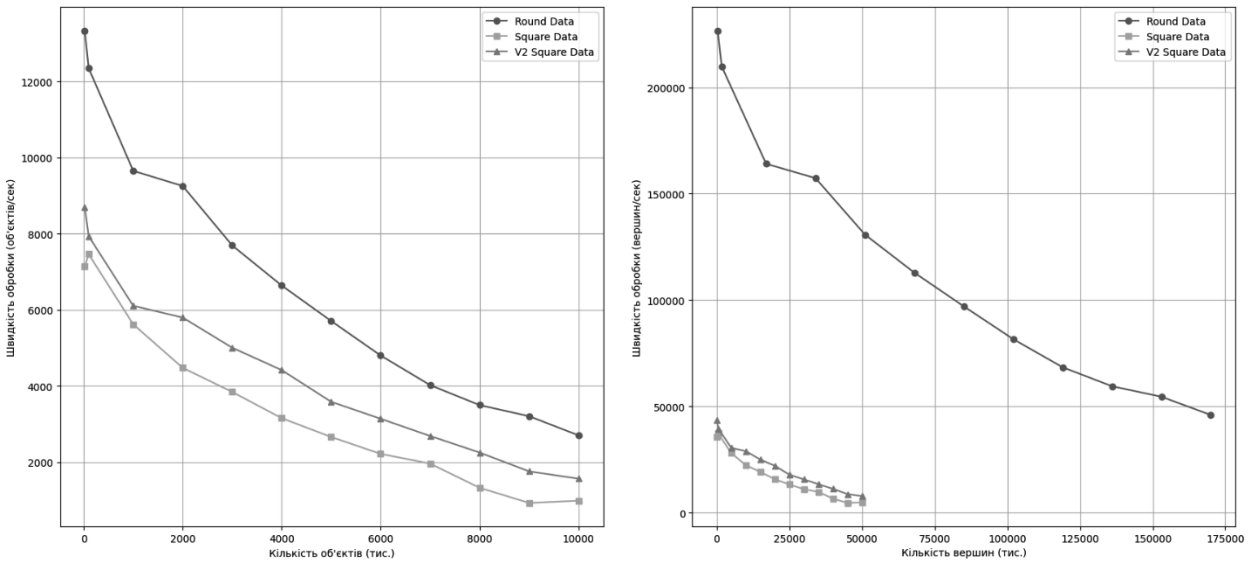


Рис. 5. Графіки відношення швидкості обробки даних до кількості об'єктів та кількості вершин

Під час обробки даних FME читає і оперує даними в оперативній пам'яті (ОП). Це дозволяє ефективно обробляти дані, але водночас створює потребу у достатньому обсязі ОП для обробки наявного масиву даних. Наскільки впливає обсяг доступної ОП на час обробки видно на прикладі тестових наборів прямокутників «squere» (Рис. 6). У разі встановлених 128Гб оперативної пам'яті на ПК, FME процес стає помітно повільнішим досягнувши 80% об'єму наявної ОП. Досягнувши ліміту по ОП, FME починає використовувати дисковий кеш (якщо він доступний в операційній системі). Додаткові процеси оптимізації використаної пам'яті, зайвий час запису\читання даних із диска пояснюють суттєве збільшення часу роботи алгоритму.

Для оцінки часової складності роботи алгоритму AreaOnAreaOverlayer ми розрахували значення кутового коефіцієнта k за отриманими графіками (Рис. 6, 7, 8).

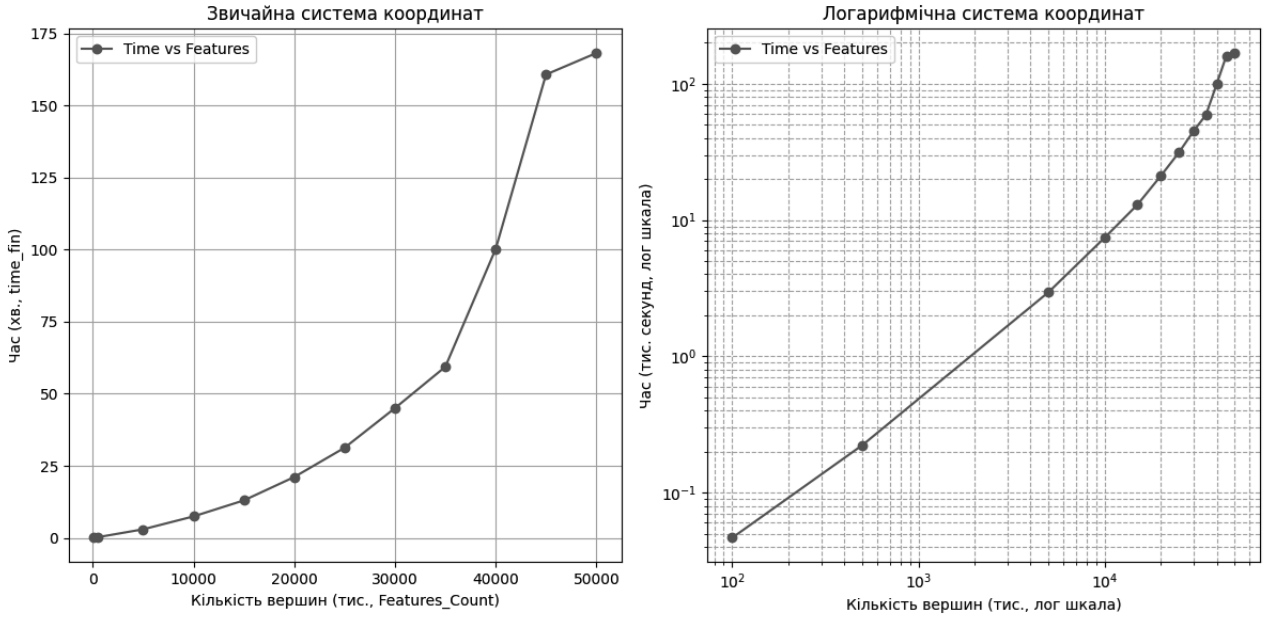


Рис. 6. Графіки часу виконання алгоритму AreaOnAreaOverlayer на наборі squere_data

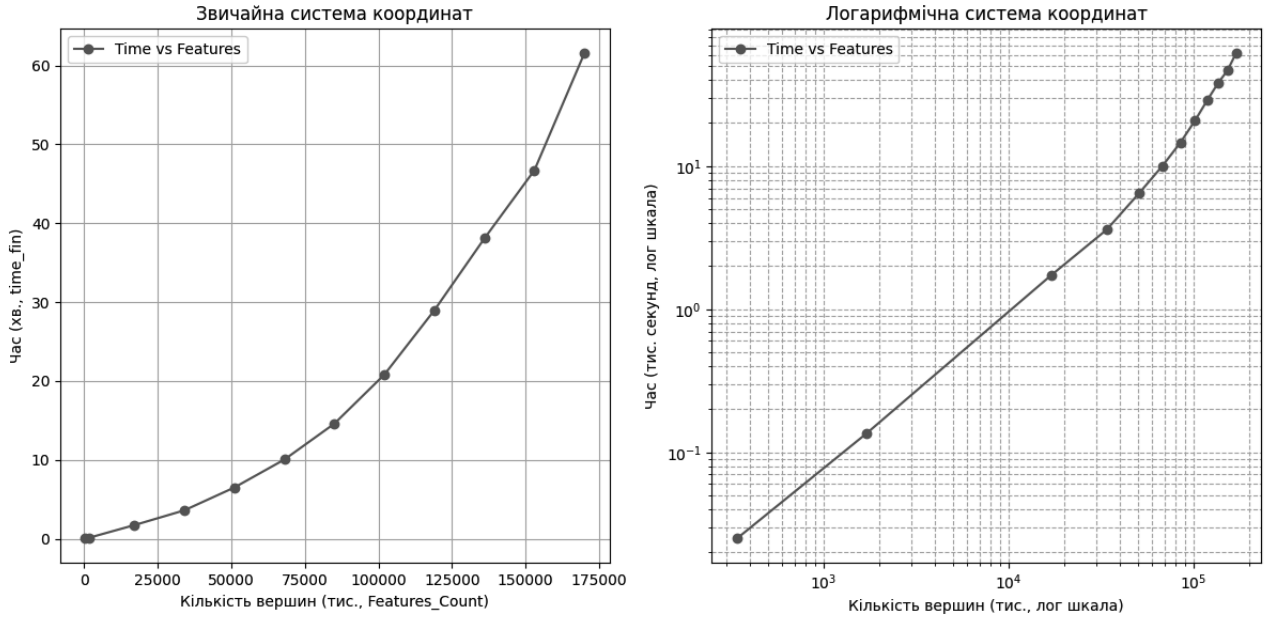


Рис. 7. Графіки часу виконання алгоритму AreaOnAreaOverlayer на наборі round_data

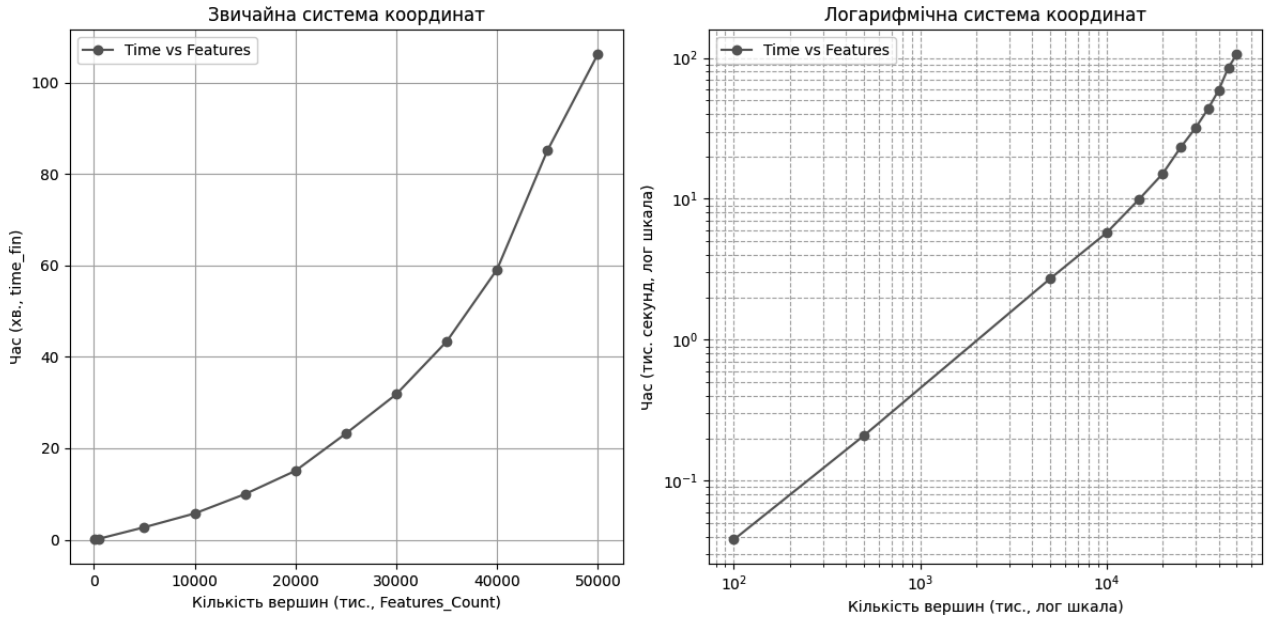


Рис. 8. Графіки часу виконання алгоритму AreaOnAreaOverlayer на наборі squere_data_v2

Розрахуємо коефіцієнт нахилу K з рівняння прямої (1) у логарифмічній системі координат:

$$Y' = KX' + B \quad (1),$$
$$X' = \log_{10}(X),$$
$$Y' = \log_{10}(Y).$$

Візьмемо з графіків дві координати з 1-го та 4-го набору: X_1=20 000, X_2=2 000 000, а значення Y_1 та Y_2 визначимо відповідно до часу роботи кожного типу наборів даних.

Формула для обчислення коефіцієнта нахилу:

$$K = \frac{X_2 - X_1}{Y_2 - Y_1}$$

Розраховані значення для різних наборів даних: round_data: K=1.0264, square_data: K=1.0715, square_data_v2: K=1.0544, square_data_v3: K=1.1166.

Середній кутовий коефіцієнт у логарифмічній шкалі за всіма наборами даних становить 1.06. Це означає, що алгоритм має майже лінійну складність.

Висновки

У цьому дослідженні було проведено аналіз алгоритму AreaOnAreaOverlayer у середовищі FME з метою оцінки часу виконання та впливу різних характеристик вхідних векторних полігонів на час обробки та використання ресурсів. Отримані результати вказують на те, що найбільший вплив на швидкість виконання алгоритму має кількість взаємних перетинів полігонів, а не їхня загальна складність (кількість вершин). Це випливає з порівняння отриманих даних за наборами round_data та square_data: набір round_data обробляється помітно швидше та потребує суттєво менше оперативної пам'яті внаслідок меншої кількості взаємних перетинів полігонів у наборі.

Іншим важливим фактором є об'єм доступної оперативної пам'яті, оскільки FME обробляє дані в оперативній пам'яті, що забезпечує швидке виконання алгоритму. Але досягнувши 80% заповнення доступної оперативної пам'яті, швидкість обробки суттєво знижується через використання дискового кешування.

Отримані результати є ключовими для подальшої розробки архітектури автоматизованого пайп-лайну по сегментації геопросторових растрів. Попри майже лінійну складність алгоритму, немає потреби у розділенні вхідного набору на частини. Однак варто враховувати об'єм доступної оперативної пам'яті, оскільки під час обробки великих масивів даних, ресурсів може не вистачити для успішного результату. У подальшій роботі ми плануємо дослідити залежність розміру вхідних даних від обсягу оперативної пам'яті, необхідної для роботи алгоритму, та розробити мо-

дуль, який буде відповідати за розподілення обчислень.

References

1. Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. http://arxiv.org/abs/1505.04597.
2. He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep Residual Learning for Image Recognition*. http://arxiv.org/abs/1512.03385
3. S. Liu, et al, The overlooked contribution of trees outside forests to tree cover and woody biomass across Europe. Science Advances 9(37), 2023, doi: 10.1126/sciadv.adh4097.
4. Liu S, Brandt M, Nord-Larsen T, Chave J, Reiner F, Lang N, Tong X, Ciais P, Igel C, Pascual A, Guerra-Hernandez J, Li S, Mugabowindekwe M, Saatchi S, Yue Y, Chen Z, Fensholt R. The overlooked contribution of trees outside forests to tree cover and woody biomass across Europe. Sci Adv. 2023 Sep 15;9(37):eadh4097. doi: 10.1126/sciadv.adh4097. Epub 2023 Sep 15. PMID: 37713489; PMCID: PMC10881069.
5. P. Hofmann, D. Tiede,Image segmentation based on hexagonal sampling grids, South-Eastern European Journal of Earth Observation and Geomatics 3, 2014 pp. 173-177.
6. J.N. Mueller, J.N. Corcoran, A Random Point Initialization Approach to Image Segmentation with Variational Level-sets. 2021, http://arxiv.org/abs/2112.12355.
7. R. Douss, I.R Farah, Extraction of individual trees based on Canopy Height Model to monitor the state of the forest. Trees, Forests and People 8, 2022, doi: 10.1016/j.tfp.2022.100257
8. W. Li, Z. Niu, S. Gao, N. Huang, H. Chen, Correlating the horizontal and vertical distribution of LiDAR point clouds with components of biomass in a Picea crassifolia forest. Forests 5(8), 2014, pp. 1910–1930. doi: 10.3390/f5081910.
9. E. Lindberg, J. Holmgren, H. Olsson, Classification of tree species classes in a hemi-boreal forest from multispectral airborne laser scanning data using a mini raster cell method. International Journal of Applied Earth Observation and Geoinformation 100, 2021, doi: 10.1016/j.jag.2021.102334.
10. M.K. Jakubowski, W. Li, Q. Guo, M. Kelly, Delineating individual trees from lidar data: A comparison of vector- and raster-based segmentation approaches. Remote Sensing 5(9), 2013, pp. 4163–4186. doi: 10.3390/rs5094163.
11. Tsaryniuk, O., Hlybovets, A., & Oletsky, O. (2023). *Automated Pipelines for Large-Scale Height-Based Vegetation Segmentation*.
12. Safe Software, FME: Feature Manipulation Engine[cited 04.05.2025]: https://www.safe.com/fme.
13. FME Hub, HexagonSampler, [cited 04.05.2025]: https://hub.safe.com/publishers/larry/transformers/hexagonsampler.

Одержано: 06.05.2025
Внутрішня рецензія отримана: 14.05.2025
Зовнішня рецензія отримана: 15.05.2025

Про авторів:
Царинюк Олександр Васильович,
PhD Комп'ютерні науки, аспірант.
https://orcid.org/0000-0003-1394-2040

Глибовець Андрій Миколайович,
доктор технічних наук, професор,
декан.
https://orcid.org/0000-0003-4282-481X

Місце роботи авторів:
Національний університет
«Києво-Могилянська академія»
https://www.ukma.edu.ua/

УДК 004.738 https://doi.org/10.15407/pp2025.02.098

Р.М. Федоренко, Ю.В. Бова, К.Ю. Юрченко, О.В. Симоненко, І.Р. Федоренко

МОДЕРНІЗАЦІЯ СИСТЕМИ УПРАВЛІННЯ ГОСПОДАРСЬКО-МАЙНОВИМ КОМПЛЕКСОМ НАЦІОНАЛЬНОЇ АКАДЕМІЇ НАУК УКРАЇНИ

Для ефективного управління господарсько-майновим комплексом Національної академії наук України 2023 року Інститутом програмних систем НАНУ було створено та впроваджено комп'ютерну інформаційно-аналітичну систему. Вона отримала назву Цифрова система управління нерухомим майном НАН України. Система призначена для управління обліком, зберіганням та використанням достовірної та актуальної інформації щодо нерухомого майна, а також - ведення договорів оренди та користування. Система розроблена з використанням сучасної програмної платформи без використання програмного забезпечення країни-агресора. В статті описана модернізована версія системи, в якій було впроваджено нові підсистеми та вдосконалено вже існуючі, що дає змогу більш ефективно управляти майновим комплексом Національної академії наук України.

Ключові слова: управління нерухомим майном, ERP-система, веб-система, автоматизація бізнес процесів

R.M. Fedorenko, Y.V. Bova, K.Y. Yurchenko, O.V. Symonenko, I.R. Fedorenko

MODERNIZATION OF THE ECONOMIC AND PROPERTY COMPLEX MANAGEMENT SYSTEM OF THE NATIONAL ACADEMY OF SCIENCES OF UKRAINE

In 2023 the Institute of Software Systems of the National Academy of Sciences of Ukraine created and implemented a computer information and analytical system known as the Digital Real Estate Management System of the NASU. It was aimed at creating effective management of the NASU as a unified economic and property complex. The system is designed to manage the accounting, storage and use of reliable and up-to-date information on real estate, as well as lease and use agreements. The system was developed using a modern software platform without using software from the aggressor country. The article describes a modernized version of the system, in which new subsystems were introduced and existing ones were improved. That allows more efficient management of the property complex of the National Academy of Sciences of Ukraine.

Keywords: real estate management, ERP system, websystem, business process automation

Вступ

Цифрова система управління нерухомим майном НАН України (далі – ЦС "Майно") є єдиним інформаційним простором електронної взаємодії підприємств, організацій, установ НАН України для інформаційно-аналітичного забезпечення та підтримки процесів управління нерухомим майном НАН України.

Призначенням ЦС "Майно" є автоматизація процесів управління нерухомим майном НАН України, збирання, накопичення, оновлення, збереження, захист, ефективний пошук, представлення інформації щодо балансоутримувачів, нерухомого майна, яке обліковується на балансі установ, організацій, підприємств НАН України, договорів оренди та користування, іншої інформації для забезпечення ефективного управління нерухомим майном НАН України [1].

Наразі ЦС "Майно" вже успішно функціонує та забезпечує [2]:

– облік об'єктів нерухомості НАН України, аналіз та супроводження їхнього стану за кожною стадією життєвого циклу;

– формування бази даних об'єктів нерухомості, установ, організацій, підприємств НАН України;

– організацію взаємодії та автоматизованого обміну даними між усіма учасниками, залученими до процесу використання та експлуатації об'єктів нерухомості НАН України.

Впровадження ЦС “Майно” в діяльність Управління справами НАН України дозволило:

– об'єднати інформаційні ресурси щодо об'єктів нерухомості НАН України;

– забезпечити безперервний моніторинг процесів управління нерухомим майном;

– організувати колективну роботу виконавців із прозорим контролем виконання робіт;

– забезпечити прозорість технологічних процесів обробки документів і даних;

– забезпечити формування та обробку статистичних даних за різними критеріями.

Основні принципи та підходи розробки ЦС “Майно”

Загальна ідея впровадження ЦС “Майно” передбачала, що вона стане одним із модулів майбутньої корпоративної ERP-системи управління НАН України.

Було здійснено аналіз наявних програмних рішень з метою вибору системи, яка б відповідала таким критеріям: сучасна технологічна платформа, масштабованість, відкритий програмний код та приналежність до продуктів національного виробництва. У разі відсутності готового модуля для управління господарсько-майновим комплексом НАН України, передбачалась можливість впровадження прототипу [1].

Для створення ЦС “Майно” було обрано програмний комплекс «Стратег.Смарт» – «Комплекс програмних модулів для побудови та експлуатації систем електронного документообігу, електронної взаємодії та самоврядування» тому, що система базується на відкритих технологіях. А це забезпечує прозорість, контроль, гнучкість та можливість масштабування та враховує особливості управління нерухомістю державного сектора.

Під час розробки ЦС “Майно” було враховано такі ключові принципи [1]:

– єдина інформаційна платформа для створення системи на всіх рівнях управління;

– інтеграція різнорідних інформаційних ресурсів у межах спільної концептуальної моделі, яка утворює метабазу системи – незалежно від форм подання, джерел, способів оновлення чи рівня достовірності;

– типовість рішень та уніфікований підхід до проєктування і реалізації всіх підсистем;

– відкритість архітектури – система легко модифікується і масштабується завдяки модульній структурі, клієнт-серверній архітектурі та відкритим інтерфейсам, що дає змогу інтегрувати її з іншими рішеннями;

– поступове впровадження системи як масштабного проєкту, що потребує значних ресурсів;

– поетапне розширення функціоналу з можливістю адаптації і внесення змін на кожному етапі;

– раціональний вибір технологій – програмні та апаратні засоби підбирались з урахуванням ефективності, надійності та зручності експлуатації;

– простота для користувачів – система розрахована на базовий рівень володіння ПК, достатньо знати Windows, програми Microsoft Office та сучасні веббраузери;

– надійність і захищеність – висока стійкість до збоїв і відповідність вимогам інформаційної безпеки;

– масштабованість – підтримка роботи багатьох користувачів, з можливістю нарощування ресурсів за рахунок більш потужного обладнання;

– розподіленість – підтримка віддаленого доступу користувачів;

– колективна робота – можливість спільної обробки документів кількома командами виконавців, контроль за виконанням завдань, автоматичне оповіщення відповідальних осіб про критичні ситуації або затримки;

– прозорість процесів – технологічний процес обробки даних є спільним і доступним усім користувачам, незалежно від їхнього розташування;

– врахування кращих практик – досвід подібних систем, створених в Україні та за кордоном, було проаналізовано та враховано у процесі проєктування.

Розробка була поділена на два основні етапи:

перший: 2023 рік – розробка та впровадження першої версії;

другий: 2024 рік – модернізація системи та розробка нового функціоналу.

У першій частині було реалізовано:

– ведення класифікатора об’єктів нерухомого та рухомого майна;

– ведення реєстру балансоутримувачів;

– формування та збирання періодичної звітності від установ, організацій, підприємств НАН України.

У другій частині було реалізовано:

– ведення реєстру об’єктів нерухомого майна (земельні ділянки, будівлі/споруди, об'єкти оренди та користування);

– ведення договорів оренди та користування;

– формування консолідованої звітності щодо використання нерухомого майна для Управління справами НАН України та Президії НАН України.

Бізнес-процеси, автоматизовані в ЦС “Майно”

На основі обстеження діяльності з управління нерухомим майном в ЦС “Майно” автоматизовано такі основні бізнес-процеси [2]:

– ведення реєстру балансоутримувачів;

– ведення реєстру орендарів;

– ведення реєстрів будівель, споруд та земельних ділянок;

– ведення договорів оренди та користування;

– формування та збирання періодичної звітності;

– формування консолідованої звітності.

Система забезпечує повний цикл управління даними, а саме:

– облік, зберігання та актуалізація даних щодо балансоутримувачів;

– облік, зберігання та актуалізація даних щодо будівель, споруд та земельних ділянок;

– облік, зберігання та актуалізація даних щодо договорів оренди та користування;

– автоматизація збирання та формування звітності.

Основні функції ЦС “Майно”

Система забезпечує виконання таких основних функцій:

– ідентифікація та автентифікація користувачів через Active Directory;

– розмежування прав доступу відповідно до ролей користувачів;

– введення, зберігання та обробка даних про об'єкти нерухомості;

– оперативний обмін електронними документами;

– формування статистичних та аналітичних звітів;

– ведення нормативно-довідникової інформації;

– використання кваліфікованого електронного підпису;

– захист даних від несанкціонованого доступу.

Архітектура ЦС “Майно”

Як програмну архітектуру у ЦС МАЙНО використано трирівневий варіант архітектури "клієнт-сервер" з тонким клієнтом. В системі застосовується сервер баз даних Microsoft SQL Server 2019 Standard. Система побудована за ієрархічною схемою: сервер баз даних – сервер додатків – клієнтське робоче місце.

Інтерфейс веб-системи клієнтської частини повністю веб-орієнтований, без необхідності додаткового встановлення ПЗ на комп'ютери користувачів. Система функціонує в комп'ютерній мережі НАН України на окремих робочих станціях з використанням хмарної інфраструктури.

Виконання повного переліку задач ЦС “Майно” вирішується взаємодією наступних програмних модулів та сервісів:

– модуль ведення реєстру балансоутримувачів;

– модуль ведення реєстру орендарів;

– модуль ведення реєстру будівель та споруд;

– модуль ведення реєстру земельних ділянок;

– модуль ведення реєстру об’єктів нерухомого майна;

– модуль ведення договорів оренди та користування;

– модуль формування щоквартальних довідок про нерухоме майно;

– модуль формування консолідованої звітності;

– модуль формування звітності;

– модуль адміністрування;

– модуль керування ролями користувачів;

– модуль управління користувачами;

– модуль ведення нормативно-довідникової інформації;

– модуль ведення організаційно-штатної структури;

– модуль авторизації користувачів через Active Directory;

– сервіс використання кваліфікованого електронного підпису;

– сервіс формування друкованих форм;

– сервіс маркування документів штрих-кодом та/або QR-кодом;

– сервіс пошуку інформації;

Загальне меню ЦС “Майно”

Загальне меню ЦС “Майно” містить такі розділи (рис. 1):

– довідки;

– об'єкти нерухомості;

– договори оренди;

– юрособи;

– класифікатор;

– НДІ;

– адміністрування;

– ОШС;

– звітність.

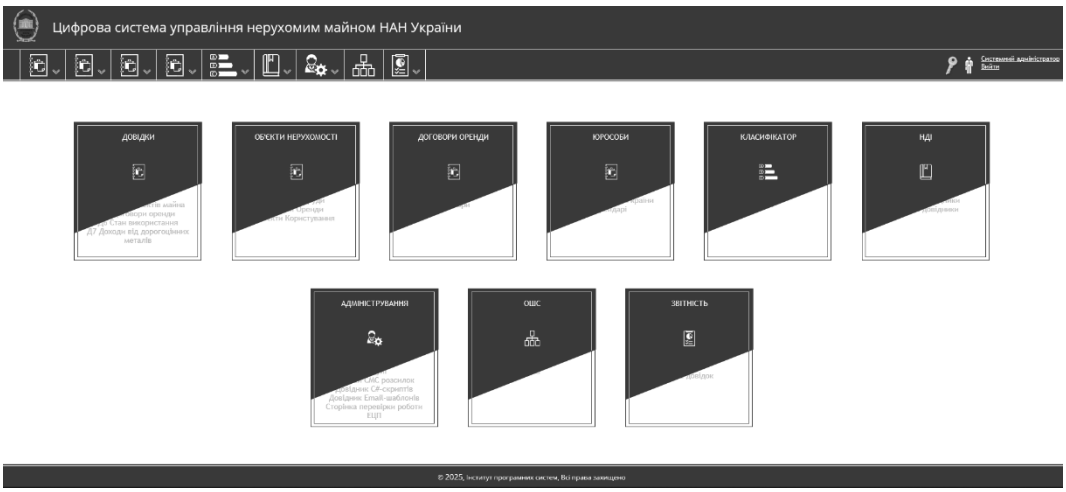


Рис. 1. Загальне меню ЦС “Майно”

Меню «Довідки» містить підменю:
– Д1 Про використання майна;
– Д2 Про заборгованість;
– Д3 Майно, надане для розміщення;
– Д4 Перелік об'єктів майна;
– Д5 Договори оренди;
– Д6 Стан використання;
– Д7 Доходи від дорогоцінних металів.
Меню «Юрособи» містить підменю:
– організації НАН України;
– орендарі.
Меню «НДІ» містить підменю:
– загальні довідники;
– складні довідники.
Меню «Звітність» містить підменю:
– звітність;
– архів довідок.

Класифікатор об'єктів нерухомого та рухомого майна

Важливим компонентом ЦС “Майно” є Класифікатор об'єктів нерухомого та рухомого майна (далі – КМ), який забезпечує єдину систему класифікації всіх типів майна НАН України. Проєктні

рішення щодо Класифікатора об’єктів нерухомого та рухомого майна ґрунтуються на створенні розгалуженого класифікатора об’єктів [2].

Структурно КМ побудований з урахуванням видів майна, де кожному виду надано відповідну кодову позначку. Класифікація гармонізована з державним класифікатором будівель та споруд ДК 018-2000 та класифікатором державного майна Фонду державного майна України.

Кожна позиція КМ містить одинадцятизначну цифрову кодову позначку і назву відповідних класифікаційних угрупувань. Угруповання об'єктів побудовані за ієрархічним методом класифікації з використанням фасетних переліків.

Інтерфейс КМ дає змогу зручно переглядати всю структуру класифікатора та редагувати окремі позиції з урахуванням змін у законодавстві чи потреб обліку.

Проєктні рішення щодо бізнес-процесу формування та збирання періодичної звітності

Проєктні рішення щодо бізнес-процесу формування та збирання періодичної звітності від установ, організацій, підприємств НАН України з питань управління нерухомим майном ґрунтуються на створенні відповідних засобів підготовки та формування звітності установами, організаціями, підприємствами НАН України з наступним поданням її в Управління справами НАН України.

Програмний інтерфейс формування довідки Д1 про використання майна НАН України установами, організаціями та підприємствами НАН України наведено на рис. 2.

Програмна реалізація формування довідки про використання майна НАН України дає змогу користувачам автоматично сформувати довідку у формі PDF-документа, підписати її електронно-цифровим підписом та подати засобами ЦС “Майно” до Управління справами НАН України. Форма PDF-документа, який створюється в системі, наведена на рис. 3.

Проєктні рішення щодо бізнес-процесу формування консолідованої звітності з використання нерухомого майна

Проєктні рішення щодо бізнес-процесу формування консолідованої звітності з використання нерухомого майна розроблено з метою забезпечення ефективного управління активами НАН України. Основна ідея полягає у створенні єдиної централізованої системи, що дає змогу фахівцям Управління справами НАН України оперативно підготувати, консолідувати та аналізувати звітність щодо нерухомого майна.

ЦС “Майно” забезпечує автоматизовану підготовку звітів на основі даних, які вводяться користувачами. Це дає змогу значно скоротити час на формування звітності, знизити ризики помилок і забезпечити прозорість.

Основні функціональні можливості:

– централізований збір даних про нерухомість у реальному часі;

– автоматизація підготовки звітів;

– підтримка різних форматів звітності;

– можливість формування запитів на отримання звітів за конкретними критеріями.

У ЦС “Майно” було реалізовано наступний перелік консолідованих звітів:

КД0 та КД0-1: Дають змогу моніторити стан подання щоквартальної звітності, забезпечуючи контроль за дотриманням термінів.

КД1: Деталізований огляд щодо використання приміщень та нарахувань орендної плати, допомагає оптимізувати управління майном.

КД2: Звіт надає дані про заборгованості орендарів, що дає змогу вчасно вживати заходів для погашення боргів.

КД3: Включає відомості про майно, яке використовується сторонніми організаціями, що сприяє ефективному управлінню ресурсами.

КД4: Містить перелік об'єктів, доступних для оренди, що уможливлює аналіз ефективності використання майна.

КД5: Звіт про активні договори оренди, включаючи дати укладання та закінчення, характеристики об'єкта, що дозволяє відстежувати умови оренди.

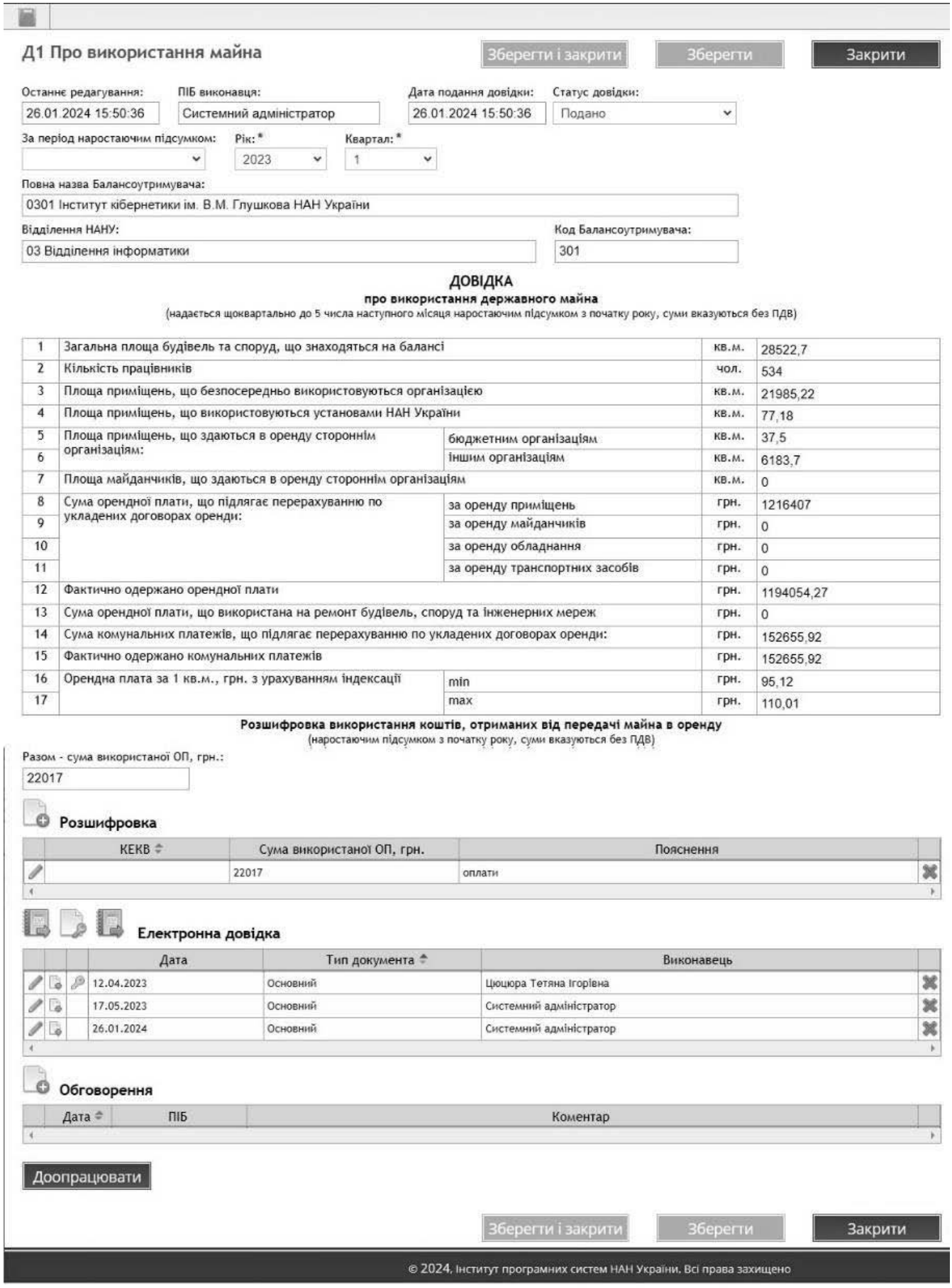


Рис. 2. Програмний інтерфейс бізнес-процесу підготовки та формування довідки Д1

Д О В І Д К А

про використання державного майна

0201 Інститут математики НАН України

(найменування підприємства, організації, установи НАН України)

станом на 22 листопада 2022 року

(надається щоквартально до 5 числа наступного місяця наростаючим підсумком з початку року, суми вказуються без ПДВ)

1	Загальна площа будівель та споруд, що знаходяться на балансі		кв.м	12,00
2	Кількість працівників		чол.	2233
3	Площа приміщень, що безпосередньо використовуються організацією		кв.м	
4	Площа приміщень, що використовуються установами НАН України		кв.м	
5	Площа приміщень, що здаються в оренду стороннім організаціям:	бюджетним організаціям	кв.м	11,00
6		іншим організаціям	кв.м	
7	Площа майданчиків, що здаються в оренду стороннім організаціям		кв.м	
8	Сума орендної плати, що підлягає перерахуванню по укладених договорах оренди:	за оренду приміщень	грн.	22333,00
9		за оренду майданчиків	грн.	446,00
10		за оренду обладнання	грн.	
11		за оренду транспортних засобів	грн.	11,00
12	Фактично одержано орендної плати		грн.	
13	Сума орендної плати, що використана на ремонт будівель, споруд та інженерних мереж		грн.	
14	Сума комунальних платежів, що підлягає перерахуванню по укладених договорах оренди		грн.	223,00
15	Фактично одержано комунальних платежів		грн.	
16	Орендна плата за 1м2 офісних приміщень з урахуванням індексації:	min	грн.	444,00
17		max	грн.	5555,00

Розшифровка використання коштів, отриманих від передачі майна в оренду

(наростаючим підсумком з початку року, суми вказуються без ПДВ)

КЕКВ	Сума використаної орендної плати	Детальні пояснення щодо напрямків використання
	грн.	
2300	2000,00	Оплата
4300	1000,00	витрати
Разом:	3000,00	

Головний бухгалтер _____ (підпис) _____ (П.І.Б.)

М.П.

Рис. 3. Форма PDF-документа довідки про використання майна

КД6 та КД6-1: Аналіз ефективності використання приміщень, зокрема, заповненість та функціональність об'єктів.

КД7: Звіт про доходи, отримані від реалізації дорогоцінних металів, що є важливою частиною фінансових показників.

КД10: Комплексний звіт, що охоплює земельні ділянки, будівлі, інженерні споруди НАН України, включаючи інформацію про кількість, площу, державну реєстрацію та статус використання.

Ці звіти формуються за запитом у розділі «Звітність» та містять детальну інформацію про використання, стан та управління нерухомим майном.

Програмний інтерфейс вибору типу консолідованого звіту використання майна наведено на рис. 4.

Програмний інтерфейс для формування консолідованої звітності на прикладі звіту «КД6 Інформація про стан використання приміщень» наведено на рис. 5.

Програмна реалізація формування консолідованих звітів установами, організаціями та підприємствами НАН України дає змогу користувачам автоматично сформувати консолідований звіт у формі PDF-документа. Форма PDF-документа, який створюється в системі, на прикладі звіту «КД6 Інформація про стан використання приміщень», наведена на рис. 6.

Реалізація бізнес-процесу формування консолідованої звітності дозволяє значно підвищити прозорість та ефективність управління нерухомим майном НАН України. Система сприяє зменшенню адміністративного навантаження на фахівців і покращенню якості ухвалення рішень на основі актуальних та достовірних даних.

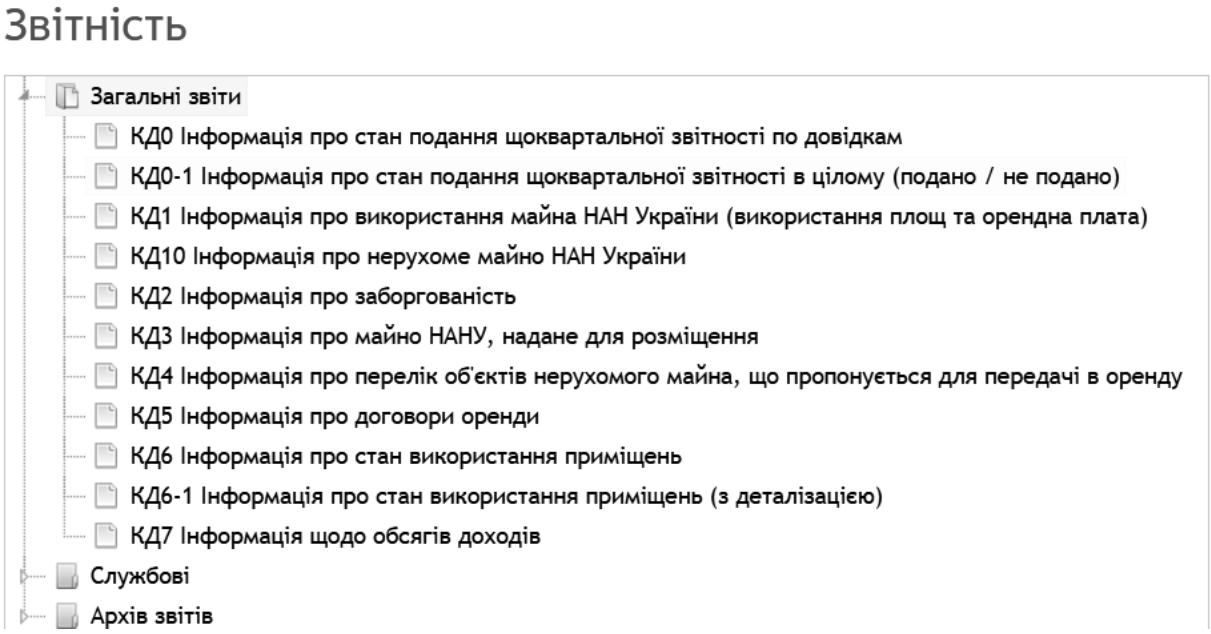


Рис. 4. Програмний інтерфейс для вибору типу консолідованої звітності

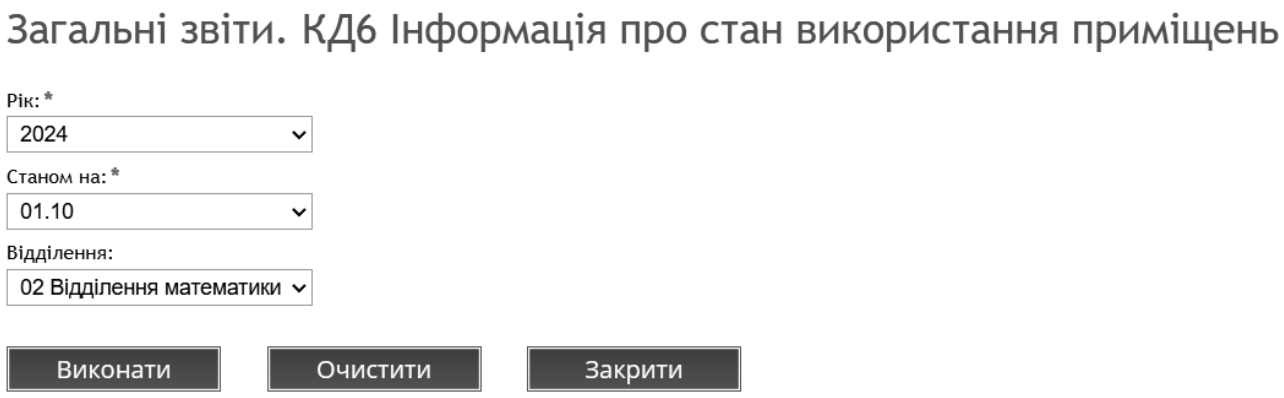


Рис. 5. Програмний інтерфейс для формування консолідованої звітності КД6

Проєктні рішення щодо бізнес-процесу обліку, зберігання та актуалізації даних про земельні ділянки, будівлі, інженерні споруди, об’єкти оренди та об’єкти користування

Проєктні рішення для бізнес-процесу обліку, зберігання та актуалізації даних щодо нерухомого майна НАН України (земельних ділянок, будівель, інженерних споруд, об'єктів оренди та об'єктів користування) створені з метою забезпечення ефективного управління активами академії, підтримки їхньої актуальності та прозорості у доступі до інформації.

Основні програмні модулі:

– модуль обліку земельних ділянок – збір та зберігання інформації про місцезнаходження, площу, кадастрові номери та правовий статус земельних ділянок;

– модуль обліку будівель та інженерних споруд – реєстрація даних про будівлі, включаючи тип об'єкта, призначення, площу, рік побудови та технічний стан;

– модуль управління орендою та користуванням: ведення обліку інформації про орендарів та умови орендної плати; актуалізація даних про об'єкти, що перебувають у користуванні установ та організацій НАН України;

– модуль формування консолідованої звітності на основі даних про об'єкти нерухомості, що забезпечує підтримку управлінських рішень та аналізу;

– модуль інтеграції з бізнес-процесами блоку пов'язані документи, що дозволяє об'єднувати внесені дані та контролювати стан об'єктів у межах обліку нерухомого майна.

КД6 Інформація про стан використання
нежитлових будинків та приміщень, які належать до державної власності
рік 2024 квартал 3 станом на 01.10.2024 року

Виконано 25.03.2025 15:02

Відділення	02 Відділення математики

Разом по НАН України	Кількість працівників, чол.	Загальна площа, кв.м	Корисна площа, кв.м	Площі, які використовуються установою, кв.м	Наявність вільних приміщень, кв.м	Площа оренди державною установою, кв.м	Площа оренди комерційною організацією, кв.м	Площа користування, кв.м
	606	16859,50	14543,10	12755,93	781,20	1206,90	1608,07	1849,60

№ з/п	Назва установи, організації	Назва установи, організаці орендаря	Назва установи, організації користувача	Адреса	Кільк. працівників, чол.	Загальна площа, кв.м	Корисна площа, кв.м	Площі, які використовує установа, кв.м	Наявн. вільних приміщ., кв.м	Площа оренди державн. установою, кв.м	Площа оренди комерц. організацією, кв.м	Початок договору оренди	Закінчення договору оренди	Площа користування, кв.м	Примітки
Відділення		02 Відділення математики													
1	Інститут математики НАН України			01024 м. Київ, вул. Терещенківська, 3	177	5296,90	4886,60	4414,30							
2			Видавничій дім «Академперіодика» НАН України	01024 м. Київ, вул. Терещенківська, 3	7									108,30	
3			Державне підприємство «Науково-виробниче підприємство «Видавництво «Наукова Думка» НАН України	01024 м. Київ, вул. Терещенківська, 3	48									461,20	
4			Інститут енциклопедичних досліджень НАН України	01024 м. Київ, вул. Терещенківська, 3	35									313,10	

Рис. 6. Форма PDF-документа консолідованого звіту на прикладі КД6

Програмний інтерфейс формування картки «Земельні ділянки» організаціями та підприємствами НАН України наведено нижче (рис. 7).

Проєктні рішення для обліку, зберігання та актуалізації даних щодо нерухомого майна НАН України спрямовані на централізацію та оптимізацію управління активами НАН України. Система забезпечує збір і ведення актуальних даних про земельні ділянки, будівлі, інженерні споруди, об'єкти оренди та користування. Реалізовано функціонал для формування облікових карток, що містять ключову інформацію про характеристики об'єктів, їхній стан та правовий статус.

Система також підтримує формування консолідованої звітності, яка надає прозорі й детальні дані для аналізу та ухвалення управлінських рішень. Інтеграція з іншими бізнес-процесами та зручний інтерфейс забезпечують простоту роботи для користувачів і підвищують ефективність управління нерухомим майном НАН України.

Проєктні рішення щодо бізнес-процесу обліку, актуалізації та архівування даних про договори оренди та договори користування

Проєктні рішення для бізнес-процесу оптимізації процесів обліку, актуалізації та архівування даних про договори оренди та користування в установах, організаціях та підприємствах НАН України розроблено ефективну систему управління цими даними. Вона забезпечує централізоване ведення інформації про умови договорів, їхню актуальність, строки дії та фінансові зобов'язання.

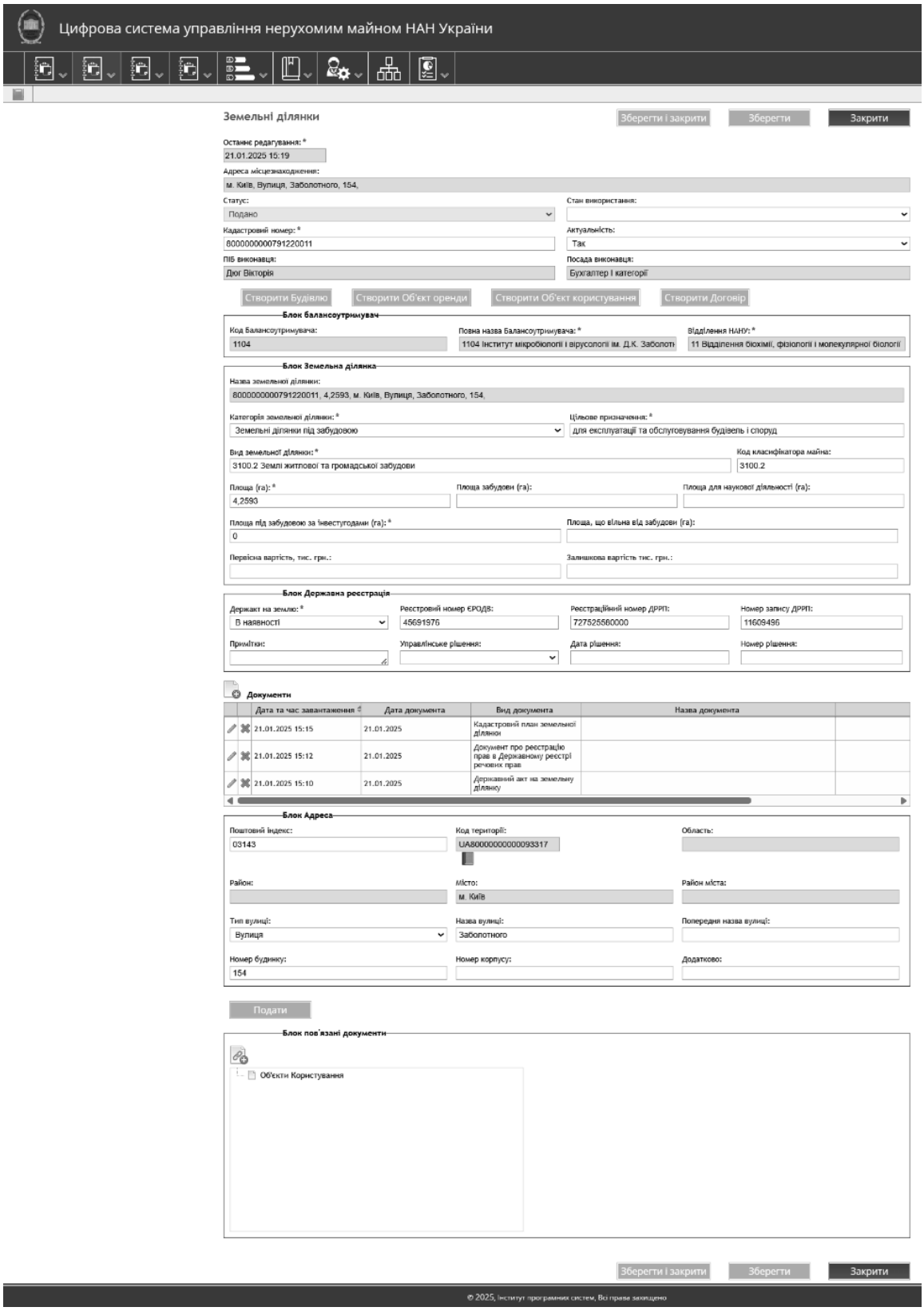


Рис. 7. Програмний інтерфейс картки «Земельні ділянки»

Основні функціональні можливості включають автоматизоване оновлення даних, інтеграцію з іншими бізнес-процесами для контролю виконання умов договорів, а також збереження архівних документів у цифровій формі для історичного аналізу та аудиту.

Програмний інтерфейс формування картки «Будівлі/Споруди», організаціями та підприємствами НАН України наведено нижче (рис. 8).

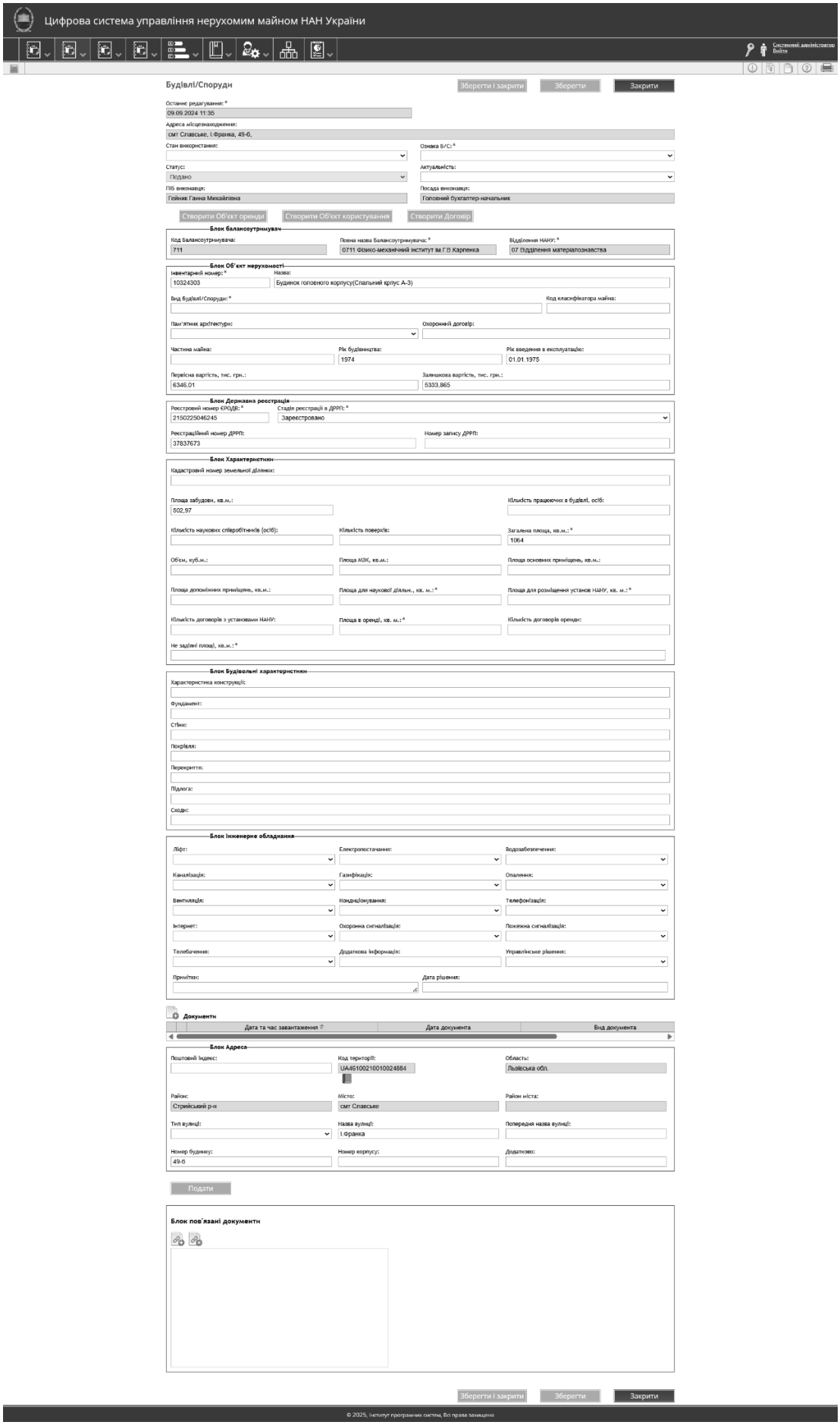


Рис. 8. Програмний інтерфейс картки «Будівлі/Споруди»

Розроблені проєктні рішення забезпечують ефективний облік, актуалізацію та архівування даних про договори оренди та користування, створюючи прозорий і централізований механізм управління цими процесами. Це сприяє підвищенню контролю за виконанням умов договорів, зменшенню ризиків, пов'язаних із втратою або застарілістю даних, та забезпечує зручний доступ до інформації для ухвалення управлінських рішень.

Проєктні рішення щодо бізнес-процесу пов’язання інформації про нерухоме майно

В рамках реалізації цифрової системи управління нерухомим майном НАН України було розроблено та впроваджено систему інтегрованого обліку, збереження та актуалізації інформації про нерухомі об'єкти, включно із земельними ділянками, будівлями/спорудами, об'єктами оренди та об'єктами користування.

Цей процес базується на ієрархічній моделі даних, де земельна ділянка виступає основним об'єктом, до якого можуть бути прив'язані всі елементи нерухомості, але також є можливість створення незалежних структур.

Земельні ділянки. На початковому етапі бізнес-процесу створюється картка земельної ділянки, яка слугує основним інформаційним елементом для обліку. Ця картка включає базову інформацію про об'єкт.

У системі реалізований механізм автоматичного відображення зв’язків між об’єктами, який підтримує як автоматичне, так і ручне налаштування, забезпечуючи адаптивність до специфічних потреб користувачів.

Будівлі/споруди. На основі даних про земельну ділянку формується картка будівлі/споруди. Однією з ключових особливостей картки є розширений сортований класифікатор, що здійснює формування даних про будівлі та споруди. На базі будівель і споруд створюються об’єкти оренди та користування, а згодом – договори, формуючи повну реляційну модель управління майном.

У кожній картці передбачений універсальний блок "Пов’язані документи", який забезпечує автоматизоване відображення асоційованих об’єктів (земельних ділянок, будівель/споруд, об’єктів оренди/користування, договорів) та виконує функції навігації. Цей підхід дає змогу централізовано представляти дані, швидко встановлювати відносини між об’єктами та оптимізувати рутинну роботу з майновою інформацією.

Така структура не лише спрощує облік нерухомого майна, а й підвищує ефективність управління за рахунок інтеграції всіх пов’язаних елементів в єдиній системі.

Блок-схема бізнес-процесу пов’язання інформації наведено нижче (рис. 9).

Реалізований процес є інноваційним підходом до управління майновими активами НАН України, оскільки він автоматизує рутинні операції, забезпечує повну інтеграцію даних між об'єктами та створює єдину систему для роботи з нерухомим майном.

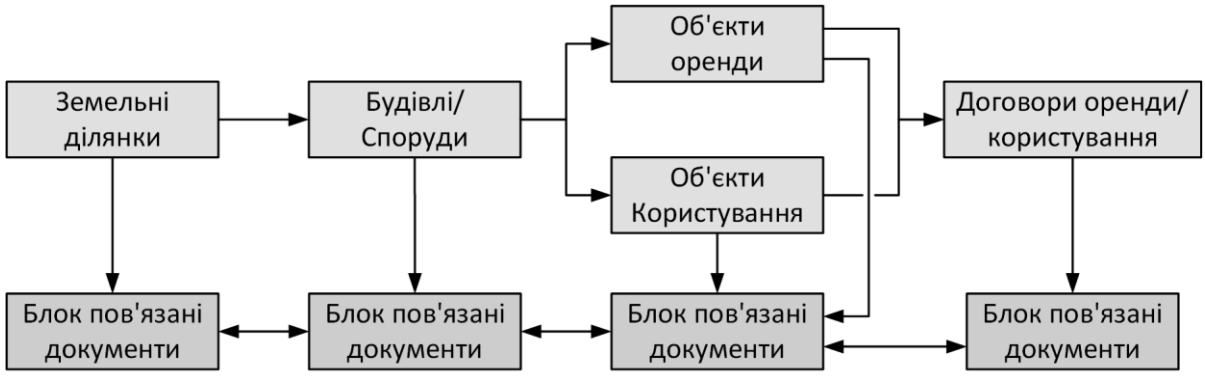


Рис. 9. Блок-схема бізнес процесу пов’язання інформації про нерухоме майно

Висновки

У статті розглянуто здійснений комплекс робіт із модернізації ЦС “Майно” та подальшої автоматизації процесів управління майновими об'єктами НАН України. ЦС “Майно” впроваджено в інститутах НАН України з урахуванням можливостей використання хмарної інфраструктури НАН України.

У результаті досягнуто таких ключових результатів:

– створено централізовану базу даних, яка забезпечує зручний доступ до інформації про майнові об'єкти;

– розроблено функціональні сервіси для автоматизованого обліку, обробки інформації та звітності.

Автоматизація роботи з даними про нерухоме майно НАН України за допомогою ЦС “Майно” забезпечує централізоване управління інформацією про нерухоме майно, підтримує автоматизований облік та зручний доступ до всіх необхідних даних для ефективного управління майновими активами НАН України. Це дозволяє швидше та якісніше виконувати адміністративні функції, а також дає можливість своєчасно отримувати основні показники про майно для підтримки ухвалення оперативних управлінських рішень.

У модернізованій версії ЦС “Майно” було впроваджено нові підсистеми, які дають змогу ефективніше управляти майновим комплексом НАН України, а саме:

– підсистема «Збирання, обліку та актуалізації інформації про нерухоме майно» у складі: «Земельні ділянки», «Будівлі/Споруди» «Об'єкти Оренди» та «Об'єкти Користування»;

– підсистема «Збирання, обліку, актуалізації та зберігання інформації про договори оренди/користування від установ, організацій, підприємств НАН України»;

– підсистема «Формування консолідованої звітності щодо використання нерухомого майна».

Крім цього було доопрацьовано попередні підсистеми:

– класифікатора об’єктів нерухомого та рухомого майна;

– форми «Формування та збирання періодичної звітності від установ, організацій, підприємств НАН України з питань управління нерухомим майном».

Розроблені підсистеми для автоматизації відповідних функцій наукових установ та управління нерухомим майном НАН України наразі є унікальними.

У подальшому заплановано продовжувати супроводження, науково-технічну підтримку та розширення функціоналу ЦС “Майно” , а також здійснювати необхідне доопрацювання програмного забезпечення ЦС “Майно” згідно зі змінами в законодавстві України та Розпорядженнями Президії НАН України.

Література

1. Чадюк А.В, Федоренко Р.М., Курченко О.А. Розроблення і впровадження системи управління господарсько-майновим комплексом НАН України (Development and implementation of the management system of the economic and property complex of the National Academy of Sciences of Ukraine) // Проблеми програмування, 2023, № 2, С. 24-38, https://doi.org/10.15407/pp2023.02.024.
2. Федоренко Р.М., Шевченко В.Л., Ігнатенко П.П. та інші (всього 14 осіб). «Модернізація Цифрової системи управління нерухомим майном НАН України (ЦС “Майно”)», Звіт, номер держреєстрації 0124U002563 – К.: ІПС НАН України, 2024, 90 с. https://doi.org/10.5281/zenodo.14931816.

References

1. A.V. Chadyuk, R.M. Fedorenko, O.A. Kurchenko. Development and implementation of the management system of the economic and property complex of the National Academy of Sciences of Ukraine // *Problems in programming*, 2023, № 2, pp. 24-38, https://doi.org/10.15407/pp2023.02.024.
2. Fedorenko R.M., Shevchenko V.L., Ihnatenko P.P. and others (only 14 people). «Modernization of the Digital Real Estate Management System of the NAS of Ukraine (DS “Mayno”)», Report, state registration number 0124U002563 – K.: NAS of Ukraine, 2024, 90 pp. https://doi.org/10.5281/zenodo.14931816.

Отримано: 08.04.2025
Внутрішня рецензія отримана: 20.04.2025
Зовнішня рецензія отримана: 21.04.2025

Про авторів:

[1] *Федоренко Руслан Миколайович*,
кандидат економічних наук,
старший дослідник, завідувач
науково-дослідного відділу.
https://orcid.org/0000-0001-9433-5458

[1] *Бова Юрій Вікторович*,
аспірант, інженер-програміст.
https://orcid.org/0009-0008-9797-5213

[1] *Юрченко Костянтин Юрійович*,
аспірант,
молодший науковий співробітник.
https://orcid.org/0000-0003-3150-0027

[1] *Симоненко Олександр Вікторович,*
аспірант, інженер-програміст.
https://orcid.org/0009-0001-7443-9492

[1, 2] *Федоренко Іван Русланович*,
студент, інженер-програміст.
https://orcid.org/0009-0001-6339-8365

Місце роботи авторів:

[1] Інститут програмних систем
НАН України,
03187, м. Київ, проспект
Академіка Глушкова, 40, корпус 5.
E-mail: r_fedorenko@ukr.net,
bova1997@gmail.com,
urchikak8@gmail.com,
symonenko@iisd.com.ua,
ivan.fedorenko@ukr.net

[2] Національний технічний університет
України "Київський політехнічний
інститут імені Ігоря Сікорського",
E-mail: fedorenko.ivan@lll.kpi.ua

УДК 004.94 https://doi.org/10.15407/pp2025.02.112

Я. Жилюк, В. Плескач

ГІБРИДНИЙ ПІДХІД ДО ОБРОБЛЕННЯ НЕПОВНИХ ПОТОКОВИХ ДАНИХ У РОЗПОДІЛЕНИХ СИСТЕМАХ РЕАЛЬНОГО ЧАСУ

У статті розглянуто проблему оброблення неповних потокових даних у розподілених інформаційних системах реального часу, зокрема у контексті інтелектуального аналізу даних. Зазначено, що традиційні методи імпутації є малоефективними в умовах обмежених ресурсів, високих вимог до швидкості оброблення та динамічного характеру потоків. Запропоновано гібридний підхід, що поєднує *федеративне* навчання, контекстну імпутацію та адаптацію до концептуального дрейфу. Метод дозволяє локальним розподіленим обчислювальним вузлам тренувати полегшені моделі імпутації на власних даних із подальшою централізованою агрегацією, зворотним поширенням глобальної моделі та її динамічним оновленням. Експериментальна перевірка на реальному датасеті показала переваги підходу за точністю (RMSE, MAE) та навантаженням на мережу порівняно з базовими методами. Отримані результати засвідчують ефективність запропонованого методу в умовах розподілених середовищ з обмеженими обчислювальними ресурсами.

Y. Zhyliuk, V. Pleskach

HYBRID APPROACH TO PROCESSING INCOMPLETE STREAM DATA IN DISTRIBUTED REAL-TIME SYSTEMS

The article considers the problem of processing incomplete streaming data in distributed real-time systems, in particular in the context of data mining. It is noted that traditional methods of imputation are ineffective in conditions of limited resources, high requirements for processing speed and dynamic nature of streams. A hybrid approach combining federated learning, contextual imputation and adaptation to conceptual drift is proposed. The method allows local distributed computing nodes to train lightweight imputation models on their own data, followed by centralised aggregation, backpropagation of the global model and its dynamic updating. Experimental verification on a real dataset has shown the advantages of the approach in terms of accuracy (RMSE, MAE) and network load compared to the baseline methods. The obtained results prove the effectiveness of the proposed method in distributed environments with limited computing resources.

Вступ

З поширенням розподілених систем реального часу (РСРЧ) зростає рівень впровадження інтелектуального аналізу потокових даних у багатьох доменах, зокрема, Інтернеті речей (IoT), сенсорних мережах та фінансових ринках.

Потоки даних вирізняються своєю безперервною та необмеженою природою, чутливістю до часу та часто великим обсягом, що створює проблеми під час керування та оброблення даних [1].

Розподіл джерел даних ще більше ускладнює ці виклики, створюючи складнощі з боку синхронізації даних, інтеграції до загальної інфраструктури розподілених інформаційних систем. Значною перешкодою для ефективного використання інформації, що міститься в цих розподілених потоках даних, є поширене явище неповних або відсутніх даних. Ця проблема виникає внаслідок різних факторів, таких як несправності обладнання, помилки мережі та непередбачуваність процесів збору даних.

Причини неповноти даних можуть варіюватися від недостатньої повноти збору даних до неузгодженості в методологіях, неточностей у вимірюваннях, проблем із достовірністю чи цілісністю даних або навіть розбіжностей у часових межах збору даних. Крім того, відсутність даних може бути зумовлена різними механізмами, зокрема, повністю випадковою відсутністю

(*M*issing *C*ompletely *A*t *R*andom, MCAR), випадковою відсутністю (*M*issing *A*t *R*andom, MAR), невипадковою відсутністю (*M*issing *N*ot *A*t *R*andom, MNAR) та структурною відсутністю даних, кожен з яких має свої наслідки для функціонування системи й процесу аналізу потокових даних [2].

Традиційні підходи до імпутації даних орієнтовані здебільшого на статичні вибірки, виявляються малоефективними в умовах розподілених потокових систем реального часу. Методи імпутації – це прийоми, які використовують для заповнення відсутніх або пропущених значень у даних на основі контексту, для забезпечення їхньої повноти для подальшого аналізу чи побудови моделей.

Їхня неспроможність адаптуватися до динамічної природи потокових даних зумовлює проблеми масштабування, а також затримки під час обробки в розподілених обчислювальних середовищах. У зв'язку з цим виникає потреба у розробці нових методів імпутації, здатних враховувати специфіку розподілених потоків даних і забезпечувати ефективність інтелектуального аналізу в умовах жорстких часових обмежень.

Метою цієї статті є розроблення та обґрунтування гібридного підходу до імпутації неповних потокових даних у розподілених інформаційних системах реального часу, що враховує динамічний характер даних, обмежені обчислювальні ресурси та потребу в адаптації до концептуального дрейфу для підвищення ефективності інтелектуального аналізу даних. Зростаюча залежність від розподілених систем для збору й аналізу даних підкреслює критичність вирішення проблем із відсутніми даними в цих потоках, щоб розкрити їхній повний потенціал для отримання значущої інформації в процесі інтелектуального аналізу.

Проблема неповних даних у розподілених потоках для інтелектуального аналізу

Наявність неповних даних у розподілених потоках значно перешкоджає ефективності задач інтелектуального аналізу. Ці проблеми проявляються по-різному в розпізнаванні образів, виявленні аномалій і прогнозному моделюванні, кожне з яких вимагає ретельного розгляду та індивідуальних рішень.

Алгоритми, призначені для ідентифікації кластерів або класифікації даних, значною мірою покладаються на повний набір атрибутів, для точного визначення подібності та відмінності аналізованих даних. У разі відсутності певної частини інформації, вказані алгоритми можуть менш точно ефективно виконувати задачі класифікації, що призводить до зниження здатності розпізнавати значущі патерни. Наприклад, у мережах датчиків, розгорнутих у географічній зоні, відсутність показань від певних датчиків може призвести до неповних просторових або часових моделей, що ускладнює розуміння відстежуваного явища. Проблема відсутніх даних може посилюватися в поєднанні з незбалансованими наборами даних, де недостатнє представлення певних класів об'єктів може додатково посилюватися неповнотою інформації, що призводить до неправильної інтерпретації даних моделями машинного навчання [3].

Відсутність даних може призвести до виявлення помилкових закономірностей або упущення справжніх закономірностей, що зрештою призведе до некоректного розуміння основних явищ.

Виявлення аномалій у даних під час інтелектуального аналізу потоків у РСРЧ значно ускладнюється неповнотою вхідних даних. Аномальні точки даних, які представляють відхилення від норми, можуть бути замасковані чи неправильно інтерпретовані як відсутні значення, або, навпаки, самі відсутні значення можуть бути позначені як аномалії.

Ігнорування відсутніх значень є поширеною стратегією попереднього оброблення даних, що становить ризик згладжування або спотворення справжніх аномалій, тим самим нівелює саму мету виявлення відхилень. У розподілених середовищах, де дані можуть надходити асинхронно та з різним ступенем повноти з різних вузлів моніторингу, завдання виявлення аномалій стає ще більш складним. Наприклад, під час моніторингу мережевого трафіку відсутня

інформація про пакети може або приховати шкідливу активність, яка виглядатиме як аномалія, якщо вона завершена, або хибно передбачити вторгнення через неповну передачу [4].

Прогнозування в контексті неповних потоків даних також створює значні перешкоди. Відсутні значення в навчальних даних, які використовуються для побудови прогнозних моделей, можуть призвести до упереджених або неефективних моделей, що зрештою вплине на їхню точність і надійність. Коли ці моделі застосовують до потоків в РСРЧ, які також містять відсутні дані, точність отриманих прогнозів може бути суттєво знижена. Проблема ще більше ускладнюється явищем зсуву інформації, яке є поширеним у потокових даних, де основний розподіл даних змінюється з часом [11].

Прогнозні моделі, базовані на навчанні на історичних даних із пропущеними значеннями, можуть виявитися недостатньо адаптованими до змін у поточному потоці даних, що може призвести до поступового зниження точності прогнозів під час розвитку процесу. Наприклад, у закладах охорони здоров'я система моніторингу показників життєдіяльності пацієнтів у реальному часі, яка спирається на прогнозні моделі, може генерувати неточні прогнози критичних подій, якщо у вхідному потоці даних відсутні важливі показники. Прогнозні моделі, навчені на неповних даних, можуть вивчити помилкові зв'язки між даними або не зафіксувати базову динаміку, що призведе до ненадійного узагальнення та неточних прогнозів на нових, потенційно також неповних даних. Відсутні значення в навчальних даних зменшують обсяг інформації, доступної для навчання, потенційно призводячи до надмірного, недостатнього підбору чи зміщення параметрів моделі.

Обмеження традиційних методів доповнення даних для розподілених потоків у реальному часі

Традиційні методи доповнення даних, хоча й ефективні в певних контекстах, часто виявляються не ефективними для оброблення відсутніх даних у потоках розподілених даних у режимі реального часу, особливо в умовах застосування методів інтелектуального аналізу, що пояснюється кількома факторами, зокрема, затримкою, обчислювальними обмеженнями та мінливим характером розподілу даних.

Одним з основних обмежень є затримка. Багато традиційних методів доповнення, такі як множинна імпутація та складні підходи на основі машинного навчання, є обчислювально ресурсомісткими та можуть призводити до значних затримок в обробленні. Оброблення потоків у реальному часі за своєю природою вимагає доповнення даних з низькою затримкою, щоб забезпечити своєчасний аналіз та ухвалення рішень. Навіть регресійна імпутація, яка може бути відносно точною, може призводити до неприйнятної затримки, особливо якщо базові регресійні моделі є складними або потребують частого оновлення. Потреба в негайному обробленні потоків даних суперечить притаманній затримці багатьох точних традиційних методів імпутації, змушуючи йти на компроміс між точністю та швидкістю. Застосунки реального часу вимагають швидкого аналізу. Методи імпутації, які потребують значного часу оброблення на централізованому вузлі, стають вузькими місцями в розподіленому потоковому середовищі, затримуючи загальний аналіз і потенційно зменшуючи релевантність результатів аналізу.

Обчислювальні обмеження також створюють значну проблему. Розподілене оброблення потоків часто відбувається на пристроях з обмеженими ресурсами, таких як датчики Інтернету речей та периферійні сервери. Багато складних методів імпутації вимагають значної обчислювальної потужності та ресурсів пам'яті, які можуть перевищувати можливості цих пристроїв. Для ефективного завершення оброблення даних у таких середовищах необхідні легкі та ефективні методи імпутації. Розгортання обчислювально ресурсомістких методів імпутації на периферії розподіленої системи обробки потоків часто є неможливим через обмежені ресурси, доступні на цих пристроях. Потоки даних часто надходять із численних

пристроїв з низьким енергоспоживанням. Алгоритми імпутації мають бути достатньо ефективними, щоб функціонувати локально або на периферії, не споживаючи надмірної енергії або обчислювальної потужності, що може вплинути на основну функцію та строк служби пристрою [6].

Традиційні методи доповнення часто припускають, що основні статистичні властивості даних залишаються постійними з часом, однак потоки даних є за своєю суттю динамічними, а їхні розподіли змінюються, коли виникають нові закономірності, а старі зникають. Моделі імпутації, навчені на знімку історичних даних, можуть з часом ставати неточними у процесі розвитку потоку даних, що призводить до низької продуктивності у врахуванні відсутніх значень. Для вирішення цієї проблеми потрібні адаптивні методи імпутації, які можуть виявляти та коригуватися до концептуального дрейфу даних [11]. Дрейф даних (data drift) – це зміна статистичних властивостей даних, які використовують у моделях машинного навчання, протягом певного часу. Це явище може призвести до зниження ефективності моделі, оскільки алгоритм, навчений на певних даних, може стати менш точним або некоректним, якщо дані, на яких вона працює, змінюють. Динамічна природа потоків даних з їхніми концептуальними закономірностями та розподілами, що розвиваються, вимагає методів імпутації, спроможних адаптуватися в режимі реального часу для підтримки точності [7].

Зрештою традиційні методи доповнення несуть потенціал для внесення систематичних помилок у дані. Прості методи, такі як імпутація середнього значення, хоча й легко реалізувати, можуть спотворювати розподіл основних даних і недооцінювати дисперсію, що потенційно може призвести до хибно позитивних висновків у подальшому аналізі. Регресійна імпутація, якщо обрана модель регресії неточно відображає справжні зв'язки між змінними, також може призвести до систематичної помилки. Ефективність будь-якого методу імпутації тісно пов'язана з механізмом, через який дані відсутні (MCAR, MAR, MNAR), а ігнорування цього механізму може призвести до систематичних помилок імпутації. Систематична помилка, внесена під час процесу доповнення, може потім поширюватися та негативно впливати на результати подальших завдань інтелектуального аналізу. Неправильно застосовані методи доповнення можуть вносити систематичні помилки в дані, що потенційно може призвести до помилкових висновків та рішень на основі подальшого інтелектуального аналізу. Мета імпутації – заповнити відсутні значення правдоподібними оцінками. Однак, якщо метод імпутації робить неправильні припущення щодо даних або механізму відсутності, це може спотворити справжні основні зв'язки між даними, які поширюються по всьому ланцюгу інтелектуального аналізу [7].

Визнаючи обмеження традиційних методів імпутації в контексті потоків даних, дослідники розробили різні методології, спеціально спрямовані на вирішення проблем, що виникають через неповні дані в цих динамічних середовищах. Ці підходи варіюються від простих методів видалення до більш складних методів імпутації, адаптованих до унікальних характеристик потокових даних.

Методи видалення представляють собою простий підхід до обробки відсутніх даних. Спискове видалення передбачає видалення всіх спостережень з набору даних, якщо вони містять одне чи декілька відсутніх значень. Хоча цей метод простий у реалізації, він може призвести до значної втрати цінної інформації, особливо коли рівень відсутніх даних високий. Більше того, якщо відсутні дані не є повністю випадковими (MCAR), спискове видалення може внести зміщення в подальший аналіз. З іншого боку, попарне видалення намагається зменшити втрату інформації, використовуючи всі доступні дані для кожного конкретного аналізу. Це означає, що аналіз може базуватися на різних підмножинах даних, залежно від того, які змінні мають відсутні значення. Хоча попарне видалення зберігає більше даних, ніж спискове, воно також може призвести до проблем, таких як матриці інтеркореляції, які не є позитивно ви-

значеними, що потенційно перешкоджає подальшому аналізу.

Статистична імпутація для потоків передбачає адаптацію традиційних статистичних методів, таких як імпутація середнього або медіанного значення до контексту потоку, часто за допомогою ковзних вікон. У ковзному вікні останніх точок даних обчислюють середнє значення або медіану спостережуваних значень для певного атрибута та використовують для імпутації будь-яких відсутніх значень у цьому вікні. Цей підхід є обчислювально простим і може бути ефективним, коли механізмом відсутніх даних є MCAR або MAR, а розподіл даних у вікні є відносно стабільним [7].

Імпутація потоків на основі машинного навчання набула поширеності завдяки своїй здатності фіксувати складніші взаємозв'язки та адаптуватися до змін у розподілі даних. Алгоритми онлайн-навчання, такі як онлайн-градієнтний спуск та адаптивна фільтрація, здатні постійно оновлювати моделі імпутації в процесі надходження нових даних, що дозволяє їм ефективно адаптуватися до концептуального дрейфу в режимі реального часу. Рекурентні нейронні мережі (RNN), зокрема LSTM, були також предметом досліджень завдяки своїй здатності моделювати часові залежності в потокових даних, що забезпечує точне імпутування відсутніх значень [8].

У розподілених середовищах методи розподіленого доповнення спрямовано на обролення відсутніх даних без необхідності об'єднання всіх даних у централізоване сховище, що може бути нездійсненним через проблеми конфіденційності чи обмеження мережі. Такі методи, як ефективна для комунікації розподілена множинна імпутація, були розроблені для горизонтально розподілених даних, де дані з різних джерел або сайтів містять однакові атрибути, але для різних суб'єктів.

Інші спеціалізовані підходи охоплюють імпутацію на основі кореляції, яка використовує зв'язки між різними потоками даних або датчиками для визначення відсутніх значень в одному потоці на основі спостережуваних значень у корельованих потоках. Адаптивна імпутація на основі нечіткої логіки використовує нечітку логіку для управління невизначеністю, пов'язаною з відсутніми значеннями, та адаптації процесу імпутації на основі характеристик незбалансованих потоків даних. Методи доповнення на основі графів будують графіки подібності екземплярів даних у певному часовому вікні, а потім використовують методи поширення повідомлень на цих графіках для використання кореляцій між екземплярами та імпутації відсутніх значень [9].

Наявні методології оброблення відсутніх даних у потоках демонструють чітку еволюцію від простих статистичних методів до більш складних підходів машинного навчання та розподілених обчислень. Спільною характеристикою багатьох досліджень із зазначеним питань є визнання часової та розподіленої природи даних, а також потреба в методах, які можуть адаптуватися до змін у розподілі базових даних.

Гібридний підхід до доповнення потокових даних у розподілених системах

Щоб усунути обмеження наявних методів оброблення неповних даних у розподілених потоках для інтелектуального аналізу, пропонується новий підхід, який використовує розподілену природу даних, охоплює контекстну інформацію, адаптується до динамічних характеристик і підтримує обчислювальну ефективність. Цей метод спирається на принципи спільного навчання, контекстно-залежного моделювання та адаптації концептуального дрейфу.

Запропонований підхід може використовувати спільне навчання або федеративні навчальні фреймворки, де кілька розподілених вузлів сприяють побудові глобальної моделі імпутації без необхідності обміну необробленими, потенційно конфіденційними даними.

Федеративне навчання – це метод машинного навчання, за якого моделі навчаються на розподілених даних, що зберігаються на пристроях або серверах, без необхідності централізованого збору чи передачі цих даних. Замість того, щоб передавати всі дані на сервер для навчання, лише параме-

три чи оновлення моделі обмінюються між пристроями та сервером, що дозволяє зберегти конфіденційність даних, оскільки вони не залишають своїх локальних джерел. У цій парадигмі кожен розподілений вузол навчатиме локальну модель імпутації, використовуючи власний потік даних. Потім центральний сервер агрегуватиме ці локально навчені моделі, потенційно використовуючи такі методи, як федеративне усереднення, для створення більш надійної та узагальненої глобальної моделі імпутації. Ця стратегія не лише вирішує проблеми конфіденційності, зберігаючи необроблені дані децентралізованими, а й використовує колективні знання, вбудовані в дані з різних розподілених джерел [10].

Крім того, запропонований метод має зосереджуватися на включенні контекстуальної інформації, щодо потоків даних. Це може охоплювати врахування часових залежностей у кожному потоці, просторових зв'язків між точками даних (якщо дані мають просторовий компонент, наприклад, у сенсорних мережах) та інших відповідних контекстуальних факторів. Приміром, у випадку даних датчиків, процес імпутації може враховувати показники сусідніх датчиків у певні проміжки часу чи історичні закономірності того ж датчика у конкретних умовах.

Враховуючи динамічну природу потоків даних, запропонований метод також має охоплювати механізми адаптації до динамічних характеристик, зокрема, концептуального дрейфу. Це може охоплювати постійний моніторинг продуктивності глобальної моделі імпутації вхідних потоків даних на кожному розподіленому вузлі. Якщо виявлено значний дрейф у розподілі даних, система може ініціювати перенавчання або оновлення локальних моделей та подальшу повторну агрегацію на центральному сервері. Для підвищення здатності моделі адаптуватися до змінних шаблонів даних можна використовувати такі методи, як рання зупинка на боці клієнта під час локального навчання або адаптивна оптимізація з урахуванням дрейфу на боці сервера під час агрегації.

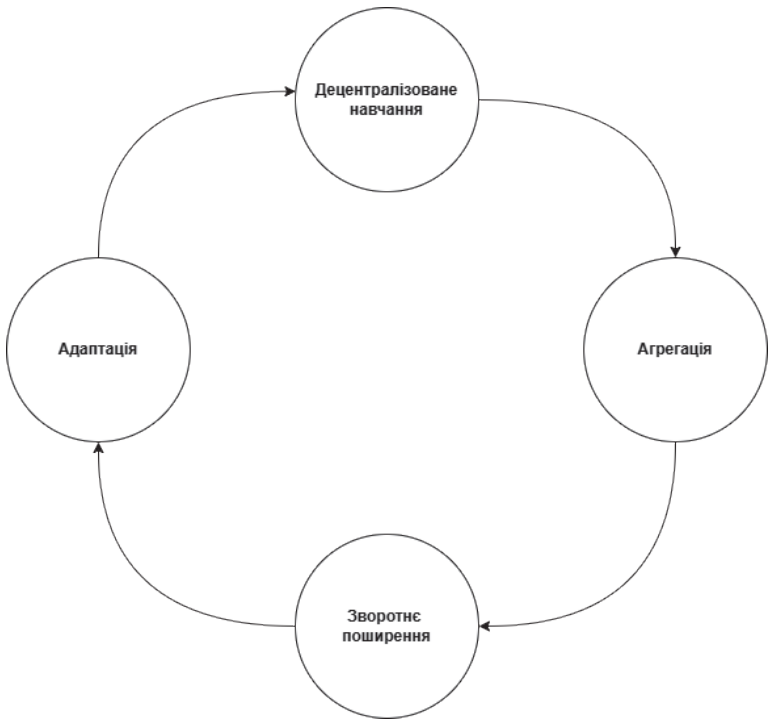


Рис. 1. Етапи пропонованого гібридного підходу до оброблення неповних потокових даних

Для забезпечення доцільності використання гібридного підходу у розподілених середовищах з обмеженими ресурсами, локальні моделі імпутації, навчені на окремих вузлах, мають бути обчислювально ефективними. Такі методи, як дистиляція

знань, коли знання складної моделі переносяться на меншу, ефективнішу модель, можна використовувати для створення моделей імпутації, які здатні ефективно працювати на периферійних пристроях без надмірного споживання ресурсів.

Пропонований гібридний підхід до оброблення неповних потокових даних у розподілених системах реального часу складається з чотирьох етапів та може бути поданий як циклічний процес (рис. 1).

На першому етапі децентралізованого навчання кожен розподілений вузол ініціалізує та навчає спрощену локальну модель імпутації на своєму конкретному потоці даних, зокрема, відповідну контекстуальну інформацію. На етапі агрегації параметрів центральний сервер керує процесом федеративного навчання, де він періодично агрегує параметри й оновлення з локальних моделей імпутації, навчених на кожному вузлі. Іншим етапом є зворотнє поширення, на якому отримана глобальна модель імпутації набуває необхідних характеристик, використовуючи переваги колективного навчання на всіх вузлах, а потім розподіляється у звортньому напрямку на окремі вузли. Кожен вузол використовує глобальну модель для імпутування відсутніх значень у свій вхідний потік даних у режимі реального часу. Цю глобальну модель можна потенційно додатково налаштувати, використовуючи локальні дані на кожному вузлі, щоб врахувати будь-які унікальні локальні характеристики. Останнім етапом є процес адаптації, що виконується періодично, під час якого система постійно контролює продуктивність процесу імпутації на кожному вузлі на наявність ознак концептуального дрейфу. Виявивши значний дрейф, система ініціює новий раунд локального навчання та глобальної агрегації для оновлення моделей імпутації.

Математично можемо визначити підхід таким чином: нехай $N = \{ N_1, N_2 \dots N_n \}$ – множина розподілених вузлів, $D_t^i = (x_j^t, m_j^t)$ – набір даних на вузлі N_i в момент часу *t,* де $x_j^t \in \mathbb{R}$ – *j*-та точка даних у момент часу t, $m_j^t \in \{0, 1\}$ – вектор бінарної маски відсутності, f_i^t – локальна модель імпутації на вузлі N_i в час *t*, $\theta_i^t -$ параметри локальної моделі f_i^t, тоді для кожного етапу можемо записати наступні твердження.

Кожен вузол знаходить вектор відсутніх даних:

$$\hat{x}_j^t = f_i^t \left(x_j^t, m_j^t, c_j^t\right)$$

де c_j^t – є контекстною інформацією (наприклад, залежності часових рядів, показники сусідніх датчиків, семантичні вбудовування для табличних даних). Локальна модель навчена мінімізувати втрати від маскованої реконструкції:

$$L_i^t = \sum_{j=1}^{D_t^i} |(1 - m_j^t) \odot (\hat{x}_j^t - x_j^t)|^2$$

Через задані проміжки часу вузли надсилають оновлення параметрів моделі θ_i^t на центральний сервер:

$$\theta^{t+1} = \sum_{i=1}^{n} \frac{w_i^t}{\sum_k w_k^t} \theta_i^t$$

де w_i^t – ваговий коефіцієнт, наприклад, пропорційний кількості зразків або оцінці продуктивності.

На етапі адаптації кожен вузол оцінює достовірність прогнозування або розподіл помилок з плином часу. Якщо виявляється дрейф (наприклад, за допомогою методу виявлення змін, такого як Пейдж-Хінклі), це запускає локальне перенавчання. За методом Пейдж-Хінклі, нехай ε_i^t – середнє ковзне помилки імпутації, тоді при:

$$\left|\varepsilon_i^t - \varepsilon_i^{t-k}\right| > \delta$$

де k – кількість кроків зворотнього порівняння, δ – межа чутливості помилки, вузол перенавчає свою модель та починає новий раунд федеративних оновлень [12].

Моделювання роботи гібридного підходу

Для оцінки нового підходу проведемо моделювання роботи запропонова-

ного алгоритму з використанням засобів мови програмування Golang. Порівняльну характеристику метрик щодо моделювання роботи методів імпутації подано у табл.1.

Таблиця 1

Порівняльна характеристика метрик щодо моделювання роботи методів імпутації

Метод	RMSE	MAE	Bandwith load
Гібридний підхід	6.2	3.8	21.5 мб
Mean Imputation	12.5	9.1	-
Deep Autoencoder	7.1	5.2	1974 мб

Як тестовий набір даних було використано набір «ElectricityLoadDiagrams20112014» (https://archive.ics.uci.edu/dataset/321/electricityloaddiagrams20112014), що репрезентує енергоспоживання 370 домогосподарств з оновленням даних один раз на 15 хвилин протягом 2011-2014 років. Цей датасет моделює мережу з розподілених датчиків, що надсилають дані на централізований вузол оброблення. І на прикладі сезонності даних – зростання або спадання споживання електроенергії залежно від пори року можна протестувати роботу запропонованого методу з концептуальним дрейфом.

Для моделювання дані розподіляють між змодельованими вузлами, водночас 20 значень випадковим чином видаляють за допомогою схем MAR і MNAR. Для порівняння використаємо методи доповнення середнім (Mean Imputation) і метод глибокого автоенкодера (Deep Autoencoder). Оцінку ефективності проведемо за допомогою метрики середньоквадратичного відхилення (RMSE) та середнього абсолютного відхилення (MAE):

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2};$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|x_i - \hat{x}_i|$$

Додатково обчислимо загальну кількість даних обміну між вузлами під час роботи для визначення навантаження на мережу (Bandwidth load). Для навчання моделей використаємо бібліотеку golearn. Модель розподіленої системи складається з 370 вузлів.

Відповідно до отриманих результатів моделювання (табл. 1) запропоновано гібридний підхід до доповнення даних, що показав кращий результат за метрикою RMSE порівняно з методом Deep Autoencoder на 13%, водночас зменшивши середньоквадратичне відхилення порівняно з методом Mean Imputation удвічі. За метрикою MAE приріст точності порівняно з Deep Autoencoder склав 27%.

Водночас гібридний підхід значно зменшив навантаження на мережу з 1974 мегабайт даних до 21.5 мегабайт порівняно з методом Deep Autoencoder. Це пояснюється зниженням розміру пакетів, що передаються між вузлами через тип даних. Якщо Deep Autoencoder вимагає передачу сирих даних з вузлів для обчислення відсутніх даних, то в пропонованому підході між вузлами передаються лише параметри локальної і агрегованої моделі, що значно впливає на обсяг споживання мережевого трафіку та пропускної здатності мережі.

Запропонований метод, хоча й перспективний, однак має потенційні обмеження. Накладні видатки на зв'язок, пов'язані з процесом федеративного навчання, потребують ретельного керування, особливо в середовищах з обмеженою пропускною здатністю або високою затримкою. Оброблення значної неоднорідності в розподілі даних між різними вузлами також може створювати проблеми, вимагаючи складних методів агрегації. Складність точного виявлення та адаптації до різних типів концептуального дрейфу в розподіленому середовищі вимагає подальшого дослідження. Крім того, є компроміс між складністю та точністю моделей імпутації та їхньою обчислювальною ефективністю, який необхідно ретельно збалансувати на основі вимог конкретного застосування. Водночас оцінка ефективності методу ім-

путації в контексті конкретних задач інтелектуального аналізу, що виконуються над доповненими потоками даних, має вирішальне значення для підтвердження його ефективності.

Висновки

Проблеми, що виникають через неповні дані в розподілених потоках для інтелектуального аналізу, є значними та поширеними в різних прикладних галузях. Традиційні методи імпутації даних, часто розроблені для статичних наборів даних, намагаються ефективно враховувати динамічний характер, обчислювальні обмеження та вимоги до реального часу цих потокових середовищ.

Запропонований метод імпутації на основі федеративного навчання визначає перспективний напрямок, дозволяючи розподіленим вузлам спільно створювати глобальну модель імпутації. Метод вирішує проблеми конфіденційності та використовує колективний інтелект. Включення контекстної інформації та постійна адаптація до концептуального дрейфу спрямовані на підвищення точності та надійності імпутованих даних з часом.

Хоча вищезгаданий метод пропонує кілька потенційних переваг, такі обмеження, як накладні витрати на зв'язок, обробка неоднорідності даних та складність виявлення та адаптації дрейфу, потребують ретельного розгляду та врахування в майбутніх дослідженнях.

Зазначимо, що ефективні методи доповнення даних мають першочергове значення для розкриття повного потенціалу інтелектуального аналізу потокових даних у РСРЧ. Зі зростанням обсягу та швидкості цих потоків потреба в інноваційних та адаптивних рішеннях для оброблення неповних даних ставатиме критичнішою. Запропонований спільний, контекстно-залежний та адаптивний метод імпутації є позитивним кроком до вирішення цих проблем і забезпечення надійнішого та ефективнішого аналізу великих обсягів даних, що генеруються в розподілених середовищах.

References

1. Handling missing values in data streams: An overview. Afonso Lima, Elaine P. M. de Sousa, 2024.
2. Distributed Data and Immersive Collaboration. Daniel Reed, Roscoe Giles, Charles E. Catlett, 1997.
3. Missing Data Imputation: A Comprehensive Review. Journal of Computer and Communications. Alwateer, M. , Atlam, E. , El-Raouf, M. , Ghoneim, O. and Gad, I., 2024.
4. A Comprehensive Review of Handling Missing Data: Exploring Special Missing Mechanisms. Youran Zhou, Sunil Aryal, 2024.
5. REAL-TIME SYSTEMS Design Principles for Distributed Embedded Applications. Hermann Kopetz, 1997.
6. Efficient Join Processing Over Incomplete Data Streams (Technical Report). Weilong Ren, Xiang Lian, Kambiz Ghazinour, 2019.
7. Emerging Issues in Data Storytelling CHAPTER 1 | The Challenges of Working With Incomplete Data Sets [Online] – Available from: https://www.icom.org/publications/data-storytelling/the-challenges-of-working-with-incomplete-data-sets (Accessed 28.04.2025).
8. Analyzing Incomplete Political Science Data: An Alternative Algorithm for Multiple Imputation. GARY KING, JAMES HONAKER, ANNE JOSEPH, KENNETH SCHEVE, 2001.
9. Chukwuemeka Obasi, Victor Oisamoje, Braimoh Ikharo. Security in Distributed System: A Review Perspective, 2022.
10. Time-Sensitive Networking [Online] – Available from: https://campaign.advantech.online/en/global/solutions/intelligent-transportation-systems/resources/white-papers/Time-Sensitive-Networking.pdf , (Accessed 28.04.2025).
11. Concept Drift [Online] – Available from: https://www.iguazio.com/glossary/concept-drift/ , (Accessed 28.04.2025).
12. Continuous Inspection Schemes. Biometrika 41. E. S. Page. 1954.

Одержано: 01.05.2025
Внутрішня рецензія отримана:17.05.2025
Зовнішня рецензія отримана: 18.05.2025

Про авторів :

Плескач Валентина,
д.е.н., к.т.н., професор
https://orcid.org/0000-0003-0552-0972

Жилюк Ярослав,
аспірант
https://orcid.org/0009-0008-9341-9164

Місце роботи авторів:

Київський національний університет
імені Т.Г. Шевченка, факультет
інформаційних технологій
v.pleskach64@gmail.com

ВИМОГИ ДО ОФОРМЛЕННЯ СТАТЕЙ

1. Загальні положення

У журналі "Проблеми програмування" публікуються наукові матеріали, які раніше не публікувалися в інших виданнях.

Мова статті: українська, англійська. 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.

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.

Для передачі до редакції тексту статті, ділової переписки та правки при коректурі автори користуються електронною поштою редакції: alengoro@isofts.kiev.ua, alengoro2022@gmail.com. Для надійності інформаційного обміну в умовах можливих відключень електрики – прохання надсилати матеріали на обидві електронні пошти одночасно. Телефон: +380 (96) 418 3082.

2. 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.

3. Послідовність розміщення та оформлення матеріалу статті

1. ***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.

2. ***Нижній колонтитул*** (тільки для першої сторінки) включає стандартну інформацію Copyright: перший рядок – прізвища авторів, рік; другий рядок – номер ISSN, назва журналу, рік, номер випуску.
3. **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.

4. **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, $F_f=(F_\lambda, F_\varphi, F_z)$.

5. **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.
6. **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].

7. **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).

Застосований у статті маркований список (наведений вище) має наступні параметри: маркер має відступ 1,25 см, текст для першого рядка – 1,9 см. Аналогічні відступи слід підтримувати і для нумерованого списку.

8. **Література**: єдиний нумерований список джерел згідно ДСТУ 8302:2015 від 01.07.2016 р., шрифт 11 пт, відступ: спеціальний, навислий, 0,63 см.
9. **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. У випадках, коли список джерел включає джерела тільки однією мовою, він подається один раз.

10. ***Дата надходження статті*** позначається редакцією цифрами окремим рядком після слова «Одержано:»/”Received:”.

11. ***Дата надходження внутрішньої рецензії*** позначається редакцією цифрами окремим рядком після слів «Внутрішня рецензія отримана»/”Internal review received:”.

12. ***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.

13. ***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).

14. ***Дані про місце роботи авторів*****:** починаються рядком “*Місце роботи авторів:*”, напівжирний курсив. Далі вказуються місце роботи, телефон, електронна пошта, веб-сайт.

15. 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.

    Для полегшення підготовки статей, що задовольняють вищенаведеним вимогам, редакція журналу розробила файл шаблону статті “shablon.dot”, який можуть використовувати автори.