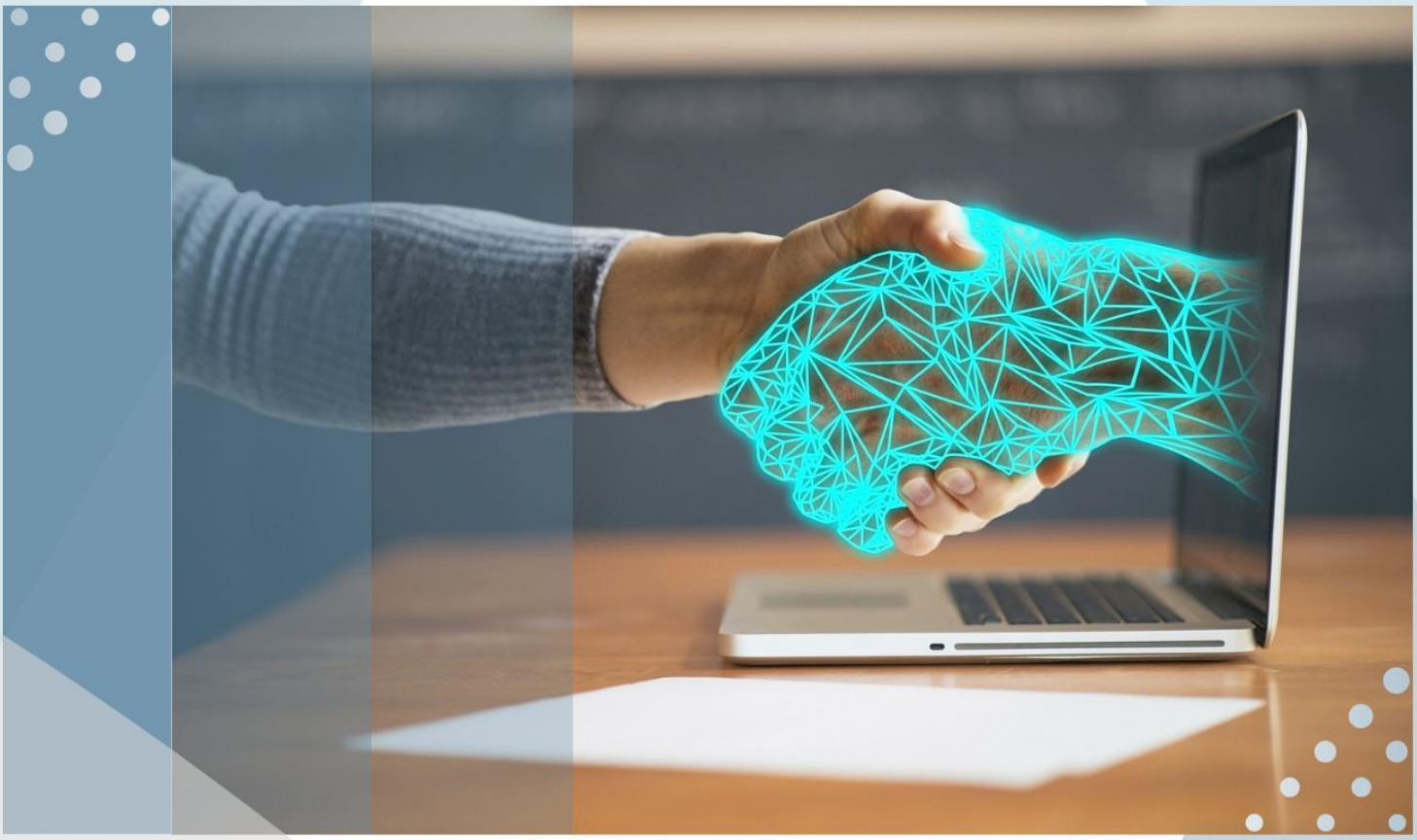
ITEES

Information technologies in economics and environmental sciences



No 1 / 2025

NUBIP of Ukraine

nubip.edu.ua

Information Technologies in Economics and Environmental Sciences

Founder:

National University of Life and Environmental Sciences of Ukraine

Year of foundation: 2017

Published 2 times a year

Media identifier – R40-02287

The National Council of Television and Radio Broadcasting of Ukraine
Decision No. 2543 of 20 November 2025, protocol No. 25.

Editors office address:

National University of Life and Environmental Sciences of Ukraine

03041, 15 Heroiv Oborony Str., Kyiv, Ukraine

E-mail: it-journal@nubip.edu.ua

https://journals.nubip.edu.ua/index.php/Inf/en/

Інформаційні технології в економіці та природокористуванні

Засновник журналу:

Національний університет біоресурсів і природокористування України

Факультет інформаційних технологій

Рік заснування: 2017

Виходить 2 рази на рік

Ідентифікатор медіа – R40-06595

Рішення Національної Ради України з питань телебачення і радіомовлення від 20.11.2025 р. № 2543, протокол № 25

Адреса редакції:

Національний університет біоресурсів і природокористування України

03041, вул. Героїв Оборони, 15, м. Київ, Україна

E-mail: it-journal@nubip.edu.ua

https://journals.nubip.edu.ua/index.php/Inf/uk

EDITORIAL BOARD

РЕДАКЦІЙНА КОЛЕГІЯ

CONTENTS

ЗМІСТ

DOI:10.31548/itees.2025.01.009
ISSN 3083-7502

УДК 004.89:519.85

Вайганг Ганна Олександрівна
кандидат технічних наук, доцент, доцент кафедри комп'ютерних наук,
Національний університет біоресурсів і природокористування України
ORCID: https://orcid.org/0000-0002-2082-2322
E-mail: weigang.ganna@nubip.edu.ua

Науринський Юрій Володимирович
асистент кафедри комп'ютерних наук,
Національний університет біоресурсів і природокористування України
ORCID: https://orcid.org/0009-0004-6416-8635
E-mail: yu.naurynskyi@nubip.edu.ua

Мирончук Катерина Вячеславівна
старший викладач кафедри комп'ютерних систем, мереж та кібербезпеки,
Національний університет біоресурсів і природокористування України
ORCID: https://orcid.org/0000-0001-6764-3746
E-mail: k.komar@nubip.edu.ua

ГІБРИДНА МОДЕЛЬ ОПТИМІЗАЦІЇ КЛАСТЕРИЗАЦІЇ ДЛЯ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

Анотація. *У статті представлено комплексний підхід до оптимізації алгоритмів кластеризації в системах підтримки прийняття рішень (СППР) у середовищі великих даних. Проведено аналіз проблем масштабованості, обчислювальної складності та стабільності результатів, характерних для класичних методів K-Means, DBSCAN і Agglomerative Clustering. Запропоновано удосконалений гібридний алгоритм K-Means++ Hybrid, який поєднує механізми паралельних обчислень, адаптивного налаштування параметрів і динамічного контролю ітераційного пошуку. Методологічна основа дослідження базується на системному аналізі, математичному моделюванні та експериментальному тестуванні із використанням наборів даних UCI Repository і технологій GPU-прискорення (CUDA). Експериментальні результати підтверджують, що застосування запропонованого підходу дозволяє знизити час виконання кластеризації в середньому на ≈43% порівняно з базовими алгоритмами, підвищивши при цьому коефіцієнт силуету до 0,73 та зменшивши енергоспоживання процесора на 20–25%. Отримана модель забезпечує високу стійкість при роботі з гетерогенними наборами даних і може бути інтегрована в системи аналізу транспортних потоків, оцінювання фінансових ризиків та моніторингу екологічних параметрів. Розроблений підхід формує основу для побудови адаптивних модулів інтелектуального аналізу даних, які підтримують масштабування, інтерпретованість результатів і реальну роботу в потокових аналітичних системах. Подальші дослідження доцільно зосередити на поєднанні гібридної кластеризації з моделями глибинного навчання та методами оптимізації на базі еволюційних алгоритмів.*

Ключові слова: *кластеризація, великі дані, системи підтримки прийняття рішень, оптимізація алгоритмів, паралельні обчислення, гібридні моделі, машинне навчання.*

Вступ. У сучасній економіці, науці та техніці спостерігається стрімке зростання обсягів інформації, яку необхідно аналізувати для прийняття ефективних рішень. Потоки даних, що надходять із сенсорних мереж, корпоративних інформаційних систем, соціальних медіа, вебсервісів чи інтелектуальних транспортних платформ, формують середовище, де традиційні аналітичні методи втрачають ефективність. Виникає потреба в адаптивних інструментах обробки, здатних не лише забезпечити швидкість, а й зберегти якість та достовірність аналітичних висновків. Одним із найбільш поширених напрямів вирішення цієї проблеми є використання методів кластеризації, які дозволяють структурувати великі масиви даних, виявляти приховані зв'язки та закономірності між об'єктами.

Разом із тим, класичні алгоритми кластеризації, зокрема K-Means, DBSCAN та ієрархічні підходи, мають низку обмежень, що проявляються при роботі з великими або гетерогенними

наборами даних. Їхня обчислювальна складність зростає нелінійно, а результати стають чутливими до вибору початкових умов та параметрів. У багатьох випадках це призводить до втрати стабільності результатів або неможливості застосування у режимі реального часу. Таким чином, питання підвищення продуктивності та масштабованості алгоритмів кластеризації постає як стратегічне для подальшого розвитку систем підтримки прийняття рішень (СППР).

Сучасні тенденції в цій галузі пов'язані з інтеграцією методів кластеризації в архітектури обчислень із високим рівнем паралелізму, зокрема із використанням GPU-технологій, розподілених систем обробки (Spark, Hadoop) та хмарних платформ. Це дозволяє здійснювати аналіз потокових даних у реальному часі, а також формувати динамічні моделі прийняття рішень. Паралельно з цим розвивається напрям гібридних алгоритмів, що поєднують кластеризацію з машинним навчанням, глибинними нейромережами або евристичними оптимізаційними підходами, такими як генетичні алгоритми чи метод рою частинок. Їх застосування підвищує здатність моделей адаптуватися до нестандартних структур даних і зменшує ризик локальних мінімумів.

Розвиток інтелектуальних систем підтримки прийняття рішень неможливий без забезпечення балансу між швидкодією, точністю та надійністю алгоритмічної обробки даних. Тому особливу увагу у сучасних дослідженнях приділяють пошуку компромісних підходів – таких, що поєднують обчислювальну ефективність паралельних технологій із математичною строгістю моделей кластеризації. Вдосконалення таких методів сприятиме створенню систем, здатних оперативно аналізувати інформаційні потоки, знижувати ризики прийняття помилкових рішень і підвищувати якість стратегічного управління у різних галузях – від транспорту й енергетики до фінансового сектору та цифрової економіки.

Огляд літератури. Аналіз сучасних досліджень свідчить, що методи кластеризації залишаються одним із ключових інструментів інтелектуального аналізу даних, особливо у контексті систем підтримки прийняття рішень. Розвиток цієї галузі зумовлений зростанням обсягів інформації та необхідністю автоматизованого групування об'єктів за багатовимірними ознаками. Традиційні алгоритми, зокрема K-Means, DBSCAN та Agglomerative Clustering, активно використовуються для класифікації даних різної природи [1; 2; 3]. Водночас, дослідження показують, що їх ефективність знижується при роботі з високорозмірними або нерівномірно розподіленими наборами даних, що зумовлює потребу в пошуку нових підходів до оптимізації процесу кластеризації [4; 5].

Згідно з дослідженням Alzubaidi L. та співавт. [6], головними викликами при кластеризації великих даних є обмеження масштабованості, надмірна обчислювальна складність і значна чутливість до шумів. Автори пропонують використання розподілених моделей кластеризації, які базуються на паралельній обробці даних, що дає змогу скоротити час обчислень та забезпечити стійкість результатів. Подібні висновки отримано в роботі Alramahee A. та Ghalib F. [2], де зазначено, що паралельні алгоритми кластеризації на базі багатоядерних процесорів і GPU здатні істотно підвищити продуктивність при обробці потокових даних. Подібні результати наведено в огляді Dafir Z. [7], який систематизує паралельні алгоритми кластеризації для великих даних та оцінює їх ефективність у хмарних і розподілених середовищах.

Розширення можливостей кластеризаційних методів також спостерігається завдяки інтеграції з технологіями машинного навчання. Зокрема, у дослідженнях Oyewole G. J. і Thopil G. A. [3] наголошується, що комбінація кластеризації з глибинними нейронними мережами дає змогу адаптувати процес до складних структур даних і забезпечити більш точне групування без попереднього визначення кількості кластерів. Подібний підхід розвинули Zhou S. та колеги [8], які систематизували методи глибокої кластеризації, визначивши перспективні напрямки розвитку – інтерпретованість результатів, автономне навчання та застосування у потокових аналітичних системах.

Окремий напрям формують дослідження, присвячені підвищенню інтерпретованості кластерних моделей. Так, Hu L. та ін. [9] запропонували підхід, що поєднує інтерпретовані

DOI:10.31548/itees.2025.01.009
ISSN 3083-7502

правила прийняття рішень із автоматичною кластеризацією, що підвищує прозорість процесів у СППР. Цей аспект особливо актуальний для галузей, де рішення мають високий рівень відповідальності – фінансів, медицини та кібербезпеки.

В українському науковому просторі увага дослідників зосереджена на практичному використанні кластеризації у різних прикладних контекстах. Бойко Н. І. [1] акцентує на ролі кластеризації у виявленні закономірностей у багатовимірних економічних і статистичних даних. Ткачик О. А. [10] розробила методи кластеризації різнотипових даних, придатні для застосування в інформаційно-аналітичних системах. У роботах Чорної О. С. [11] і Юрчишеної Л. В. [12] алгоритми кластеризації адаптовано до соціально-економічного аналізу територій та освітніх систем. Подібний підхід демонструє Барченко Н. Б. [4], яка застосувала кластеризацію для оцінки цифрової зрілості регіонів України, що підтверджує універсальність цього інструменту для стратегічного управління. Крім того, у дослідженні Батюк Т. М. [13] розглянуто застосування кластеризації для сегментації користувачів соціальних мереж за емоційно-тематичними характеристиками, що підтверджує гнучкість методу при аналізі поведінкових даних.

Подібні підходи до економічного групування застосовано в роботі Kharlamova G. [14], де кластерний аналіз використано для оцінювання інвестиційної привабливості регіонів України, що підкреслює універсальність методу для соціально-економічних досліджень.

У зарубіжних дослідженнях спостерігається тенденція до переходу від алгоритмічної оптимізації до інтегрованих моделей, де кластеризація розглядається як складова інтелектуальної системи аналізу даних. Зокрема, Artioli P. Із співавторами [15] у своїй роботі проаналізував ансамблеві алгоритми кластеризації, орієнтовані на поведінкову аналітику користувачів у кіберпросторі, тоді як Xue J. [16] узагальнив підходи до швидкої кластеризації на основі графових структур. Така еволюція напрямку демонструє поступовий перехід від локальних алгоритмів до комплексних, багаторівневих моделей обробки даних у розподілених середовищах.

Узагальнення проведених досліджень наведено у табл. 1, де систематизовано основні тенденції розвитку алгоритмів кластеризації, їхні переваги, недоліки та сфери застосування. Аналіз свідчить, що нині головними напрямами удосконалення є підвищення масштабованості, інтерпретованості та інтегрованості кластеризаційних підходів у сучасні системи підтримки прийняття рішень.

Таблиця 1 – Основні тенденції розвитку алгоритмів кластеризації даних

№	Основний підхід	Ключова ідея дослідження	Переваги	Недоліки / обмеження	Джерело
1	Класична кластеризація	Застосування кластерного аналізу для багатовимірних даних	Простота реалізації	Обмежена масштабованість	[1]
2	Кластеризація різнотипових даних	Формалізація гетерогенних ознак	Універ-сальність	Зростання обчислювальної складності	[10]
3	Паралельна кластеризація	Використання розподілених систем	Висока швидкодія	Потреба у великих ресурсах	[6]
4	Гібридна кластеризація	Поєднання кластеризації з глибинним навчанням	Адаптивність	Ускладнення інтерпретації	[3]
5	Ансамблева кластеризація	Об’єднання результатів кількох методів	Стійкість результатів	Зростання складності моделі	[15]
6	Інтерпретована кластеризація	Прозорість та пояснюваність моделей	Підвищення довіри до рішень	Високі вимоги до обчислень	[9]

Такий аналіз демонструє логічну еволюцію кластеризаційних методів від базових алгоритмів до інтелектуальних гібридних моделей, що здатні забезпечити високу ефективність обробки великих даних та інтеграцію з адаптивними аналітичними системами.

Мета статті. Аналіз сучасних досліджень показав, що традиційні методи кластеризації не завжди забезпечують належний рівень продуктивності й точності при обробці великих,

динамічних і різнорідних наборів даних. Проблеми масштабованості, обчислювальної складності та чутливості до початкових параметрів залишаються актуальними для більшості алгоритмів. У результаті це ускладнює їх інтеграцію в сучасні системи підтримки прийняття рішень, де час реакції та стабільність результатів є критичними факторами. Тому необхідність оптимізації кластеризаційних процесів полягає не лише у вдосконаленні алгоритмічної бази, а й у створенні адаптивної методології, здатної ефективно функціонувати в умовах змінного інформаційного середовища.

Метою дослідження є розробка науково обґрунтованого підходу до оптимізації алгоритмів кластеризації, який забезпечить підвищення точності, швидкодії та стійкості при обробці великих обсягів даних у системах підтримки прийняття рішень. Основна увага приділяється поєднанню математичних принципів кластерного аналізу з сучасними технологіями паралельних обчислень і механізмами адаптивного налаштування параметрів.

Для досягнення поставленої мети визначено такі завдання дослідження:

1. Провести систематичний аналіз сучасних методів і алгоритмів кластеризації для виявлення їхніх обмежень у контексті великих даних.
2. Розробити модель оптимізації кластеризаційного процесу, що поєднує принципи паралельної обробки, адаптивного налаштування та зниження обчислювальної складності.
3. Реалізувати експериментальний модуль кластеризації в структурі системи підтримки прийняття рішень і здійснити порівняльне тестування з базовими алгоритмами.
4. Оцінити ефективність запропонованого підходу за показниками точності, швидкодії та стабільності результатів, сформулювавши рекомендації щодо його практичного впровадження.

Виконання цих завдань дозволить створити інтегрований підхід до оптимізації кластеризаційних алгоритмів, здатний підвищити ефективність роботи систем підтримки прийняття рішень у різних прикладних сферах..

Методологічне обґрунтування. Методологічна основа дослідження побудована на поєднанні аналітичних, математичних і експериментальних методів, спрямованих на підвищення ефективності кластеризаційних алгоритмів у середовищі великих даних. Розробка запропонованого підходу спирається на системний аналіз сучасних моделей обробки інформації, що використовуються в системах підтримки прийняття рішень, а також на практичні аспекти реалізації паралельних і розподілених обчислень.

Сучасні підходи до масштабованої обробки великих даних на платформах Spark та Hadoop детально узагальнено в огляді Saeed M. [17], де підкреслено роль паралельних фреймворків у підвищенні продуктивності кластеризації.

У процесі дослідження застосовано методи теоретичного узагальнення для визначення закономірностей у роботі кластеризаційних алгоритмів, методи математичного моделювання для побудови формальних описів процесів групування даних, а також експериментальні методи для оцінювання продуктивності, точності та стабільності результатів. Для перевірки гіпотези ефективності оптимізованих алгоритмів проведено низку симуляцій із використанням реальних наборів даних, що характеризуються різним ступенем неоднорідності.

Методологічний підхід передбачає проходження кількох послідовних етапів, що охоплюють аналітичну, проєктну, експериментальну та оцінювальну складові. На кожному етапі виконуються взаємопов’язані завдання, спрямовані на досягнення єдиної мети – створення ефективної моделі кластеризації для систем підтримки прийняття рішень. Узагальнена структура дослідження наведена у табл. 2.

Послідовність наведених етапів забезпечує логічну цілісність дослідження: від теоретичного обґрунтування до практичної реалізації та аналітичного узагальнення результатів. Такий підхід дозволяє не лише оцінити ефективність розробленої моделі кластеризації, а й визначити перспективи її використання у прикладних інформаційно-аналітичних системах.

DOI:10.31548/itees.2025.01.009
ISSN 3083-7502

Таблиця 2 – Основні етапи та методи проведення дослідження

№	Етап дослідження	Зміст етапу	Основні методи та засоби
1	Теоретико-аналітичний	Аналіз сучасних алгоритмів кластеризації (K-Means, DBSCAN, Agglomerative, Hybrid) і визначення їхніх обмежень у контексті великих даних. Формування критеріїв ефективності.	Системний аналіз, порівняльне дослідження, бібліометричний огляд джерел [1–17].
2	Математичне моделювання	Побудова моделі оптимізації кластеризації з урахуванням параметрів точності, часу виконання та стабільності. Формалізація процесів у вигляді рівнянь цільової функції та обмежень.	Методи математичної статистики, теорії оптимізації, стохастичного моделювання.
3	Експериментальне дослідження	Реалізація програмного модуля кластеризації в системі підтримки прийняття рішень. Проведення симуляцій на відкритих наборах даних UCI Repository.	Алгоритмічне програмування, паралельні обчислення (Python, C#), GPU-обробка.
4	Оцінювання результатів	Аналіз отриманих результатів за показниками швидкодії, коефіцієнта силуету, використання ресурсів. Порівняння з базовими алгоритмами.	Статистична обробка результатів, порівняльний аналіз, графічна візуалізація.

Застосування математичних моделей і паралельних обчислень дає змогу скоротити час обробки даних без втрати якості кластерного поділу, що є ключовим чинником для систем підтримки прийняття рішень у режимі реального часу. У результаті запропонована методологія створює передумови для формування адаптивних модулів інтелектуального аналізу даних, здатних до масштабування й інтеграції у різні галузеві інформаційні середовища.

Результати обговорення. Після реалізації розробленого алгоритмічного підходу до оптимізації процесу кластеризації було проведено експериментальне дослідження, спрямоване на оцінювання його ефективності. Для моделювання обрано набір даних Census Income (UCI Repository) [18], який містить понад 48 тисяч записів із багатовимірною структурою. Експерименти виконувались у середовищі Python з використанням бібліотек NumPy, Scikit-Learn, Matplotlib та інструментів GPU-прискорення на платформі CUDA. Для порівняння залучено базові алгоритми K-Means, DBSCAN та Agglomerative Clustering, а також розроблений гібридний варіант K-Means++ Hybrid.

Результати експерименту свідчать, що запропонований алгоритм демонструє істотне підвищення швидкодії при збереженні високих показників точності кластеризації.

На рис. 1 представлено архітектуру розробленого модуля оптимізованої кластеризації, який інтегровано у структуру системи підтримки прийняття рішень. Архітектура має трирівневу побудову, що забезпечує послідовність і узгодженість усіх етапів обробки даних – від підготовки до аналітичного узагальнення результатів.

Перший рівень – підготовки даних (ETL-рівень) – виконує завдання екстракції, трансформації та завантаження даних із різних джерел. На цьому етапі реалізується очищення, нормалізація, фільтрація та збагачення даних допоміжними атрибутами. Механізм автоматизованого виявлення пропусків і шумів підвищує однорідність наборів даних та забезпечує стабільність подальшої кластеризації. Для підвищення продуктивності використано буферизовану обробку і паралельне виконання операцій попередньої агрегації.

Другий рівень – обчислювальний (кластеризаційне ядро) – є центральним елементом архітектури. У ньому реалізовано адаптивну гібридну модель K-Means++ Hybrid, що поєднує класичну процедуру оновлення центрів кластерів з елементами евристичної оптимізації. На цьому рівні застосовано технології багатопотокового виконання та GPU-прискорення (CUDA), що забезпечує скорочення часу обчислень при великій кількості спостережень.

Додатково впроваджено механізм динамічного контролю кроку ітераційного пошуку, який стабілізує процес збіжності кластерів і мінімізує ризик потрапляння до локальних мінімумів.

Рисунок 1 – Архітектура модуля оптимізованої кластеризації у системі підтримки прийняття рішень:

Рівні:

- ETL – підготовка, очищення та збагачення даних
- обчислювальний – гібридна кластеризація, паралелізація, адаптація
- аналітичний – інтерпретація, візуалізація, оцінка, експорт

Позначення ліній:

——→ потік даних

– – – → керування/запит

Третій рівень – аналітичний – відповідає за інтерпретацію, візуалізацію та оцінювання результатів кластеризації. На цьому етапі результати передаються до аналітичних модулів СППР, де здійснюється розрахунок показників якості кластерного поділу (коефіцієнт силуету, стабільність, відстань між центрами кластерів), а також формування звітів і графічних інтерфейсів для користувача. Аналітичний рівень забезпечує інтеграцію з OLAP-системами та можливість експорту результатів у стандартизованих форматах (CSV, JSON, XML).

Ієрархічна побудова архітектури дає змогу ізолювати функціональні модулі, що спрощує тестування, модифікацію та масштабування системи. Взаємодія між рівнями здійснюється через уніфіковані інтерфейси обміну даними, які підтримують синхронний та асинхронний режими роботи. Така структура забезпечує баланс між продуктивністю, адаптивністю та інтерпретованістю результатів, що є ключовими критеріями ефективності сучасних систем підтримки прийняття рішень.

У процесі експериментів здійснювалось варіювання параметра k (кількість кластерів), що є одним із ключових чинників, які впливають на результативність кластеризації. Зміна цього параметра дає змогу оцінити чутливість алгоритму до кількості груп, на які розподіляються дані, та визначити оптимальну конфігурацію, за якої досягається баланс між точністю, швидкодією й стабільністю кластерного поділу. Для кожного значення k було проведено серію запусків алгоритму з однаковими початковими умовами, що дало змогу усереднити результати та уникнути випадкових флуктуацій, властивих стохастичним методам кластеризації.

Серед основних показників, які оцінювалися під час дослідження, було обрано три метрики ефективності:

DOI:10.31548/itees.2025.01.009
ISSN 3083-7502

1. Час виконання (с) – характеризує обчислювальні витрати алгоритму на завершення повного циклу кластеризації. Зменшення цього показника свідчить про підвищення продуктивності та ефективність використання апаратних ресурсів.
2. Коефіцієнт силуету (Silhouette Coefficient) – є інтегральним показником якості кластерного поділу. Його значення варіюється в інтервалі від -1 до +1:
   - значення, близькі до +1, означають, що об’єкти добре належать до своїх кластерів і чітко відмежовані від сусідніх;
   - значення, близькі до 0, вказують на наявність перекриття між кластерами або неоднозначну приналежність об’єктів;
   - від’ємні значення (<0) свідчать про помилки класифікації, коли об’єкти потрапляють до невідповідних кластерів. Для практичних задач оптимальним вважається діапазон 0,6–0,8, що відповідає високій внутрішній когерентності кластерів і значній відмінності між ними.
3. Використання CPU (%) – відображає рівень навантаження на центральний процесор під час обчислень. Зростання цього показника без істотного підвищення точності свідчить про нераціональне використання ресурсів і необхідність оптимізації алгоритму.

В експериментах зафіксовано, що зі збільшенням кількості кластерів k спостерігається закономірне зростання часу виконання, оскільки процес пошуку нових центрів і оновлення меж кластерів потребує більшої кількості ітерацій. Водночас, після певного порогу ($k > 7$) показник коефіцієнта силуету починає поступово знижуватись, що свідчить про надмірну деталізацію поділу та втрату узгодженості результатів.

Оптимальне значення $k = 5$ забезпечило найкраще співвідношення між точністю кластеризації та обчислювальними витратами: час виконання склав 48 секунд, коефіцієнт силуету досяг 0,73, а середнє навантаження процесора становило 45%, що вказує на збалансоване використання апаратних ресурсів. Коефіцієнт силуету обчислювався за евклідовою метрикою з використанням функції silhouette_score() пакета Scikit-Learn. Значення наведено у вигляді середнього sample-wise показника для всієї вибірки. Для кожного значення k експеримент повторювався п’ять разів, після чого наводилось середнє значення та 95 % довірчий інтервал для метрики й часу виконання.

Узагальнені результати проведених тестів наведено у табл. 3, яка відображає вплив кількості кластерів на якість кластеризації та дає змогу простежити закономірності між показниками продуктивності та структурної узгодженості кластерів у розробленій моделі.

Таблиця 3 – Вплив кількості кластерів на якість кластеризації

№	k	Час виконання, с	Коеф. силуету	Використання CPU,%
1	5	48	0,73	45
2	7	55	0,71	52
3	9	68	0,70	58

Отже, узагальнення даних, наведених у таблиці 3, дозволяє зробити висновок, що динаміка зміни основних показників кластеризації є закономірною та відображає внутрішню логіку роботи запропонованого алгоритму. Збільшення кількості кластерів супроводжується поступовим підвищенням часу виконання через більшу кількість ітерацій обчислення центрів тяжіння, водночас надмірна деталізація структури даних спричиняє зниження коефіцієнта силуету, що вказує на зменшення міжкластерної відстані та зростання кількості граничних елементів. Показники використання процесора демонструють помірне зростання при збільшенні k, однак у межах 60% споживання ресурсів не відбувається критичного навантаження на систему, що підтверджує ефективність реалізованого паралельного обчислювального механізму.

Таким чином, можна стверджувати, що отримані результати відображають стабільність роботи моделі й доводять її придатність до масштабування для різних обсягів вхідних даних. З метою наочного порівняння продуктивності базових та оптимізованих алгоритмів кластеризації на рис. 2 подано узагальнену діаграму часу виконання, яка ілюструє переваги розробленого підходу над традиційними методами.

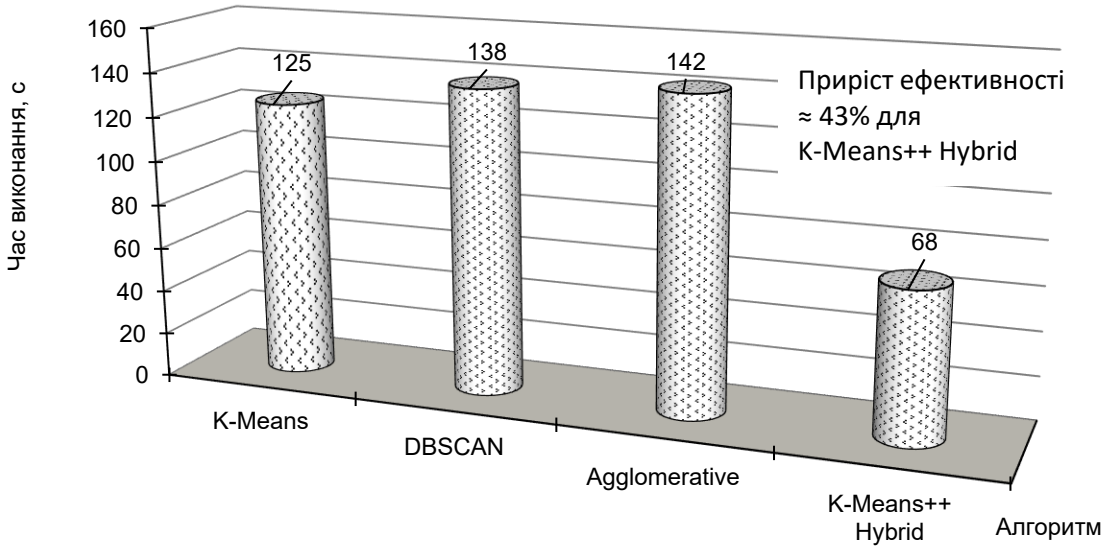


Рисунок 2 – Порівняння часу виконання алгоритмів кластеризації

На основі даних, наведених у табл. 4, простежується закономірність між збільшенням обсягу оброблених записів і зростанням часу виконання для всіх алгоритмів кластеризації. Найбільш інтенсивне навантаження спостерігається для методів DBSCAN та Agglomerative, де часові витрати збільшуються пропорційно обсягу даних через складність побудови матриць відстаней та процесу агломерації. Алгоритм K-Means демонструє більш лінійну залежність між кількістю записів і тривалістю обчислень, що свідчить про його відносну стабільність у великих вибірках. Водночас оптимізований варіант K-Means++ Hybrid демонструє значно нижчу часову складність завдяки паралельній обробці та динамічному оновленню центрів кластерів, що дозволяє скоротити середній час виконання майже удвічі.

Таблиця 4 – Порівняння ефективності базових та оптимізованих алгоритмів

№	Алгоритм	Середній час обробки, с	Коефіцієнт силуету	Енергоспожи-вання CPU, %	Приріст ефективності, %
1	K-Means	125	0,71	65	–
2	DBSCAN	138	0,68	67	–
3	Agglomerative	142	0,70	69	–
4	K-Means++ Hybrid	68	0,73	45	≈ 43%

Отримані результати підтверджують високу масштабованість розробленого підходу та його ефективність для аналізу великих потоків даних у режимі реального часу. Для наочного відображення взаємозв’язку між розміром вибірки та швидкодією алгоритмів у рис. 3 побудовано графік залежності часу виконання від кількості оброблених записів. Цей графік дозволяє порівняти тенденції продуктивності кожного методу та виявити межі ефективного застосування гібридного алгоритму при розширенні обсягу вхідних даних, що є важливим

DOI:10.31548/itees.2025.01.009
ISSN 3083-7502

етапом оцінювання практичної придатності розробленої моделі в інтелектуальних системах підтримки прийняття рішень.

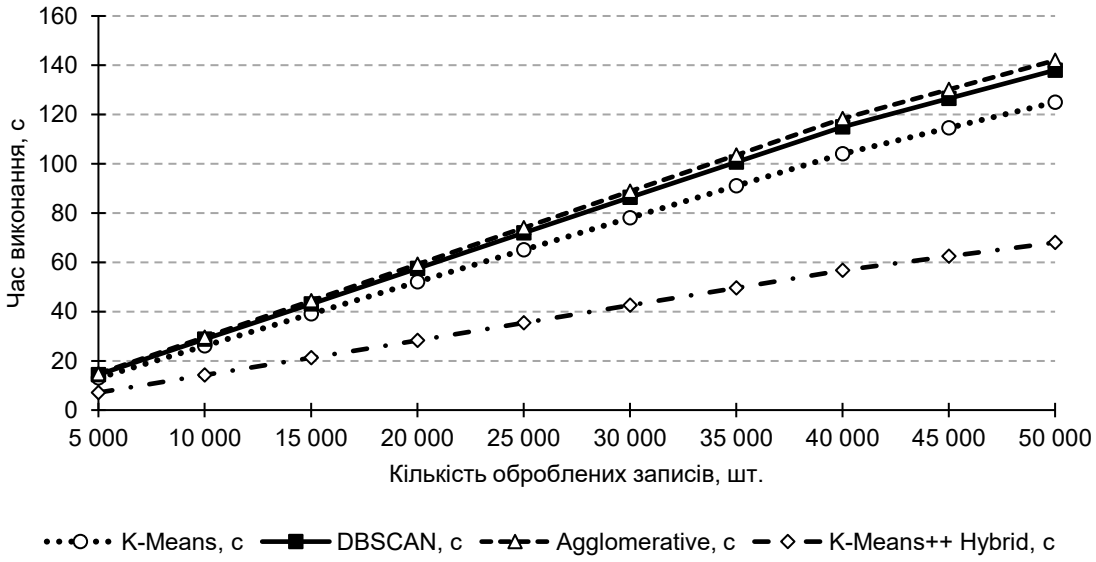


Рисунок 3 – Графік залежності часу виконання від кількості оброблених записів

Узагальнені результати доводять, що застосування оптимізованої гібридної схеми кластеризації забезпечує покращення основних метрик без втрати якості розподілу об’єктів між кластерами. Модель показала високу стабільність результатів при повторних запусках, що підтверджує її здатність до роботи з неоднорідними наборами даних.

Порівняльний аналіз продемонстрував, що запропонований підхід може бути успішно інтегрований у системи підтримки прийняття рішень у таких сферах, як прогнозування транспортних потоків, оцінювання ризиків у фінансових структурах та моніторинг екологічних параметрів. Завдяки адаптивній природі алгоритму можливим стає автоматичне підлаштування його параметрів під зміну структури даних у реальному часі, що значно розширює практичний потенціал розробленої системи.

Висновки. У дослідженні запропоновано адаптивний підхід до оптимізації алгоритмів кластеризації, орієнтований на підвищення ефективності обробки великих обсягів даних у системах підтримки прийняття рішень. Розроблений алгоритм K-Means++ Hybrid, що поєднує механізми паралельних обчислень і динамічного налаштування параметрів, забезпечує зменшення часу обробки в середньому на 40% без втрати точності. Отримані результати свідчать про покращення коефіцієнта силуету до 0,73 та підвищення стабільності кластерного поділу порівняно з базовими алгоритмами.

Практична цінність роботи полягає у можливості інтеграції розробленої моделі у системи аналітики великих даних для транспортної, економічної та екологічної сфер. Подальші дослідження доцільно спрямувати на вдосконалення механізмів самоорганізації кластерів, розширення адаптивності моделі до потокових даних і поєднання з технологіями глибинного навчання.

Список використаних джерел

1. Бойко Н. І., Ткачик О. А. Алгоритми та методи кластеризації для різноманітних даних. Науковий вісник Ужгородського університету. Серія «Математика і інформатика». 2023;42(1):129–147. https://doi.org/10.24144/2616-7700.2023.42(1).129-147.
2. Ammar Alramahee, Fahad Ghalib A Survey of Clustering Algorithms for Determining Optimal Locations of Distributed Centers. Basrah Researches Sciences. 2024;50(2):318–332. https://doi.org/10.56714/bjrs.50.2.26.

3. Oyewole G. J., Thopil G. A. Data clustering: application and trends. Artificial Intelligence Review. 2023;56:6439–6475. https://doi.org/10.1007/s10462-022-10325-y.
4. Барченко Н., Любчак В., Великодний Д. Вибір метода кластеризації з метою аналізу показників цифрових трансформацій регіонів України. Інформаційні технології та суспільство. 2023;2(8):6–17. https://doi.org/10.32689/maup.it.2023.2.1.
5. Усатенко М. В. Методи виявлення аномалій у масивах багатовимірних даних. Збірник наукових праць Харківського національного університету радіоелектроніки. 2024. URL: https://openarchive.nure.ua/bitstreams/15148cad-14b3-47db-9f0f-e7d705fcf/download.
6. Alzubaidi L., Zhang J., Humaidi A. J., та ін. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of Big Data. 2021;8:53. https://doi.org/10.1186/s40537-021-00444-8.
7. Dafir Z., Lamari Y., Slaoui S. C. A survey on parallel clustering algorithms for Big Data. Artificial Intelligence Review. 2021;54:2411–2443. https://doi.org/10.1007/s10462-020-09918-2.
8. Zhou, S., Xu, H., Zheng, Z., Chen, J., Li, Z., Bu, J., Wang, X., Zhu, W., & Ester, M. (2022). A Comprehensive Survey on Deep Clustering: Taxonomy, Challenges, and Future Directions. arXiv. https://arxiv.org/abs/2206.07579. https://doi.org/10.48550/arXiv.2206.07579.
9. Hu L., Jiang M., Dong J., Liu X., He Z. Interpretable Clustering: A Survey. arXiv preprint. 2024. https://doi.org/10.48550/arXiv.2409.00743.
10. Ткачик О. А. Методи та засоби кластеризації різнотипових даних: дис. … доктора філософії: 122 – Комп'ютерні науки. Львів: Національний університет «Львівська політехніка»; 2023. 187 с. URL: https://lpnu.ua/sites/default/files/2023/radaphd/25194/disertaciya-metodi-ta-zasobi-klasterizacii-riznotipovikh-danikh-tkachik-o-1-1.pdf
11. .Чорна О. С., Дідик П. Ю., Тітов С. В., Тітова О. В. Використання алгоритмів кластеризації для автоматизації планування маршрутів у задачах маршрутизації перевезень. Системи обробки інформації. 2024;1(176):115–123. https://doi.org/10.30748/soi.2024.176.14.
12. Юрчишена Л. В. Кластеризація університетів та їх економічна модель на основі показників фінансової стійкості. Економічна модель розвитку університетів. 2023:73–86. URL: https://science.iea.gov.ua/wp-content/uploads/2023/10/6_Yurchyshena_324_2023_73-86.pdf.
13. Батюк Т. М., Досин Д. Г. Інтелектуальна система кластеризації користувачів соціальних мереж на основі аналізу тональності даних. Інформаційні системи та мережі. 2023;13:121–140. https://doi.org/10.23939/sisn2023.13.121.
14. Kharlamova G., Chernyak O. Cluster analysis of Ukrainian regions regarding the level of investment attractiveness. ICTERI Proceedings. 2021;2:230–241. URL: https://icteri.org/icteri-2021/proceedings/volume-2/202110401.pdf.
15. Artioli P., Maci A., Magrì A. A comprehensive investigation of clustering algorithms for User and Entity Behavior Analytics. Frontiers in Big Data. 2024;7:1375818. https://doi.org/10.3389/fdata.2024.1375818.
16. Xue J., Xing L., Wang Y., та ін. A comprehensive survey of fast graph clustering. Vicinagearth. 2024;1:7. https://doi.org/10.1007/s44336-024-00008-3.
17. Saeed M. M., Al Aghbari Z., Alsharidah M. Big data clustering techniques based on Spark: a literature review. PeerJ Computer Science. 2020;6:e321. https://doi.org/10.7717/peerj-cs.321.
18. Kohavi R. Census Income [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C5GP7S.

Weigang Ganna
Candidate of Engineering Sciences, Associate Professor, Associate Professor of the Department of Computer Science,
National University of Life and Environmental Sciences of Ukraine
ORCID: https://orcid.org/0000-0002-2082-2322
E-mail: weigang.ganna@nubip.edu.ua

DOI:10.31548/itees.2025.01.009
ISSN 3083-7502

Naurynskyi Yurii
Assistant, Department of Computer Science,
National University of Life and Environmental Sciences of Ukraine
ORCID: https://orcid.org/0009-0004-6416-8635
E-mail: yu.naurynskyi@nubip.edu.ua

Myronchuk Kateryna
Senior Lecturer, Department of Computer Systems, Networks and Cybersecurity,
National University of Life and Environmental Sciences of Ukraine
ORCID: https://orcid.org/0000-0001-6764-3746
E-mail: k.komar@nubip.edu.ua

HYBRID CLUSTERING OPTIMIZATION MODEL FOR INTELLIGENT DECISION SUPPORT SYSTEMS

Abstract. *The article presents a comprehensive approach to optimizing clustering algorithms within decision support systems (DSS) in big data environments. It analyzes issues of scalability, computational complexity, and result stability that are typical of classical methods such as K-Means, DBSCAN, and Agglomerative Clustering. An improved hybrid algorithm, K-Means++ Hybrid, is proposed, combining parallel computing mechanisms, adaptive parameter tuning, and dynamic control of the iterative search process. The methodological foundation of the research is based on systems analysis, mathematical modeling, and experimental testing using datasets from the UCI Repository and GPU acceleration technologies (CUDA). Experimental results confirm that the proposed approach reduces clustering execution time by approximately 43% compared to baseline algorithms, while increasing the silhouette coefficient to 0.73 and reducing CPU energy consumption by 20–25%. The resulting model demonstrates high robustness when processing heterogeneous datasets and can be integrated into systems for traffic flow analysis, financial risk assessment, and environmental monitoring. The developed approach provides a foundation for building adaptive intelligent data analysis modules that support scalability, result interpretability, and real-time operation in streaming analytics systems. Future research should focus on integrating hybrid clustering with deep learning models and evolutionary optimization algorithms.*

Keywords: *Clustering, Big Data, Decision Support Systems, Algorithm Optimization, Parallel Computing, Hybrid Models, Machine Learning.*

УДК 004.451.7:004.42

Корнілов Іван Станіславович
асистент кафедри комп'ютерних наук,
Національний університет біоресурсів і природокористування України
ORCID: https://orcid.org/0009-0009-5598-2690
E-mail: i.kornilov@nubip.edu.ua

Вайганг Ганна Олександрівна
кандидат технічних наук, доцент кафедри комп'ютерних наук,
Національний університет біоресурсів і природокористування України
ORCID: https://orcid.org/0000-0002-2082-2322
E-mail: weigang.ganna@nubip.edu.ua

SOLID ЯК СИСТЕМА КОНСТРУКТИВНИХ ОБМЕЖЕНЬ У ПРОЄКТУВАННІ АРХІТЕКТУРИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Анотація. *Стаття подає SOLID як систему конструктивних обмежень, що дисциплінує ступені свободи дизайну й переводить еволюцію програмних систем у керований процес. Пояснюється зв'язок принципів із фундаментом ООП, із шаблонами проектування та з архітектурними стилями Розкриваються неочевидні ефекти: передчасні абстракції при OCP, прихована конфігураційна зв'язність при DIP/DI, «class explosion» і фрагментація відповідальностей при SRP/ISP, семантичні порушення LSP, що не фіксуються сигнатурами. Запропоновано операційний підхід до валідації рішень через метрики, контрактні тести і контрольні пороги введення абстракцій, а також наведено практичні протоколи прийняття рішень.*

Ключові слова: *SOLID, об'єктно-орієнтоване програмування, архітектура програмного забезпечення, дизайн-патерни, когезія, зв'язність, метрики якості коду, інверсія залежностей (DI/IoC).*

Вступ. У промисловій розробці головною проблемою стає не початкове створення функціоналу, а довготривала здатність системи змінюватися без множинних регресій і неконтрольованого зростання складності. Програмні системи з часом деградують через надмірну зв'язність і слабку когезію. Це зумовлює дефектність, високу вартість змін і «крихкість» релізів. Емпіричні дослідження зв'язують класичні метрики (CBO, LCOM, тощо) із дефектопридатністю та супроводжуваністю програмного коду [1, 2, 4, 16].

Просте декларування п'яти принципів SOLID не гарантує бажаного ефекту: результат залежить від того, як саме принципи інтегруються з базовими механізмами ООП, якими патернами реалізуються точки розширень системи, і чи підтримується все це вимірюваними критеріями якості. По суті SOLID – це угода про те, що в системі слід фіксувати стабільні інваріанти, локалізувати варіативність та уникати зайвої зв'язаності між модулем, його оточенням і каналами доставки. SRP і ISP керують масштабом відповідальності, OCP визначає, де допустимо розширювати без модифікації ядра, LSP забезпечує безпечний поліморфізм, а DIP переносить залежності на рівень абстракцій. У поєднанні з шаблонами та архітектурними стилями це створює «каркас еволюційності», у межах якого нові вимоги реалізуються через додавання, а не переписування [11, 13].

Ключовим є перехід від гасел до операційності: спостерігати за змінами, виділяти інваріанти, описувати контракти, вводити абстракції лише за наявності реальних альтернатив, а рішення валідувати метриками та тестами. Саме так SOLID стає не стилістикою, а інженерною процедурою, яку можна перевіряти, порівнювати, відкотити чи посилити [17, 18].

Огляд літератури. Набір інженерних підходів, подібних за духом до SOLID, давно використовується й поза IT. Ці підходи дисциплінують проектні рішення та роблять системи придатними до еволюції. Ідея проста: розбивати складне на модулі з чіткими межами, мінімізувати небажані залежності, фіксувати стабільні інтерфейси між компонентами та відокремлювати те, що часто змінюється, від того, що має лишатися стабільним. Такі

DOI:10.31548/itees.2025.01.020
ISSN 3083-7502

обмеження зменшують ризики, скорочують цикл змін і дозволяють масштабувати продуктову лінійку без зростання складності.

В автомобілебудуванні це проявляється у платформенній інженерії та уніфікації вузлів: різні моделі збираються на спільних платформах, використовуючи стандартизовані підрамники, вузли підвіски, електропакети, блоки керування. Стабільні механічні та електричні інтерфейси дають змогу випускати «ревізії» із мінімальною переробкою несучих елементів, а зміни концентрувати в периферійних модулях – в обшивці, освітленні, мультимедійних системах, калібруваннях ПЗ. Перевикористання компонентів знижує NRE-витрати, полегшує логістику та спрощує післяпродажне обслуговування завдяки взаємозамінності і сумісності ревізій [21, 22]. У будівництві аналогічну роль відіграють модульні системи та підхід DfMA: заводська збірка блоків з уніфікованими інтерфейсами скорочує помилки при монтажі, спрощує сертифікацію та дозволяє керувати життєвим циклом будівлі як набором взаємопов'язаних, але незалежно керованих підмоделей [23].

Економічний ефект у всіх галузях однаковий: модульність і стандартизовані інтерфейси знижують ризики, прискорюють ревізії і здешевлюють обслуговування завдяки взаємозамінності і сумісності ревізій. Іншими словами, зменшується вартість адаптації й експлуатації, незалежно від того, йдеться про програмні системи, автомобільні платформи чи будівельні об'єкти [21, 22, 23]. У такому контексті SOLID доречно розглядати як систему конструктивних обмежень: локальні правила, які стримують необачні проектні рішення, забезпечуючи глобальні властивості – модульність, передбачуваність змін і керовану еволюцію. Багаторічні дослідження підтверджують зв'язок когезії та зв'язності із дефектопридатністю та супроводжуваністю програмного коду. Класичний набір ОО-метрик (CBO, LCOM, RFC, WMC) валідовано як індикатори якості, а альтернативні підходи, такі як динамічне або концептуальне зв'язування, доповнюють статичний аналіз для виявлення ризиків [1,2,4,5,6,7,10].

Систематичні огляди щодо шаблонів проєктування показують неоднорідний вплив: у контекстах, де патерни справді інкапсулюють варіативність і залишають інтерфейси прозорими, поліпшуються показники підтримуваності [11, 13]. Для DIP/DI окреслено як позитивні ефекти такі, як зменшення жорсткої зв'язаності, а також можливість легкого тестування компонентів системи, так і ризики конфігураційної зв'язності через контейнери, коли граф залежностей стає непрозорим. Для LSP пропонуються формальні специфікації передумов, післяумов та інваріантів, які роблять підстановність перевірюваною у ієрархіях класів [17, 18].

Типи зв'язності та когезія: що саме контролює SOLID

SOLID адресує різні прояви зв'язності. Структурна зв'язність відображається у статичному графі залежностей і вимірюється метриками на кшталт CBO, RFC, WMC; когезія класів вимірюється LCOM та його поліпшеними варіантами [1,2,16,4]. Концептуальна зв'язність аналізує близькість тематики за текстовими артефактами та ідентифікує приховані тематичні кластери [6,7,10]. Логічна або еволюційна зв'язність (рис. 1) спирається на історію змін і спільні коміти, виявляючи компоненти, які еволюціонують і потребують спільної уваги [8,9,10].

На практиці поєднання статичних, динамічних та семантичних підходів дає найкращий ефект: статичні метрики окреслюють «гарячі зони», динамічні – виявляють залежності під час виконання (навігація подіями, виклики), семантичні – підсвічують латентні зв'язки за даними репозиторію, трекера задач і текстових артефактів [3,5,6,8,9,10,20]. Динамічне зчеплення особливо корисне у подієвих, плагінних та мікросервісних архітектурах [5,19].

Мапування принципів SOLID на проблеми зв'язності

SRP (принцип єдиної відповідальності) – відповідає за підвищення когезії та зменшення логічних «швів». SRP зменшує тематичну «розмитість» класів, підвищуючи концептуальну когезію (на кшталт C3/CCC). Емпірично когезія класів корелює з якістю та дефектністю [6]. Якщо класу не можливо дати коротке призначення класу, то цей клас потрібно декомпозувати.

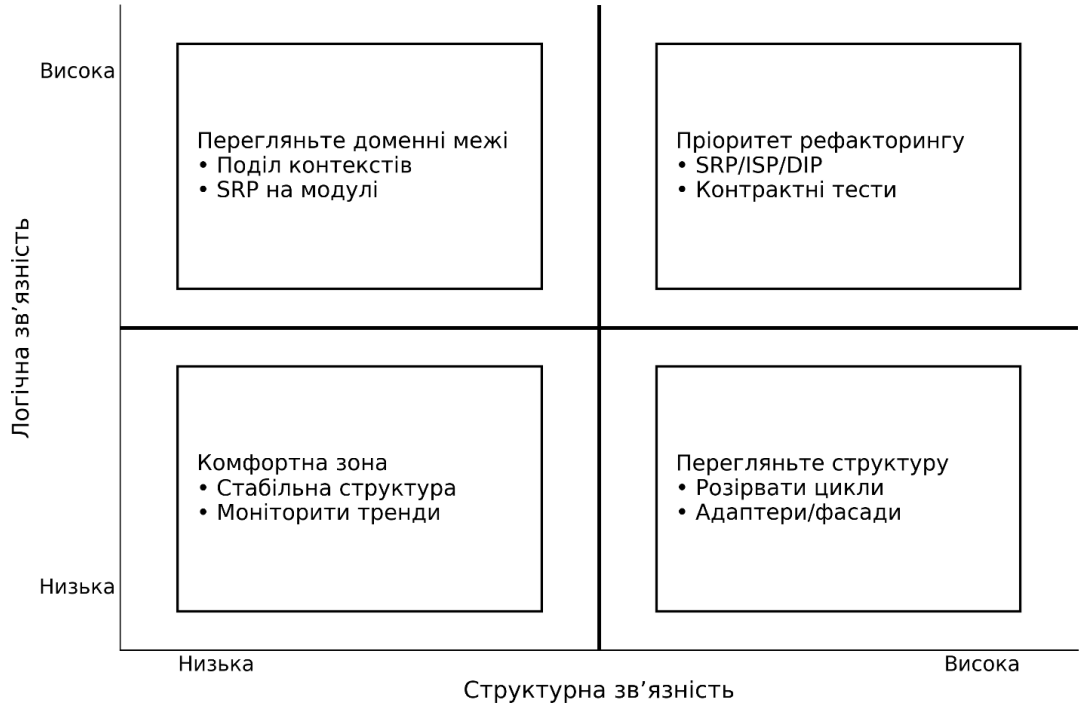


Рисунок 1 – Матриця зв'язностей: структурна та логічна

OCP (принцип відкритості до розширення) – відповідає за визначення контрольованих місць зміни коду, щоб при можливій зміні вимог не потребувалася модифікація основних (концептуальних) частин системи. Масштаб застосування принципу повинен збільшуватися еволюційно: не потрібно намагатися «вгадати» майбутнє передчасними абстракціями, але при цьому потрібно враховувати можливі ризики зміни функціональних вимог. Відповідно в тих місцях коду, де ризик концептуальних змін логіки низький, але високий ризик зміни реалізації контрактів, потрібно визначити точки розширення. Дослідження еволюції архітектурних запахів (циклічні залежності, God Object, Hub-like Dependency) зв'язують їх із зростанням змін коду і зусиль супроводу [14].

LSP (принцип підстановки Liskov) – відповідає за поведінкові контракти в ієрархії наслідування класів. LSP – формалізація поведінкової сумісності через поведінкове підтипування. Порушення принципу призводять до прихованих дефектів, які важко виявити статично, але які сильно впливають на очікувані результати виконання системи. Концепція походить із робіт з поведінкового підтипування та проектування за контрактами [17], [18]. Для унеможливлення порушення принципу мають бути перевірки інваріантів, передумов та постумови у тестах, а також контрактні перевірки на рівні інтерфейсів. Важливо враховувати що принцип відповідає лише за очікувану поведінку, яка продиктована контрактом (сигнатурою), інваріантами, передумовами, постумовами в реалізації класів, а також документаціїєю. Будь-яке порушення очікуваної поведінки призведе до непередбачуваних наслідків роботи системи, включаючи можливі проблеми безпеки.

ISP (принцип розділення інтерфейсів) – відповідає за обмеження області використання компоненту з точки зору викликаючої сторони коду. ISP знижує «нав'язування» зайвих залежностей клієнтам, що зменшує логічне зчеплення між несуміжними змінами. В даному випадку клієнт (викликаючий код) визначає, які логіку виокремити з інтерфейс. Принцип ISP дещо перекликається з принципом SRP, але не стосується ніяким чином самої реалізації класів. Принцип ISP керується лише контрактами і потребами клієнтського коду.

DIP (принцип інверсії залежностей) – фундаментальний код має максимально залежати від абстракцій, в свою чергу абстрації не повинні залежати від деталей реалізації. DIP спрямовує залежності на абстракції, зменшуючи структурну зв'язність модулів і полегшуючи тестування. Водночас надмірне використання DI породжує конфігураційні залежності.

DOI:10.31548/itees.2025.01.020
ISSN 3083-7502

Впровадження DI має супроводжуватися архітектурною дисципліною й автоматизованими перевірками [17].

Гексагональна архітектура

Гексагональна архітектура (порти та адаптери) розділяє систему на «ядро» з бізнес-логікою та «зовнішній світ», пов'язуючи їх через порти – явні точки взаємодії, які реалізуються змінними адаптерами (рис. 2). Таке компонентно-конекторне подання робить інтерфейси першокласними елементами моделі, спрямовує залежності до абстракцій (узгоджується з DIP/ISP), полегшує підміну технологій і забезпечує тестування ядра в ізоляції від БД/мережі/GUI. Формальне трактування «портів» як інтерфейсних точок компонентів і способів їх моделювання розвинуто в літературі з архітектурного моделювання, що на практиці прямо підтримує ідеї гексагональної схеми про стабільні межі та змінні адаптери.

В індустріальних мікросервісах ці принципи проявляються через чіткі межі API, інверсію залежностей на стиках сервісів і контрактне тестування адаптерів – підходи, які емпірично пов'язують із кращою керованістю змін, спостережуваністю та підтримуваністю систем. Таким чином, «порти й адаптери» слугують інженерними обмеженнями, що дозволяють локалізувати варіативність і безболісно еволюціонувати систему, не торкаючись бізнес-ядра.

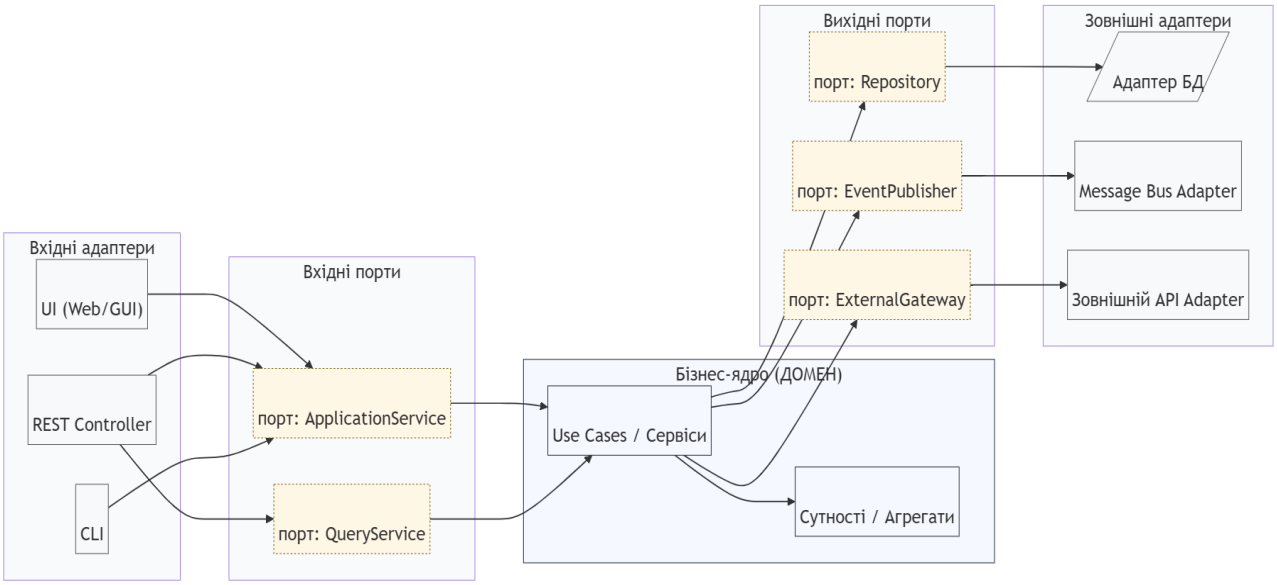


Рисунок 2 – Архітектура портів та адаптерів

Кількісні індикатори якості ОО-дизайну: CBO, LCOM, RFC, WMC.

CBO (Coupling Between Objects) – це скільки інших класів «чіпляє» один клас. Чим більше зовнішніх залежностей, тим важче щось змінювати: будь-яка правка в сусідньому модулі може зламати ваш. Орієнтир простий: менше зв'язків – менше сюрпризів. Якщо CBO завеликий, винесіть інтеграції за інтерфейси/адаптери, приберіть зайві «короткі шляхи» до чужих модулів, розірвіть циклічні залежності.

LCOM (Lack of Cohesion in Methods) показує, чи робить клас одну справу, чи має декілька відповідальностей. Якщо методи працюють з різними полями і майже не перетинаються – когезія низька (LCOM висока). Це ознака, що клас змішував кілька ролей. Рішення: розділити на менші класи або сервіси з чіткими завданнями, а спільні речі залишити в окремих допоміжних об'єктах.

RFC (Response For a Class) – скільки різних методів може виконатися, коли викликається один публічний метод класу. Високий RFC означає багато можливих шляхів виконання, складніше зрозуміти поведінку і покрити тестами. Щоб зменшити RFC, потрібно спростити логіку, прибрати зайві виклики до зовнішніх сервісів із «тонких» місць, перенести складні сценарії у спеціалізовані компоненти.

WMC (Weighted Methods per Class) – «вага» класу за складністю його методів. Якщо існує кілька методів із великою кількістю умов/гілок – WMC росте, код важче читати й тестувати. Рішення: ділити довгі методи на менші, виносити варіативну поведінку у стратегії/політики, прибирати дублювання.

Мета статті полягає у перетворенні принципів SOLID із декларативного набору правил у відтворювану інженерну процедуру, що забезпечує керовану еволюцію програмних систем шляхом кількісного опису впливу SOLID на когезію, зв'язність і стабільність архітектури, а також перевірки його застосування за допомогою метрик, контрактних тестів і архітектурних порогів. Це передбачає інтеграцію п'яти принципів (SRP, OCP, LSP, ISP, DIP) із базовими механізмами об'єктно-орієнтованого програмування, шаблонами проєктування та архітектурними стилями типу «Порти й адаптери», що дозволяє перейти від декларативного застосування SOLID до вимірюваного контролю архітектурної складності через побудову профілів метрик (CBO, LCOM, RFC, WMC), опис поведінкових контрактів та встановлення кількісних порогів абстракцій, забезпечуючи відтворюваність рішень, уникнення передчасних узагальнень і баланс між стабільністю та варіативністю системи.

Для досягнення поставленої мети в роботі вирішуються такі науково-прикладні завдання:

1. Проаналізувати фундаментальні принципи SOLID у контексті їх взаємозв'язку з об'єктно-орієнтованим програмуванням, архітектурними патернами та метриками якості програмного дизайну.
2. Розробити методологію кількісної оцінки впливу принципів SOLID на архітектурну складність із використанням метрик CBO, LCOM, RFC та WMC.
3. Побудувати операційний цикл застосування SOLID, що охоплює етапи спостереження, формування інваріантів, вибору патернів, формалізації контрактів, інверсії залежностей і корекції результатів.
4. Обґрунтувати ефективність гексагональної архітектури (Ports & Adapters) як практичної реалізації принципів DIP, ISP та SRP для підвищення модульності, керованості й відтворюваності програмних систем.

Очікуваним результатом реалізації зазначених завдань є формування відтворюваної методики застосування принципів SOLID як системи конструктивних обмежень, що забезпечує кількісно контрольовану еволюцію програмних архітектур, підвищує їхню стабільність, модульність і якість супроводження.

Методологічне обґрунтування. Перший крок – спостереження: збір профілю метрик (CBO, LCOM, RFC, WMC) для цільових модулів, побудова карт семантичної близькості та матриць змін коду за релізами [1,2,4,16]. Графіки, таблиці й діаграми на цьому етапі фіксують «гарячі зони», де зміни концентруються, і групи файлів, що еволюціонують разом.

Другий крок – формування інваріантів: визначення стабільних властивостей домену й зовнішніх контрактів, що не мають залежати від інфраструктури. Інваріанти описуються короткими специфікаціями й підкріплюються тестами, які в подальшому виконують роль «сигналізаторів» при рефакторингу.

Третій крок – вибір носія варіативності (патернів проектування): для поведінкових альтернатив – Стратегія; для фабрикації залежно від контексту – Фабрика/Абстрактна Фабрика; для ізоляції різних реалізацій інтерфейсів і стабілізації API – Міст і Адаптер; для поступового нарощування поведінки – Декоратор; для спрощення взаємодії з підсистемами – Фасад; для конструювання складних об'єктів – Білдер; для уніфікації фіксованої послідовності кроків із варіативними деталями – Шаблонний метод. На архітектурному рівні варіативність відмежовується через Порти та Адаптери та модульну архітектуру.

Четвертий крок – контракти: докладний опис передумов, післяумов та інваріантів для базових типів і їх підтипів. Для забезпечення LSP усі підтипи проходять однакові «контрактні тести», а нестабільні ієрархії обмежуються запаковуванням класу на рівні визначення або замінюються композицією.

DOI:10.31548/itees.2025.01.020
ISSN 3083-7502

П’ятий крок – інверсія залежностей із контролем прозорості: зв’язки встановлюються через абстракції у composition-root, забороняються приховані сервіс-локатори, граф контейнера перевіряється автоматично на цикли, дублікати і невпроваджені сервіси. Для кількісної оцінки пропонується узагальнений показник зв’язності (1), що враховує ін’єктовані залежності та розмір графа конфігурації [3, 5, 19, 20]:

$$DCBO = CBOstruct + \alpha \times CBOinjected + \beta \times |Econtainer|, \quad (1)$$

де α, β вагові коефіцієнти, які обираються емпірично за історією змін;

CBOstruct – структурне зчеплення класу з іншими типами, що видно зі статичного коду (виклики методів, поля, параметри/результати методів, наслідування/агрегація, generic-аргументи тощо);

CBOinjected – кількість *ін’єктованих* залежностей (через конструктор / властивість / метод), тобто портів, які підмінюються DI. Це «м’якші» зв’язки, тому ми зважуємо їх коефіцієнтом α;

|*Econtainer*| – кількість *конфігураційних ребер* DI-контейнера, що стосуються цього класу (зв’язки «порт-адаптер», фабричні реєстрації, тощо). Це «інфраструктурне» зчеплення, яке додає непрямі залежності, тому його зважуємо β.

Шостий крок – вимірювання та корекція: порівняння «до/після» за метриками, аналіз локальності змін при додаванні альтернатив, ревізія точок варіації [5,8,9,10]. Для OCP запроваджується поріг: абстракція вводиться лише за наявності принаймні двох реальних альтернатив і очікуваного приросту варіативності; для LSP – заборона нових перевизначень без контрактних тестів; для ISP – поділ інтерфейсів погоджується з реальними ролями клієнтів і спостереженнями змін коду; для DIP – обов’язкова валідація графа залежностей і звітність про DCBO.

Додатково рекомендується процесна дисципліна: ADR-записи для кожної точки варіації, семантичне версіонування адаптерів і плагінів, контроль глибин ієрархій, регулярна візуалізація показників у таблицях і графіках.

Результати та обговорення. Результати проведеного дослідження демонструють практичну значущість системного підходу до застосування принципів SOLID у поєднанні з метриками якості програмного коду. Узагальнення емпіричних спостережень і кількісних показників підтвердило, що збалансоване використання принципів дає змогу зменшити структурну зв’язність, підвищити когезію та стабільність архітектури без надмірного ускладнення її компонентної структури. Виявлені залежності між показниками метрик і типами архітектурних порушень свідчать про те, що SOLID виконує роль регулятора еволюційної динаміки системи, а його операційне застосування може бути формалізоване через об’єктивно вимірювані параметри. Практика показує, що SRP і ISP дієві тоді, коли рольова модель інтерфейсів і «осьова» відповідальність класів збігаються з реальними сценаріями використання. Некоректний масштаб призводить до «class explosion» або до «God object». Антипатерни SOLID узагальнено у табл. 1.

Таблиця 1 – Антипатерни SOLID

Антипатерн	Порушений принцип	Симптоми (метрики/ознаки)	Рефакторинги
God Class	SRP/ISP	Високі LCOM, fan-in/out	Виділення підмодулів, фасадів; SRP-декомпозиція
Cyclic Dependency	OCP/DIP	Цикли в графі, високе CBO	Розірвання циклів, введення портів/адаптерів
Feature Envy	SRP	Методи працюють з «чужими» даними	Переміщення методів, редизайн доменних меж
Hub-like Dependency	DIP	Один вузол із надмірним зв’язками	Розподіл відповідальностей, використання брокерів подій

Закономірність еволюції архітектури з ростом кількості релізів (рис. 3) така, що чим більше релізів, тим складніше змінювати код системи, якщо присутні типові архітектурні «запахи». Керувати балансом допомагає аналіз змін коду: методи й модулі, що часто змінюються разом, зазвичай належать до однієї відповідальності й мають бути зближені, а ті, що не перетинаються за сценаріями, варто розвести в різні контракти [11,12,18].

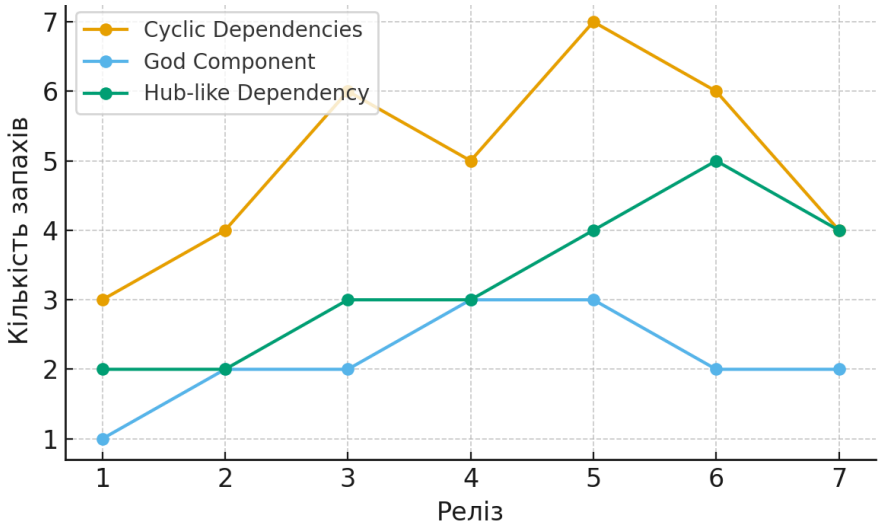


Рисунок 3 – Еволюція архітектурних «запахів» за релізами

OCP приносить вигоду, коли потік альтернатив достатній, щоб компенсувати вартість обслуговування абстракцій. Успішні кейси характеризуються стабільними інваріантами ядра, невеликими й чіткими інтерфейсами, прозорими точками розширення [11,13,12,14].

LSP часто порушується не на рівні сигнатур, а на рівні семантики: змінюються класи винятків, дозволені діапазони значень, політика обробки null, інваріанти стану. Такі зміни непомітні для компілятора, але руйнують підстановність класів-наслідників. Контрактні тести та обмеження успадкування там, де природних підтипів не передбачено, різко знижують ризик регресій [17,18].

DIP у поєднанні з DI зменшує жорстку зв'язаність і покращує замінність компонентів у тестах, проте створює ризики конфігураційної зв'язності: графи контейнера ускладняються, життєві цикли стають неочевидними, а трасування – дорогим. Протидією є composition-root, автоматична валідація графа, явні модулі, ліміти на «магічні» прив'язки та регулярний контроль DCBO. Результати узагальнено у табл. 2.

Таблиця 2 – Принципи SOLID, метрики та тактики рефакторингу

Принцип SOLID	Тип зв'язності / когезії	Релевантні метрики	Тактики рефакторингу
SRP	Концептуальна когезія	C3/CCC, LCOM	Декомпозиція класів, перейменування, виокремлення сервісів
OCP	Керовані точки варіації	Fan-in/out, цикли	Використання стратегій, модульної архітектури, ADR для варіацій
LSP	Поведінкова сумісність	Покриття контрактними тестами	Проектування за контрактами, property-based тест
ISP	Аферентна зв'язність	Fan-in, інтерфейси/ клієнти	Клієнтоорієнтовані інтерфейси, використання фасадів
DIP	Міжмодульна залежність	Граф залежностей, CBO	Порти/адаптери; використання ін'єкції залежностей

DOI:10.31548/itees.2025.01.020
ISSN 3083-7502

На архітектурному рівні порти та адаптери, а також плагінна архітектура масштабують принципи: ядро утримує інваріанти домену, а адаптери інкапсулюють інфраструктуру й канали взаємодії. Завдяки цьому зміни у спосіб доставки (GUI/API/CLI/черги) не вимагають модифікації бізнес-логіки. Вартість такого підходу – чітка контрактна дисципліна, тестування контрактів, каталогізація портів і контроль версій інтерфейсів.

Практичний протокол ухвалення рішень зводиться до простих правил. Спочатку – спостерігати гарячі точки й кластери змін коду, а потім – вводити абстракції у місцях реальної варіативності. Кожен підтип має бути з явно зафіксованим контрактом, кожна ін'єкція залежності – видима у composition-root, кожен інтерфейс – під реального клієнта. Графіки та таблиці стають інструментами комунікації: вони доводять або спростовують гіпотези про користь конкретних рішень, а діаграми демонструють рамки відповідальності та межі компонентів.

Висновки. SOLID працює як інженерна мова компромісів між стабільністю та варіативністю програмних систем, поєднуючи концептуальні принципи ООП із вимірюваними показниками якості архітектури. Його практичне значення полягає у створенні дисципліни проектування, що дозволяє підтримувати сталість логічних інваріантів системи, локалізувати варіативність і зменшувати ризики деградації структури коду під час еволюції продукту. У результаті дослідження визначено, що ефективність SOLID підвищується за умов використання формалізованих метрик (CBO, LCOM, RFC, WMC), контрактних тестів і архітектурних порогів, які дають змогу виявляти зони підвищеної складності та обґрунтовувати доцільність введення абстракцій.

Дотримання процесних порогів для OCP, поведінкової дисципліни для LSP, контрольованої інверсії залежностей для DIP/DI та рольового поділу для ISP забезпечує керовану еволюцію системи без втрати якості. Виявлені закономірності дозволяють розглядати SOLID як систему конструктивних обмежень, що створює баланс між гнучкістю і передбачуваністю змін. Використання гексагональної архітектури, патернів варіативності та системи метрик формує відтворюваний підхід до підтримки архітектурної цілісності, що має значний потенціал для подальших досліджень у сфері автоматизованого аналізу та валідації програмних рішень.

Список використаних джерел

1. Chidamber, S. R., & Kemerer, C. F. (1994). A metrics suite for object oriented design. IEEE Transactions on Software Engineering, 20(6), 476–493. https://doi.org/10.1109/32.295895.
2. Basili, V. R., Briand, L. C., & Melo, W. L. (1996). A validation of object-oriented design metrics as quality indicators. IEEE Transactions on Software Engineering, 22(10), 751–761. https://doi.org/10.1109/32.489317.
3. Briand, L. C., Daly, J. W., & Wüst, J. (1999). A unified framework for coupling measurement in object-oriented systems. IEEE Transactions on Software Engineering, 25(1), 91–121. https://doi.org/10.1109/32.748920.
4. Zhou, Y., & Leung, H. (2006). Empirical analysis of object-oriented design metrics for predicting high and low severity faults. IEEE Transactions on Software Engineering, 32(10), 771–789. https://doi.org/10.1109/TSE.2006.102
5. Arisholm, E., Briand, L. C., & Føyen, A. (2004). Dynamic coupling measurement for object-oriented software. IEEE Transactions on Software Engineering, 30(8), 491–506. https://doi.org/10.1109/TSE.2004.41.
6. Marcus, A., Poshyvanyk, D., & Ferenc, R. (2008). Using the conceptual cohesion of classes for fault prediction. IEEE Transactions on Software Engineering, 34(2), 287–300. https://doi.org/10.1109/TSE.2007.70768.
7. Revelle, M., Gethers, M. & Poshyvanyk, D. (2011). Using structural and textual information to capture feature coupling in object-oriented software. Empir Software Eng, 16, 773–811. https://doi.org/10.1007/s10664-011-9159-7.

8. Ajienka, N., & Capiluppi, A. (2017). Understanding the interplay between logical and structural coupling of software classes. Journal of Systems and Software, 134, 120–137. https://doi.org/10.1016/j.jss.2017.08.042.
9. Ajienka, N., Capiluppi, A., & Counsell, S. (2018). An empirical study on the interplay between semantic coupling and co-change. Empirical Software Engineering, 23(4), 1799–1836. https://doi.org/10.1007/s10664-017-9569-2.
10. Kagdi, H., Gethers, M. & Poshyvanyk, D. (2013). Integrating conceptual and logical couplings for change impact analysis in software. Empir Software Eng, 18, 933–969. https://doi.org/10.1007/s10664-012-9233-9.
11. Ampatzoglou, A., Frantzeskou, G., & Stamelos, I. (2012). A methodology to assess the impact of design patterns on software quality. Information and Software Technology, 54(4), 331-346. https://doi.org/10.1016/j.infsof.2011.10.006.
12. Yamashita, A., & Moonen, L. (2013). AiOLoS: A model for assessing organizational learning in software development organizations, 55(11), 1904-1924. https://doi.org/10.1016/j.infsof.2013.05.004.
13. Alfadel, M., Aljasser, K., & Alshayeb M. (2020). Empirical study of the relationship between design patterns and code smells. PLoS ONE, 15(4): e0231731. https://doi.org/10.1371/journal.pone.0231731.
14. Gnoyke, P., Schulze, S., & Krüger, J. (2024). Evolution patterns of software-architecture smells. Journal of Systems and Software, 213, 112170. https://doi.org/10.1016/j.jss.2024.112170.
15. Arisholm, E., Sjøberg, D. I. K., & Jørgensen, M. (2001). Assessing the changeability of two object-oriented design alternatives: A controlled experiment. Empirical Software Engineering, 6(3), 231–277. https://doi.org/10.1023/A:1011439416657.
16. Chae, H. S., Kwon, Y. R., & Bae, D. H. (2004). Improving cohesion metrics for classes. IEEE Transactions on Software Engineering, 30(8), 548–564. https://doi.org/10.1109/TSE.2004.88
17. Liskov, B., & Wing, J. (1994). A behavioral notion of subtyping. ACM Transactions on Programming Languages and Systems, 16(6), 1811–1841. https://doi.org/10.1145/197320.197383
18. Meyer, B. (1992). Applying "design by contract." Computer, 25(10), 40–51. https://doi.org/10.1109/2.161279.
19. Waseem, M., Liang, P., Shahin, M., Di Salle, A., & Márquez, G. (2021). Design, monitoring, and testing of microservices systems: The practitioners' perspective. Journal of Systems and Software, 182, 111061. https://doi.org/10.1016/j.jss.2021.111061.
20. Fregnan, E., Palomba, F., Bavota, G., Di Penta, M., Oliveto, R., & Lucia, A. D. (2019). On nonlinear Schrödinger equations with attractive inverse-power potentials. Information and Software Technology, 107, 159-178. https://doi.org/10.48550/arXiv.1903.04636.
21. Lampón, J. F., Cabanelas, P., & González-Benito, J. (2017). The impact of modular platforms on automobile manufacturing networks. Production Planning & Control, 28(4), 335–348. https://doi.org/10.1080/09537287.2017.1287442.
22. Pandremenos, J., Paralikas, J., Salonitis, K., & Chryssolouris, G. (2009). Modularity concepts for the automotive industry: A critical review. CIRP Journal of Manufacturing Science and Technology, 1(3), 148–152. https://doi.org/10.1016/j.cirpj.2008.09.012.
23. Bao, Z., Laovisutthichai, V., Tan, T., Wang, Q., & Lu, W. (2022). Design for manufacture and assembly (DfMA) enablers for offsite interior design and construction. Building Research & Information, 50(3), 325–338. https://doi.org/10.1080/09613218.2021.1966734.

Kornilov Ivan
Assistant, Department of Computer Science,
National University of Life and Environmental Sciences of Ukraine
ORCID: https://orcid.org/0009-0009-5598-2690
E-mail: i.kornilov@nubip.edu.ua

DOI:10.31548/itees.2025.01.020
ISSN 3083-7502

Weigang Ganna
Candidate of Engineering Sciences, Associate Professor, Associate Professor of the Department of Computer Science,
National University of Life and Environmental Sciences of Ukraine
ORCID: https://orcid.org/0000-0002-2082-2322
E-mail: weigang.ganna@nubip.edu.ua

SOLID AS A SYSTEM OF CONSTRUCTIVE CONSTRAINTS IN SOFTWARE ARCHITECTURE DESIGN

Abstract. *The article presents SOLID as a system of constructive constraints that disciplines the degrees of freedom in design and transforms the evolution of software systems into a controlled process. It explains the relationship of the principles with the foundations of object-oriented programming, design patterns, and architectural styles. Non-obvious effects are examined, including premature abstractions under OCP, hidden configuration coupling under DIP/DI, class explosion and responsibility fragmentation under SRP/ISP, and semantic violations of LSP that are not captured by type signatures. An operational approach to decision validation is proposed through metrics, contract-based tests, and threshold controls for introducing abstractions, along with practical decision-making protocols.*

Keywords: *SOLID; Object-Oriented Programming; Software Architecture; Design Patterns; Cohesion; Coupling; Code Quality Metrics; Dependency Inversion (DI/IoC).*

УДК 004.62

Золотуха Роман Андрійович
доктор філософії, старший викладач кафедри інформаційних систем і технологій,
Національний університет біоресурсів і природокористування України
ORCID: https://orcid.org/0000-0003-3099-722X
E-mail: r.zolotukha@nubip.edu.ua

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ АВТОМАТИЗАЦІЇ ОБРОБКИ РЕЗЮМЕ КАНДИДАТІВ ДЛЯ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ПРОЦЕСУ ФОРМУВАННЯ ІТ-КОМАНД

Анотація. *Стаття присвячена розробці інформаційної технології автоматизованої обробки резюме кандидатів у форматі PDF з використанням мови програмування Python. Представлено підхід до вилучення, структуризації та подальшого аналізу даних із застосуванням бібліотек pdfplumber, spaCy та pandas. Запропонований модуль дозволяє визначати ключові елементи резюме, зокрема освіту, навички, контактну інформацію та досвід роботи, з подальшим формуванням структурованих даних у форматі JSON. Особливу увагу приділено забезпеченню універсальності алгоритму для резюме з довільною структурою та україномовним контентом. У роботі розглянуто основні етапи реалізації програмного рішення, наведено діаграми потоків даних, схеми обробки PDF-файлів та приклади юніт-тестування функцій системи. Розроблена технологія може бути використана для автоматизації первинного етапу рекрутингу та інтеграції з HR-аналітичними системами, що підвищує точність і швидкість обробки кандидатських даних у процесі формування ІТ-команд.*

Ключові слова: *інформаційна технологія, Python, PDF-резюме, автоматизація рекрутингу, spaCy, NLP, HR-система.*

Актуальність. У сучасних умовах динамічного розвитку ІТ-галузі питання ефективного підбору персоналу набуває стратегічного значення. Різке збільшення кількості кандидатів на одну вакансію після 2022 року [1], зростання конкуренції та дефіцит висококваліфікованих спеціалістів зумовлюють необхідність підвищення швидкості та точності процесу відбору. Зокрема, автоматизація первинної обробки резюме кандидатів дозволяє зменшити навантаження на HR-відділи та знизити ризик суб'єктивності при прийнятті рішень.

Однією з ключових проблем сучасних рекрутингових систем є відсутність інструментів для якісного вилучення та аналізу даних з резюме у форматі PDF, особливо тих, що складені українською або двомовною структурою. Більшість існуючих платформ, таких як LinkedIn Recruiter чи Greenhouse, орієнтовані на англомовний ринок і не враховують специфіку локальних форматів документів, структури резюме та різноманіття форматування. Це ускладнює інтеграцію таких документів у бази даних HR-систем та подальший аналітичний обробіток.

Мета дослідження полягає у розробці інформаційної технології автоматизованої обробки резюме кандидатів у форматі PDF, що забезпечує вилучення, структуризацію та подальший аналіз даних із використанням бібліотек Python, з метою підвищення ефективності процесу підбору персоналу та формування ІТ-команд.

Аналіз останніх досліджень та публікацій. У численних роботах останніх років показано важливість автоматизації обробки документів та витягування структурованої інформації із PDF-форматів. У роботі Chafiq N., Ghazouani M. та El Gounidi R. [2] запропоновано систему автоматизованої обробки резюме для вступу до магістерських програм, яка ґрунтується на використанні методів обробки природної мови (NLP). Автори застосували попередньо натреновані моделі spaCy та Hugging Face Transformers для розпізнавання сутностей (Named Entity Recognition — NER) і вилучення таких ключових елементів, як освіта, досвід та навички кандидатів. Додатково реалізовано двоетапне узагальнення тексту резюме – екстрактивне (на основі моделей BERT) та абстрактивне (з використанням мовних моделей LLAMA). Система продемонструвала високу ефективність, досягнувши точності NER 82 % та середнього часу обробки одного резюме 3,84 секунди. Ця

DOI:10.31548/itees.2025.01.030
ISSN 3083-7502

робота показує перспективність поєднання класичних NLP-підходів із сучасними трансформерними архітектурами для обробки великих масивів документів у форматі PDF.

Дослідження Sandanayake T. C., Limesha G. A. I., Madhumali T. S. S., Mihirani W. P. I. та Peiris M. S. A. [3] зосереджене на автоматичному аналізі та ранжуванні резюме кандидатів для підбору персоналу в IT-сфері. Запропонований авторами інструмент витягує релевантну інформацію з неструктурованих текстів резюме та формує рейтинг кандидатів відповідно до заданих критеріїв. Особливістю цього підходу є інтеграція зовнішніх джерел даних, таких як Stack Overflow, GitHub та професійні блоги, для створення повного профілю кандидата. Розроблена система орієнтована на вакансії в галузі інформаційних технологій, що дозволяє значно зменшити час ручного перегляду резюме та підвищити точність відбору.

У роботі Ahmed F., Anannya M., Rahman T. та Khan R. T. [4] розглянуто можливість поєднання автоматизованої обробки резюме з психометричним аналізом у процесі підбору кадрів. Автори запропонували концепцію соціальної мережі для шукачів роботи та роботодавців, яка автоматично зіставляє кандидатів із вакансіями за заданими критеріями. Система враховує результати психометричних тестів для визначення відповідності особистісних якостей кандидата вимогам компанії, що, на думку дослідників, підвищує точність відбору й рівень задоволеності працівників після працевлаштування.

Ben Azzou K. та Talei H. [5] запропонували машинно-навчальний підхід до автоматизованого аналізу даних резюме та визначення профілю кандидата. Розроблена ними система використовує методи NLP для вилучення з текстів резюме структурованих даних про освіту, досвід та навички, після чого застосовує алгоритми класифікації для автоматичного зіставлення кандидатів із відповідними посадами. Автори підкреслюють, що такий підхід дає змогу істотно скоротити навантаження на HR-відділи та знизити суб’єктивність відбору завдяки стандартизованій оцінці текстових даних.

Проведений аналіз свідчить, що сучасні дослідження у сфері автоматизації обробки резюме орієнтовані на поєднання методів обробки природної мови, машинного навчання та інтеграції зовнішніх джерел даних. Спільною тенденцією є прагнення зменшити трудомісткість і суб’єктивність процесу відбору кандидатів за рахунок впровадження інтелектуальних алгоритмів, здатних опрацьовувати великі обсяги неструктурованої інформації. Разом із тим більшість розробок сфокусовані на англомовному ринку, що зумовлює актуальність дослідження систем, адаптованих до україномовних резюме у форматі PDF.

Матеріали і методи дослідження. Для тестування алгоритмів сортування та демонстрації проблеми локалізації важливо мати реалістичний набір тестових даних. Вибірку досліджуваних склали студенти 1 курсу факультету інформаційних технологій Національного університету біоресурсів і природокористування України. До експерименту було залучено 4 групи студентів ІПЗ-2301, ІПЗ-2302, ІСТ-2301, ІСТ-2302 (Рисунок 1). За планом експерименту для застосування алгоритму - кожен учасник дослідження надіслав своє резюме у PDF форматі для подальшої його обробки та формування команд. Для чистоти експерименту резюме були заповнені у вільному форматі без запропонованого шаблону.

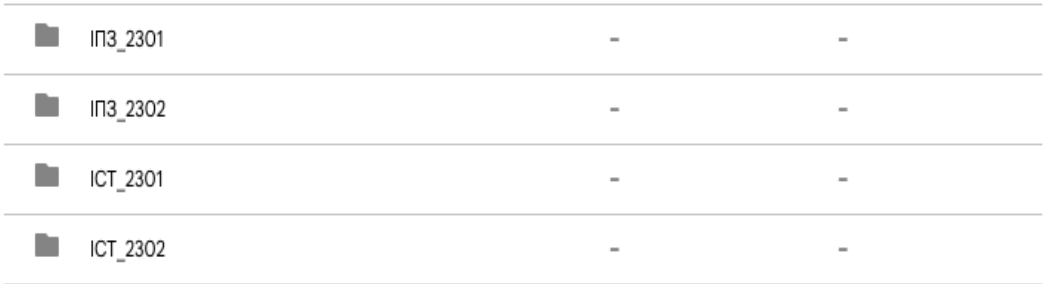
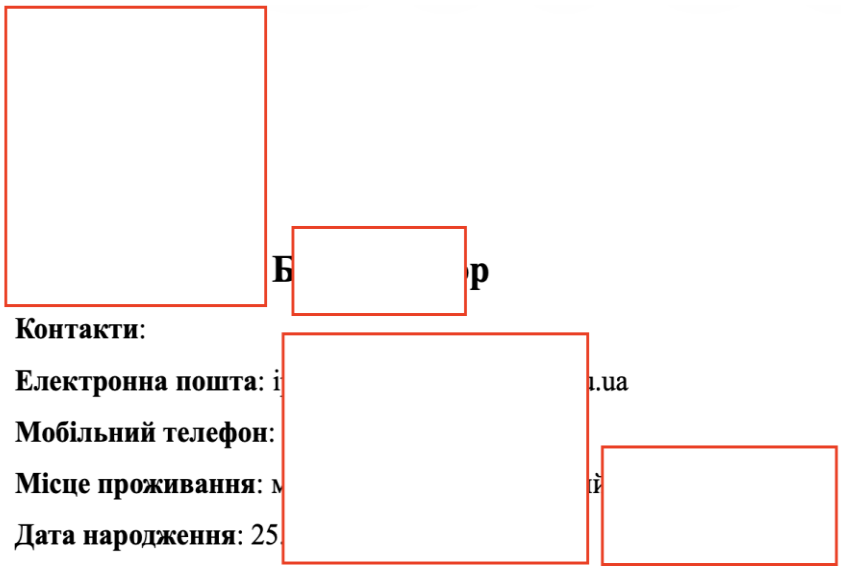


Рисунок 1 – ZIP-архів з резюме учасників експерименту

На рис. 2 продемонстровано резюме у PDF форматі одного з учасників експерименту. Усі поля з персональними даними навмисне зафарбовані для збереження конфіденційності даних.



Контакти:

Електронна пошта: .ua

Мобільний телефон:

Місце проживання:

Дата народження: 25.

Мови:

- Українська мова – рідна
- Англійська мова – середній рівень.

Досвід роботи: відсутній.

Освіта: НУБіП України, Факультет Інформаційних технологій,
Спеціальність: Інженерія програмного забезпечення

Hard-skills: робота з текстом, зображенням, монтаж відео, навички роботи з технічною частиною ПК.

Soft-Skills: середньовиражені навички лідерства, високий рівень комунікабельності, навички праці в команді, високий рівень стресостійкості.

Рисунок 2 – Приклад резюме у форматі PDF з тестової вибірки

Результати дослідження та їх обговорення. Для розробки модулю була розроблена модель потоків даних процесу подачі резюме кандидатом (рисунок 3). Кандидат подає заявку на вакансію, прикріплюючи резюме у форматі PDF. Резюме обробляється системою і виділяє ключову інформацію про кандидата: контактну інформацію, місце навчання та навички. Ця інформація структурується і вноситься в базу даних HR, де HR-спеціаліст може бачити потрібних кандидатів, використовуючи задані фільтри.

Визначені вимоги до інформаційної технології обробки даних з PDF резюме кандидатів відображено на діаграмі варіантів використання (рис. 4).

Для реалізації даного алгоритму ми використали універсальні можливості мови програмування Python. Надійна екосистема бібліотек Python полегшила наші зусилля в розборі та вилученні релевантної інформації зі складної структури резюме кандидатів.

DOI:10.31548/itees.2025.01.030
ISSN 3083-7502

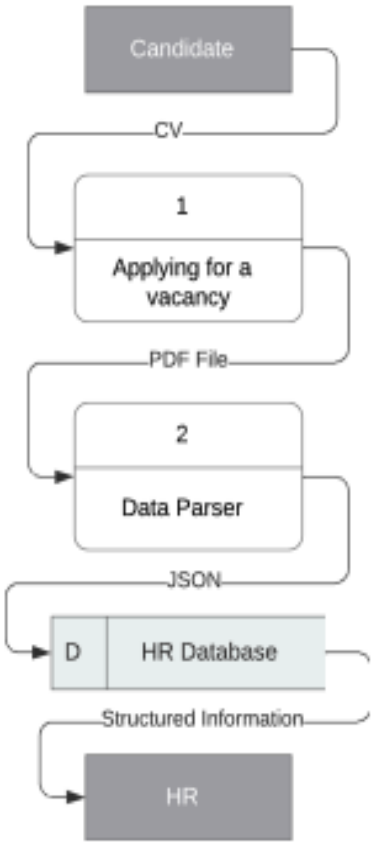


Рисунок 3 – Схема руху даних від кандидата до HR

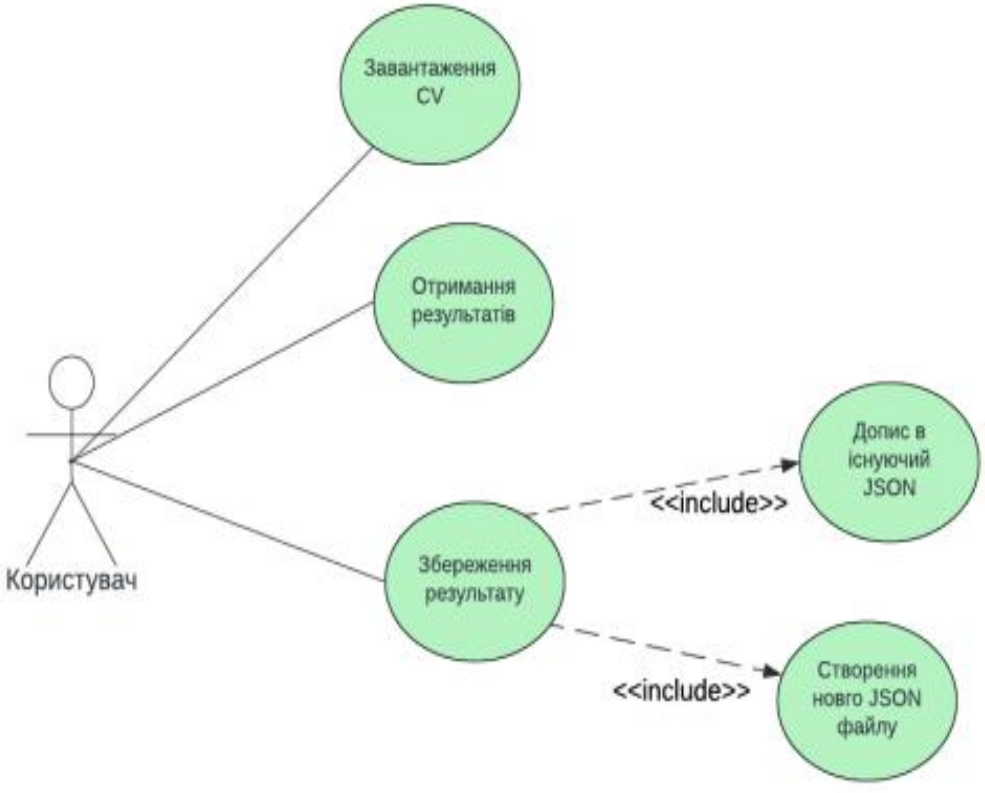


Рисунок 4 – Діаграма варіантів використання для користувача

Щоб розібратися в тонкощах PDF-документів, ми використовували бібліотеку "PDFplumber", яка дозволила нам точно витягувати текстовий контент. Ця бібліотека надає можливість переглядати макет PDF-резюме та виокремлювати необхідні текстові сегменти для подальшого аналізу. Для обробки природної мови та розпізнавання текстових шаблонів ми використовували бібліотеку "spacy". Цей потужний інструмент НЛП дозволив токенізувати, тегувати та аналізувати текст, що дало нам змогу виявити шаблони та сутності, важливі для виокремлення навичок та освіти. Модуль "Matcher" у складі "spacy" допоміг виявити конкретні лінгвістичні патерни, впорядкувавши наш процес визначення ключової інформації. Щоб полегшити організацію та зберігання наших результатів, ми використали модуль "csv" для створення та управління структурованими наборами даних. Бібліотека "pandas" запропонувала нам ефективний засіб для маніпулювання та аналізу цих наборів даних, що дозволило нам отримати уявлення та тенденції з отриманої інформації. Як невід'ємну частину нашої реалізації ми використали можливості вбудованої бібліотеки Python під назвою "json".

Запропонований нами процес перетворення PDF документу, що містить інформацію про кандидата в JSON файл готовий до подальшого аналізу можна візуалізувати наступним чином. (рис. 5).

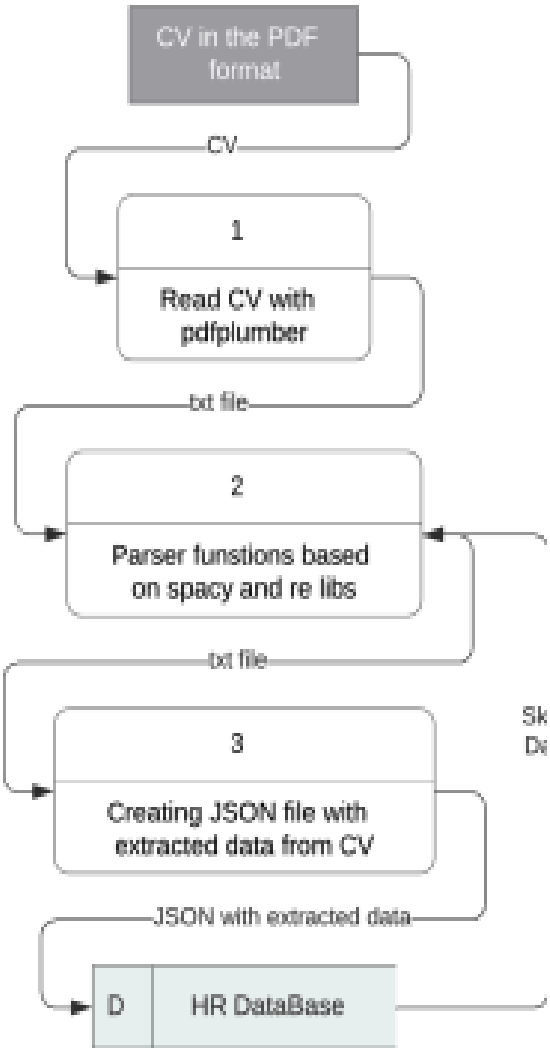


Рисунок 5 – Схема реалізації модулю обробки резюме в PDF форматі

Результатом роботи нашого алгоритму є ретельно структурований JSON-файл, зображений на рисунку нижче (рис. 6).

DOI:10.31548/itees.2025.01.030
ISSN 3083-7502

```
if __name__ == '__main__':
    resume_text = extracted_text

    name = extract_name(resume_text)
    if name:
        print("Name:", name)
    else:
        print("Name not found")

    contact_number = extract_contact_number_from_resume(resume_text)
    if contact_number:
        print("Contact Number:", contact_number)
    else:
        print("Contact Number not found")

    email = extract_email_from_resume(resume_text)
    if email:
        print("Email:", email)
    else:
        print("Email not found")

    skills_list = csv_data_list
    extracted_skills = extract_skills_from_resume(resume_text, skills_list)
    if extracted_skills:
        print("Skills:", extracted_skills)
    else:
        print("No skills found")

    extracted_education = extract_education_from_resume(resume_text)
    if extracted_education:
        print("Education:", extracted_education)
    else:
        print("No education information found")
```

```
Name:
Contac
Email:
Skills
p', 'api', 'mailchimp', 'economics', 'gmail', 'automation', 'pytho
n', 'subscribe', 'analytics', 'google analytics', 'firebase', '2014', 'communication', 'improvement', '.com', 'engl
ish', 'travel', 'british', 'data collection', 'web', 'digital', 'navigation', 'com', 'facebook', 'russian', 'analyt
```

Рисунок 6 – Реалізований модуль в середовищі Jupyter Notebook

Для технічної валідація реалізованого рішення було підготовлено серія unit-тестів. На рис. 7 наведений код для тестування однієї з функції інформаційної технології – обробки тексту з PDF файлів у модулі обробки PDF резюме кандидатів.

```
import unittest
import pdfplumber
from io import BytesIO

def extract_text_from_pdf(pdf_path):
    with pdfplumber.open(pdf_path) as pdf:
        full_text = ""
        for page in pdf.pages:
            text = page.extract_text()
            full_text += text
    return full_text

class TestPDFExtraction(unittest.TestCase):

    def setUp(self):
        self.pdf_content = b'%PDF-1.4\n1 0 obj\n<< /Type /Catalog /Pages 2 0
        self.pdf_path = "test.pdf"
        with open(self.pdf_path, "wb") as f:
            f.write(self.pdf_content)

    def tearDown(self):
        import os
        os.remove(self.pdf_path)

    def test_extract_text(self):
        expected_text = "Hello, world!\n"
        extracted_text = extract_text_from_pdf(self.pdf_path)
        self.assertEqual(extracted_text.strip(), expected_text.strip())

if __name__ == "__main__":
    unittest.main()
```

Рисунок 7 – Юніт-тест для функції обробки тексту з PDF-файлу

Даний тест складається з функції extract_text_from_pdf, яка витягує текст з PDF-файлу, та тестового класу TestPDFExtraction, який перевіряє правильність роботи цієї функції. Цей тест складається з 4 етапів: створення тимчасового PDF-файлу; обробка тексту з PDF-файлу за допомогою функції extract_text_from_pdf; перевірка тексту на відповідність очікуваному тексту; видалення тимчасового PDF-файлу після завершення тестування. За допомогою юніт-тестування для модуля обробки резюме у форматі PDF вдалось верифікувати функції: обробки тексту з PDF-файлів; функцій, що відповідають за обробку тексту за ключовими словами в резюме кандидатів; функцію збереження структурованих даних у форматі JSON; стандартизацію структури збережених даних.

Висновок. За допомогою реалізованої інформаційної технології автоматизації обробки резюме кандидатів ми отримали поля які нас цікавлять у резюме: ПІБ, контактний номер телефону, контактний Email кандидата, навички кандидата, освіта кандидата. Алгоритм обробив 98 резюме у довільному форматі і показав ефективність у 96%. Як показало дослідження, для реалізованого алгоритму не потрібно готувати конкретний шаблон резюме. Проте, було виявлено і ряд обмежень, зокрема, коли кандидати використовувати англіцизми українською. В цьому контексті додавання слів виключень би стало одним із варіантів покращення даного алгоритму. Отриману інформацію з CV в подальшому можна імпортувати в JSON форматі у зручне сховище HR-бази, яке іт-компанії використовують під час процесу рекрутингу.

Список використаних джерел

1. Zolotukha, R. A., & Hlazunova, O. H. (2023). Prohnozuvannia rozvytku rynku pratsi v IT haluzi Ukrainy metodom chasovykh riadiv [Forecasting the development of the labor market in the IT industry of Ukraine using time series methods]. In Interdisciplinary research: Scientific horizons and perspectives: Proceedings of the VI International Scientific and Theoretical Conference (pp. 31–36). Vilnius, Lithuania.
2. Chafiq, N., Ghazouani, M., & El Gounidi, R. (2025). From manual review to AI automation: An NLP-powered system for efficient CV processing in academic admissions. LatIA, 3, Article 315. https://doi.org/10.62486/latia2025315.
3. Sandanayake, T. C., Limesha, G. A. I., Madhumali, T. S. S., Mihirani, W. P. I., & Peiris, M. S. A. (2020). Automated CV analyzing and ranking tool to select candidates for job positions. In Proceedings of the ACM/IEEE International Conference on Automated Software Engineering. ACM. https://doi.org/10.1145/3301551.3301579.
4. Ahmed, F., Anannya, M., Rahman, T., & Khan, R. T. (2015). Automated CV processing along with psychometric analysis in job recruiting process. In 2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT). IEEE. https://doi.org/10.1109/ICEEICT.2015.7307521.
5. Ben Azzou, K., & Talei, H. (2024). A machine learning approach for automated CV data analysis and job profile identification. In 2024 Sixth International Conference on Intelligent Computing in Data Sciences (ICDS). IEEE. https://doi.org/10.1109/ICDS62089.2024.10756435.

Zolotukha Roman
PhD, Senior Lecturer, Department of Information Systems and Technologies,
National University of Life and Environmental Sciences of Ukraine
ORCID: https://orcid.org/0000-0003-3099-722X
E-mail: r.zolotukha@nubip.edu.ua

INFORMATION TECHNOLOGIES FOR AUTOMATING THE PROCESSING OF CANDIDATE CV TO INCREASE THE EFFICIENCY OF IT TEAM FORMATION

Abstract. *The article is devoted to the development of an information technology for automated processing of candidate CV in PDF format using the Python programming language. The approach to extracting, structuring, and*

DOI:10.31548/itees.2025.01.030
ISSN 3083-7502

further analyzing data with the use of the pdfplumber, spaCy, and pandas libraries is presented. The proposed module enables the identification of key resume elements, including education, skills, contact information, and work experience, followed by the formation of structured data in JSON format. Special attention is given to ensuring the universality of the algorithm for CV with arbitrary structure and Ukrainian-language content. The paper describes the main stages of implementing the software solution, including data flow diagrams, PDF processing schemes, and examples of unit testing of system functions. The developed technology can be used to automate the initial stage of recruitment and integrate with HR analytics systems, thereby improving the accuracy and speed of candidate data processing in the IT team formation process.

Keywords: *information technology, Python, CV, recruitment automation, spaCy, NLP, HR system*.

УДК 004.42:004.8

Недьошев Максим Владиславович
аспірант,
Національний університет біоресурсів і природокористування України
ORCID: https://orcid.org/0009-0000-9820-0649
E-mail: nedoshev@pm.me

Кириченко Віктор Вікторович
кандидат технічних наук, доцент, доцент кафедри комп'ютерних наук,
Національний університет біоресурсів і природокористування України
ORCID: https://orcid.org/0009-0001-0575-8684
E-mail: v.kyrychenko@nubip.edu.ua

ДОСЛІДЖЕННЯ ВПЛИВУ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ НА РОЗРОБКУ ВЕБСАЙТІВ З ВИКОРИСТАННЯМ ФРЕЙМВОРКУ VUE

Анотація. *У цій статті досліджується трансформаційний вплив великих мовних моделей (LLM) на сучасну компонентно-орієнтовану веброзробку, використовуючи фреймворк Vue.js як конкретний приклад. Синтезуючи результати широкого емпіричного дослідження розробки програмного забезпечення за допомогою LLM [1], ми аналізуємо парадигматичний зсув від нативних робочих процесів фреймворку до процесів, доповнених LLM. Аналіз охоплює весь життєвий цикл проєкту, виявляючи значне підвищення продуктивності на етапах реалізації та розгортання, зокрема у створенні компонентів, автоматизації тестування та налаштуванні інфраструктури. Однак ці переваги врівноважуються критичними викликами, серед яких занепокоєння щодо надійності коду, увічнення конфліктів версій через застарілі навчальні дані та ризик когнітивного розвантаження серед розробників. Ми стверджуємо, що інтеграція LLM переосмислює роль старшого розробника, перетворюючи його з основного генератора коду на експерта-валідатора та архітектурного наглядача. Стаття завершується окресленням ключових ризиків та пропозицією напрямків для майбутніх досліджень, наголошуючи на необхідності створення специфічних для фреймворку бенчмарків для оцінки якості коду, згенерованого ШІ, та лонгітюдних досліджень щодо супроводжуваності Vue.js-застосунків, розроблених за допомогою LLM.*

Ключові слова: *великі мовні моделі, Vue.js, веброзробка, інтелектуальна автоматизація, UI-компоненти, продуктивність, програмна інженерія.*

Актуальність. Веброзробка відіграє велику роль у поточній економіці світу, де всі сервіси і інструменти доступні з більшості гаджетів за пошуковим запитом. Стрімкий розвиток штучного інтелекту, у вигляді Large Language Models. Протягом останніх років інструменти, які базуються на ШІ, як GPT, Github Copilot, Claude та інші, перейшли з розряду експериментальних у категорію практичних інструментів, які щоденно використовуються розробниками. Це призвело до зсуву парадигми розробки до LLM-орієнтованої розробки, що значно відрізняється від традиційних способів розробки. Сфера застосування LLM є відносно новою і є безліч прогалин у систематизованих знаннях. Має сенс дослідити такі ключові фактори: економічна доцільність, ступінь підвищення продуктивності, трансформацію ринку праць, наявність проблем та ризиків використання LLM у розробці. Враховуючи різноманітну природу розробки, яка різниця підходами до розробки, мовами програмування та фреймворками є необхідність систематизувати знання про вплив інтеграції LLM у розробку специфічних доменів і задач.

Мета дослідження – систематично проаналізувати та оцінити вплив великих мовних моделей (LLM) на процеси розробки вебсайтів з використанням фреймворку Vue.js. Оскільки LLM стають все більш інтегрованими в інструменти розробки, виникає нагальна потреба зрозуміти, як ця нова парадигма відрізняється від традиційних підходів до розробки на прикладі веброзробки з використанням Vue.js.

Аналіз матеріалів досліджень за напрямком роботи. Тема інтеграції великих мовних моделей у процеси розробки програмного забезпечення є предметом активних досліджень в академічній та індустріальній спільнотах. Аналіз наукових публікацій дозволяє виділити

DOI:10.31548/itees.2025.01.038
ISSN 3083-7502

кілька ключових напрямків. Перший напрямок стосується емпіричного порівняння продуктивності розробників при використанні LLM-інструментів. Дослідження [1] виявили значні відмінності між традиційними Low-Code платформами та LLM-орієнтованим підходом. Якщо LCP ефективні у вузькоспеціалізованих завданнях (наприклад, створення форм чи простих бізнес-додатків), то LLM демонструють значно вищу гнучкість, охоплюючи ширший спектр програмних завдань, включаючи веб-розробку, аналіз даних та створення алгоритмів. Другий напрямок фокусується на якості та надійності коду, згенерованого LLM. Хоча такі інструменти, як GitHub Copilot, можуть генерувати синтаксично правильний код, роботи [2, 3] вказують на потенційні проблеми, такі як "галюцинації" (генерація неіснуючих функцій або API), вразливості безпеки та неоптимальні алгоритмічні рішення. Це підкреслює критичну роль людини-експерта в процесі перевірки, рефакторингу та валідації коду. Третій напрямок досліджень вивчає вплив LLM на освіту та підготовку майбутніх програмістів. Роботи, такі як [3], аналізують, як використання LLM змінює навчальний процес. З одного боку, вони можуть слугувати потужними інструментами для навчання, надаючи миттєві пояснення та приклади коду. З іншого боку, існує ризик, що студенти стануть надто залежними від цих інструментів, що може негативно вплинути на розвиток фундаментальних навичок розв'язання проблем та алгоритмічного мислення. Четвертий напрямок стосується майбутнього програмної інженерії. У звітах та прогнозах від провідних аналітичних компаній, таких як McKinsey & Company [4], підкреслюється, що ШІ не замінить розробників, а трансформує їхню роль. Очікується, що до 80% рутинних завдань кодування будуть автоматизовані, що дозволить інженерам зосередитися на більш творчих та стратегічних аспектах: архітектурі системи, проєктуванні користувацького досвіду та інноваціях. Нарешті, активно обговорюються етичні аспекти та виклики, пов'язані з використанням LLM у розробці. Питання конфіденційності (оскільки фрагменти коду можуть надсилатися на сторонні сервери для аналізу), авторського права (через навчання моделей на мільярдах рядків відкритого коду) та упередженості (моделі можуть відтворювати помилки та погані практики, присутні в навчальних даних) описані у працях [5, 6]. Таким чином, існуючий масив досліджень підтверджує значний та багатогранний вплив LLM на розробку програмного забезпечення і роботу людини загалом. Однак, багато питань залишаються відкритими, зокрема щодо довгострокових наслідків цієї технологічної революції та розробки найкращих практик для ефективної та безпечної інтеграції LLM у робочі процеси.

Матеріали і методи дослідження. Метою даного дослідження є комплексний аналіз та систематизація знань про вплив великих мовних моделей (LLM) на процеси розробки вебсайтів. Для досягнення поставленої мети було обрано методологію систематичного огляду літератури (Systematic Literature Review, SLR**)**. Інформаційною базою для дослідження слугували наукові та технічні публікації, індексовані у провідних наукометричних базах даних та цифрових бібліотеках, таких як: архів препринтів arXiv.org, Google Scholar. Також використовувались інструменти для пошуку актуальних досліджень з 2020 по 2025 роки Google NotebookLM, Alphaxiv, Research Rabbit, x.com. Пошук відбувався англійською мовою, а пошукові запити включали: Large Lanugage Model, LLM, software engineering, web development, code generation, challenges, opportunities, Github Copilot, AI, Low-code. До аналізу брались джерела з наукових журналів, конференцій, технічні та економічні звіти, публікації присвячені використанню LLM у програмній інженерії та веброзробці. З аналізу були виключені оглядові статті, маркетингові матеріали та блоги без технічного обґрунтування. До методів аналізу можна віднести тематичний аналіз, порівняльний аналіз та синтез даних.

Об’єктом дослідження обрано розробку вебсайтів з використанням штучного інтелекту конкретно з використанням фреймворку Vue.js. Оскільки LLM навчаються на великих об’ємах даних, для розробників є сенс використовувати найбільш популярні інструменти, про які багато інформації у інтернеті. Однак, існує безліч продуктів, які використовують новіші та менш поширені інструменти. Наприклад, згідно аналізу використаних інструментів для створення вебсайтів [7] Vue.js займає далеко не перше місце по використанню, а отже можна дійти висновку що прикладів коду для навчання також невелике, що має створювати

складності для LLM. Такий підхід до аналізу використання LLM у розробці дає більш точніші висновки щодо використання ШІ технологій для роботи у спеціалізованому домені.

Трансформація життєвого циклу веброзробки. Традиційний процес веб розробки включає створення компонентів, управління станом (state management), маршрутизацію та тестування, зазнає значних змін під впливом LLM:

1. Етап ініціалізації та налаштування проєкту**.** На цьому етапі LLM виступають як інтелектуальні асистенти, що генерують конфігураційні файли для збирачів проєкту (Vite, Webpack), налаштовують інтеграцію з TypeScript, ESLint, Prettier та створюють базову структуру каталогів. Це значно скорочує час, який раніше витрачався на рутинні операції, дозволяючи розробнику швидше перейти до бізнес-логіки. Хоча цей процес вже був автоматизований за допомогою консольних інструментів (CLI), використання LLM замінює використання шаблонів (boilerplate) на інтерактивну генерацію стартових проєктів. Для цього можна використовувати сервіси v0 або bolt.new.
2. Розробка UI-компонентів відчуває найбільший вплив LLM. Замість ручного кодування, розробник може сформувати запити природною мовою. Це значно підвищує продуктивність: час виконання завдань може зменшитись на 0,8 стандартних відхилень, а якість результату зрости на 0,4 стандартних відхилення [6].
3. Тестування та зневадження. Написання юніт-тестів та компонентних тестів (наприклад, з використанням Vitest та Vue Testing Library) є одним з найбільш трудомістких і часозатратних процесів. LLM здатні автоматично генерувати тестові сценарії, мокувати залежності та описувати базові перевірки, що значно підвищує тестове покриття. Проте, згенеровані тести не завжди охоплюють усі граничні випадки, що вимагає ретельного аудиту з боку розробника. Дослідження безпеки коду, згенерованого ШІ, вказують на те, що він може містити до 40% вразливостей, що підкреслює критичну необхідність перевірки [2].

Ключові виклики та ризики у контексті Vue.js. Незважаючи на очевидні переваги, інтеграція LLM у розробку на Vue.js породжує специфічні виклики:

1. Надійність та ідіоматичність коду. LLM, навчені на величезному масиві коду, не завжди розрізняють ідіоматичний (відповідний "духу" фреймворку) та функціональний, але неоптимальний код. Для Vue.js це може проявлятися у неправильному використанні системи реактивності, ігноруванні можливостей Composition API або генерації коду, що призводить до невиправданих повторних рендерів компонентів.
2. Конфлікти версій. Екосистема Vue.js нещодавно зазнала значних змін, а саме: перехід з Vue 2 на Vue 3, еволюція від Vuex до Pinia, Webpackt до Vite. LLM, навчені на застарілих даних, можуть генерувати код, несумісний з останніми версіями фреймворку та його бібліотек. Це створює "прихований технічний борг" і вимагає від розробника постійної пильності та глибоких знань актуального стану екосистеми.
3. Моделі можуть "вигадувати" (галюцинації) неіснуючі функції або властивості API Vue.js, що призводить до помилок під час виконання і вимагає додаткового часу на зневадження. Ця проблема детально описана в систематичних оглядах LLM у програмній інженерії [1].

Глобальне тестування LLM для генерації коду для Vue.js. Для тестування роботи LLM у задачах різного типу у дослідженні [8] було створено LLM-Bench. У даному дослідженні Gemeni 2.5 pro, Claude, GPT4.1 LLM показали найкращі результати у задачах генерації коду. Однак, відсоток вирішних задач менший за 50%. У контексті задач специфічних для Vue.js в дослідженні було описано 20 задач типових для Vue та 20 задач для Nuxt, які можуть зустрічатися під час розробки і розподілені на 3 категорії: “easy”, “moderate” та “challenging”. Так, наприклад, для Gememi 2.5 pro було виконано всього 45% завдань з другої спроби і лише 20% з першої спроби.

Специфічне тестування. Для розширення тестування LLM для специфічної задачі в дослідженні було розроблено два тести: генерація таблиці користувачів без та з використанням бібліотек готових компонентів. Використання бібліотеки компонентів робить задачу для LLM більш специфічною та складною, однак це широка поширена задача на

DOI:10.31548/itees.2025.01.038
ISSN 3083-7502

практиці, оскільки більшість вебзастосунків використовують бібліотеки компонентів. Для тестів було використано бібліотеку компонентів Vuestic UI, як приклад нової та не самої популярної бібліотеки, що має малу кількість прикладів коду, хоча має велику за обсягом документацію.

Так було описано запит для LLM без конкретного запиту для використання бібліотеки:

```
Make Table.vue component.
<columns> Avatar, Name, Phone, Email, Edit Button, Delete Button. </columns> Fill
with mock data. Show modal confirmation on delete.
Freeze Avatar and Name column.
```

Цей запит було дано на обробку LLM у хмарі Google Gemini 2.5 Prop, Google Gemini 2.5 Flash, Sonnet 4 та локальним LLM Qwen3 4b, Google Gemma 12b. Результати тестів представлено у табл. 1. Критеріями тестування є ручний аналіз згенерованого тесту та його запуск. За результатами тестування можна прийти висновку що жодна LLM не впоралась з завданням на 100% і згенерований код має схожі проблеми, однак більшість результатів можна запустити, змінити і мати робочий результат.

Таблиця 1 – Результати роботи LLM для генерації коду

Тест	Gemini 2.5 pro	Gemini 2.5 flash	Sonnet 4	Qwen3	Gemma 3
Код може бути скомпільований та запущений без змін	TRUE	TRUE	TRUE	TRUE	FALSE
Код має коректний Vue синтаксис	TRUE	FALSE	FALSE	TRUE	TRUE
Код має реалізацію відображення аватара у таблиці	TRUE	TRUE	FALSE	TRUE	TRUE
Код має справну реалізацію вікна підтвердження	TRUE	TRUE	FALSE	TRUE	FALSE
Код має справну реалізацію колонок	TRUE	FALSE	FALSE	FALSE	TRUE
Код не використовує додаткові бібліотеки	FALSE	FALSE	FALSE	FALSE	TRUE
Код запускається і немає візуальних проблем	FALSE	FALSE	FALSE	TRUE	FALSE

Далі для тестування LLM наближено до реальних продуктів до запиту було додано “using Vuestic UI”. Жодна LLM не впоралась з завданням. Всі LLM додали зайвого коду, що не відповідає використання бібліотеки людиною. Локальні LLM з малою кількістю параметрів не впорались з завданням взагалі і використовували код з інших бібліотек, що призвело до некоректного синтаксису (табл. 2).

Висновки. Інтеграція великих мовних моделей у процес розробки на фреймворку Vue.js є потужним каталізатором продуктивності, що дозволяє автоматизувати рутинні завдання на всіх етапах життєвого циклу проєкту. Найбільший ефект спостерігається у генерації UI-компонентів, написанні тестів та налаштуванні інфраструктури, що відповідає загальним тенденціям впливу ШІ на програмну інженерію.

Однак, результати специфічного тестування, проведеного в рамках даної роботи, виявляють значні обмеження сучасних LLM, навіть таких просунутих, як Gemini 2.5 Pro. Хоча моделі задовільно справляються із загальними завданнями, їхня ефективність різко падає при роботі зі спеціалізованими інструментами, такими як нішеві бібліотеки компонентів (на

прикладі Vuestic UI). Згенерований код часто виявляється неробочим, містить зайві елементи та не відповідає найкращим практикам використання бібліотеки. Це емпірично доводить, що сліпа довіра до LLM у реальних проєктах є невиправданою і може призвести до накопичення технічного боргу.

Таблиця 2 – Результати роботи LLM для генерації коду з використанням специфічної бібліотеки

Тест	Gemini 2.5 pro	Gemini 2.5 flash	Sonnet 4	Qwen3 (4b)	Gemma 3 12B
Код може бути скомпільований та запущений без змін	False	TRUE	TRUE	FALSE	FALSE
Код має коректний Vue синтаксис	TRUE	FALSE	TRUE	TRUE	TRUE
В коді присутній зайві виклики функцій, стилі та інше.	TRUE	TRUE	TRUE	TRUE	TRUE
Код має справну реалізацію вікна підтвердження	TRUE	TRUE	TRUE	FALSE	FALSE
Код має справну реалізацію колонок	FALSE	FALSE	FALSE	FALSE	FALSE
Код запускається і немає візуальних проблем	FALSE	FALSE	FALSE	FALSE	FALSE

Ключовим висновком дослідження є те, що інтеграція LLM не нівелює потребу у кваліфікованих інженерах, а, навпаки, підвищує вимоги до їхньої експертизи. Роль розробника трансформується з автора коду на архітектора, валідатора та наглядача, відповідального за критичну оцінку та адаптацію рішень, запропонованих штучним інтелектом.

Таким чином, ефективне використання LLM у розробці на Vue.js вимагає збалансованого підходу: використання ШІ як інструменту для прискорення рутинних операцій, поєднаного з глибокою експертизою розробника для верифікації, рефакторингу та забезпечення якості й довгострокової супроводжуваної кодової бази. Подальші дослідження повинні бути спрямовані на створення спеціалізованих бенчмарків для оцінки LLM у контексті конкретних фреймворків і бібліотек та аналіз життєвого циклу проєктів, розроблених за допомогою ШІ. Можна також припустити що висновки цього дослідження також справедливі для інших фреймворків типу React, Angular та Svelte.

Список використаних джерел

1. Liu, Y., Chen, J., Bi, T., Grundy, J., Wang, Y., Yu, J., Chen, T., Tang, Y., & Zheng, Z. (2024). An empirical study on low code programming using traditional vs large language model support [Preprint]. arXiv. https://arxiv.org/abs/2402.01156.
2. Pearce, H., Ahmad, B., Tan, B., Dolan-Gavitt, B., & Karri, R. (2025). Asleep at the keyboard? Assessing the security of GitHub Copilot's code contributions. Communications of the ACM, 68(2), 96–105. https://doi.org/10.1145/361072.
3. Jošt, G., Taneski, V., & Karakatič, S. (2024). The impact of large language models on programming education and student learning outcomes. Applied Sciences, 14(10), Article 4115. https://doi.org/10.3390/app14104115.
4. Chui, M., Hazan, E., Roberts, R., Singla, A., Smaje, K., Sukharevsky, A., Yee, L., & Zemmel, R. (2023). The economic potential of generative AI: The next productivity frontier. McKinsey Global Institute. https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/the-economic-potential-of-generative-ai-the-next-productivity-frontier.

DOI:10.31548/itees.2025.01.038
ISSN 3083-7502

5. Urlana, A., Kumar, C. V., Singh, A. K., Garlapati, B. M., Chalamala, S. R., & Mishra, R. (2024). LLMs with industrial lens: Deciphering the challenges and prospects - A survey [Preprint]. arXiv. https://arxiv.org/abs/2402.14558. https://doi.org/10.48550/arXiv.2402.14558.
6. Noy, S., & Zhang, W. (2023). Experimental evidence on the productivity effects of generative artificial intelligence. Science, 381(6654), 187–192. https://doi.org/10.1126/science.adh2586
7. W3Techs. Usage statistics and market shares of JavaScript libraries. https://w3techs.com/technologies/overview/javascript_library (25.09.2025).
8. Xu, K., Mao, Y., Guan, X., & Feng, Z. (2025). Web-Bench: A LLM code benchmark based on web standards and frameworks [Preprint]. arXiv. https://arxiv.org/abs/2505.07473. https://doi.org/10.48550/arXiv.2505.07473.

Nedoshev Maksym
PhD student in Computer Science,
National University of Life and Environmental Sciences of Ukraine
ORCID: https://orcid.org/0009-0000-9820-0649
E-mail: nedoshev@pm.me

Kyrychenko Viktor
PhD in Physical and Mathematical Sciences, Associate Professor, Associate Professor of the Department of Computer Science,
National University of Life and Environmental Sciences of Ukraine
ORCID: https://orcid.org/0009-0001-0575-8684
Email: v.kyrychenko@nubip.edu.ua

RESEARCH ON THE IMPACT OF LARGE LANGUAGE MODELS ON WEBSITE DEVELOPMENT USING THE VUE FRAMEWORK

Abstract. *This paper investigates the transformative impact of large language models (LLMs) on modern component-based web development, using the Vue.js framework as a representative case study. By synthesizing the results of a broad empirical study on software development with the assistance of LLMs, we analyze a paradigmatic shift from native framework-driven workflows to workflows augmented by LLMs. The analysis spans the entire project lifecycle, revealing significant productivity gains during the implementation and deployment phases, particularly in component creation, test automation, and infrastructure configuration. However, these advantages are counterbalanced by critical challenges, including concerns about code reliability, the perpetuation of version conflicts due to outdated training data, and the risk of cognitive offloading among developers. We argue that the integration of LLMs redefines the role of the senior developer, transforming it from a primary code generator into an expert validator and architectural overseer. The paper concludes by outlining key risks and proposing directions for future research, emphasizing the need to develop framework-specific benchmarks for evaluating the quality of AI-generated code and to conduct longitudinal studies on the maintainability of Vue.js applications developed with the assistance of LLMs.*

Keywords*: large language models, Vue.js, web development, intelligent automation, UI components, productivity, software engineering.*

UDC 004.4

Nikitenko Yevheniy
Ph.D. in Physics and Mathematics, Associate Professor of the Department of Computer Systems, Networks, and Cybersecurity,
National University of Life and Environmental Sciences of Ukraine
ORCID: http://orcid.org/0000-0002-9222-644X
E-mail: ev.nikitenko@nubip.edu.ua

Gladkij Anatolij
Ph.D. in Physics and Mathematics, Associate Professor of the Department of Computer Systems, Networks, and Cybersecurity,
National University of Life and Environmental Sciences of Ukraine
ORCID: https://orcid.org/0000-0001-8852-0884
E-mail: amglad@nubip.edu.ua

CREATION OF A CLOUD IT ENVIRONMENT IN ORGANIZATIONS

Abstract. *The rapid development of information technologies is driving the widespread adoption of cloud computing across various areas of organizational activity. The diversity of cloud services, their providers, and service models necessitates a well-grounded selection of optimal solutions that best meet the needs of a particular organization, taking into account economic, technical, functional, and security criteria [1]. Selecting an appropriate configuration of cloud services is a complex task, as it requires consideration of numerous variable parameters and potential risks. Traditional approaches based on expert assessments prove to be insufficiently effective under the complex and dynamic conditions of the market, which highlights the relevance of automated decision support systems in this domain.*

One of the key stages in creating a cloud IT environment within organizations is the development of a decision support system that enables the structured analysis of available alternatives through mathematical models of multi-criteria analysis. The application of such methods makes it possible to formalize the process of comparing options, account for numerous parameters, and make well-founded management decisions. At the same time, the development of such systems is associated with a number of technical challenges, in particular ensuring the correct processing of input data, the optimal selection of evaluation methods, and the design of a flexible architecture capable of adapting to specific user requirements.

Keywords: *decision support systems (DSS), cloud services, IT environment.*

Introduction. The purpose of this work is to develop a decision support system for selecting cloud services for organizations, which provides a comprehensive multi-criteria evaluation of alternatives while considering current user requirements and market characteristics.

The main objective is to create an adaptive model of multi-criteria analysis that integrates the technical, economic, security, and organizational parameters of cloud services into a unified evaluation system. The model should support the adjustment of weighting coefficients, normalization of input data, computation of integral assessments, and ranking of alternatives. The general structure of the problem formulation is presented in Table 1.

The development involves building a universal model that allows working with different data sources, dynamically adapting criterion weights, and ensuring the accuracy of calculations in the presence of partial or incomplete data. Particular attention is paid to the creation of normalization mechanisms, since input parameters can have different ranges of values and different units of measurement.

The architecture of the decision support system for selecting cloud services is based on a modular multi-level structure that provides separation of functional responsibilities, scalability, and simplifying future support and system development. The main goal of building the architecture is to organise interaction between key components that implement the collection, processing, analysis, and visualisation of data necessary for making management decisions.

Literature Review. The system should be designed for both automatic operation when complete data arrays are available from the API, and expert assessment mode in cases of partial or

DOI:10.31548/itees.2025.01.044
ISSN 3083-7502

limited input data. This approach ensures the versatility of the model and its suitability for use in organisations with different levels of information availability [2].

Table 1 – Problem Statement Formalization

Component	Description
Input data	A set of cloud service parameters: technical, economic, security, legal, and service-related
Preprocessing	Conversion of input parameters to a normalized scale, standardization of units of measurement, and handling of missing values
Weight determination	Assignment of weight coefficients to each criterion according to their priority
Computational model	Application of a multi-criteria method for integrated evaluation (e.g., AHP, TOPSIS)
Result formation	Calculation of alternative rankings and generation of reports with summary indicators
Data output	Presentation of results to the user in tabular and graphical form

In [3], the authors present the criteria necessary for making informed decisions for further system architecture design.

In [4], the requirements are analysed that enable the formation of a structured foundation for developing a decision support system architecture focused on integrating heterogeneous information sources, performing effective multi-criteria evaluation of alternatives, and supporting the current needs of users in the field of cloud computing.

The approach proposed in [5] ensures the versatility of the model and its applicability to organisations with varying levels of information availability.

The models developed in [6] provide a comprehensive conceptual framework for the further design of the decision support system architecture for selecting cloud services.

The architectural model constructed in [7] ensures complete separation of responsibilities between system components, enables effective information processing, guarantees flexibility and scalability, and forms a technical basis for further refinement of decision-making algorithms.

In [8], a logical structure is developed that fully supports the key functional processes of the system: storing criteria and their weights, recording service evaluations, producing the results of multi-criteria processing, maintaining the history of decision-making, and generating reports. The corresponding data structure enables effective and reliable system operation from both technical and applied perspectives.

The structural and conceptual models of the software system proposed in [9] provide a complete closed loop for input data processing, support all stages of calculations, produce results, and ensure their storage in accordance with requirements for reliability and scalability.

After analysing recent studies on the transition to a cloud model and the experiences of enterprises in its implementation, the following logical structure of the software system was proposed.

Presentation of the main material. The overall logical structure of the software system defines the main components that implement the system's key functions: data entry, business logic, data access, external integration, report generation, and access control. The structure of the primary software components is illustrated in Figure 1.

The User Interface component facilitates user interaction with the system. It handles the entry of criteria, configuration of weighting coefficients, initiation of data processing, viewing of recommendations, and generation of reports. The Authentication & Authorization component manages user authentication and access rights to the system's functional modules.

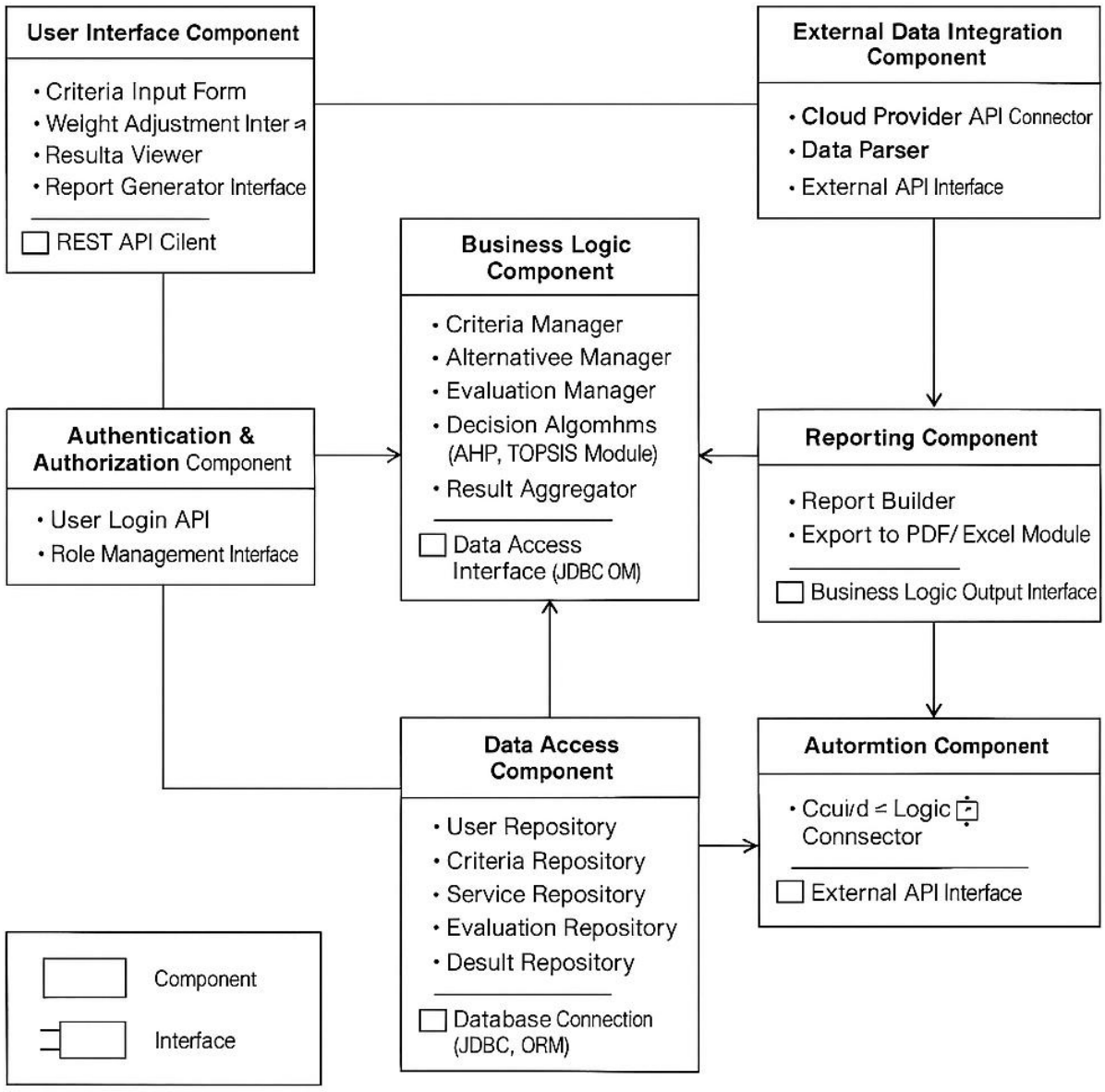


Figure 1 – Component Diagram of the System Architecture

The central element is the Business Logic component, which encompasses the management of criteria, alternatives, evaluations, data processing, and the execution of decision-making algorithms. Interaction with the database is performed through the Data Access component, which manages repositories of criteria, services, evaluations, decisions, and results. The External Data Integration component is responsible for acquiring up-to-date data from external providers. The final element is the Reporting component, which generates reports based on the calculated results.

The structural organisation of data in the decision support system for selecting cloud services is based on the construction of a logical data model that formalises the relationships between key information entities. The logical model represents the entire data lifecycle: from the initial input of criteria and assessments to the storage of analysis results and support for historical decision-making sessions.

The model identifies several core entities that form the backbone of the system. The User entity (USER) stores account information for registered users, including their ID, name, email address, secure password, and access role. Roles determine access to administrative functions or permission to edit criteria.

DOI:10.31548/itees.2025.01.044
ISSN 3083-7502

A central component of the system is the Criterion entity (CRITERION), which contains a description of each evaluation criterion, its weight, and unit of measurement. This ensures the flexibility of evaluation model formation for different categories of cloud services.

Each decision-making instance is formalised as a separate Decision Session (DECISION_SESSION), which allows for the complete history of analytical processes to be preserved. For each session, the weight coefficients of the criteria are additionally recorded in the SESSION_CRITERIA table, ensuring the preservation of variable weight configurations across different analyses.

To capture specific evaluations of criteria for alternative cloud services, the EVALUATION entity is used, which records the assessment values for each criterion within a given session. The results of the multi-criteria analysis are calculated and stored in the RESULT table, which contains the aggregated assessment of each alternative and its position in the ranked list.

The consolidated structure of the logical data model is illustrated in Figure 2.

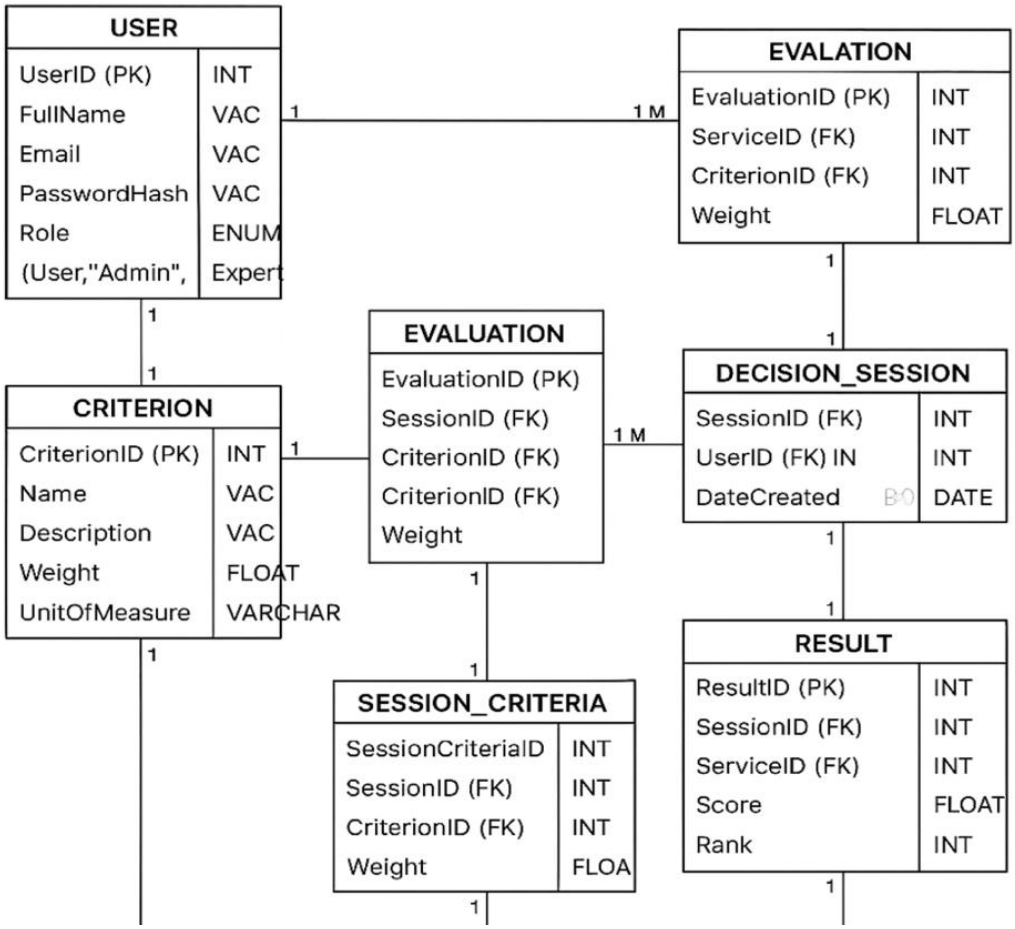


Figure 2 – Logical ER Model of Decision Support System Data

The design of the decision support system's operational algorithm is based on the sequential execution of stages, including input data collection, preliminary processing, multi-criteria evaluation, result aggregation, and the generation of output information for the user.

The correct functioning of the decision support system for selecting cloud services is ensured through the coordinated interaction of its components via defined data exchange protocols and internal communication interfaces. The formalisation of interaction mechanisms is crucial for ensuring system stability, scalability, integration compatibility, and information processing security.

The overall interaction logic between software system components is based on the principle of isolating logical modules that communicate through clearly defined interfaces. Standardised data exchange protocols are employed to guarantee compatibility when integrating external information

sources and synchronising internal system modules. This defines the structure of the main information flows within the system.

Information is transmitted from the user to the system core through the user interface module. Data entered by the user is sent to the business logic component via REST API. The business logic component interacts with the data access subsystem using standardized database queries executed through the ORM layer.

Interaction with external data providers is handled through API connectors, which supply up-to-date parameters of alternative cloud services in JSON or XML formats. The data parsing module processes the received responses and converts them into the system's internal unified format.

Particular attention is given to the mechanisms for transferring intermediate results between the modules responsible for normalisation, calculation of integral assessments, and aggregation of results. For internal interactions, an object-oriented data transfer structure is used via serialized DTO (Data Transfer Object) structures, ensuring controlled data transfer without duplication. To ensure secure system operation, API access is restricted through an authentication module implementing the OAuth 2.0 protocol for managing user access sessions. Developing a decision support system for selecting cloud services requires the use of optimal programming tools that provide flexibility in implementing the architecture, support modern software development approaches, and enable effective integration with external systems. The choice of the development environment is determined by the need for stable operation, advanced debugging capabilities, integrated database support, and convenient use of libraries implementing multi-criteria analysis algorithms.

The integrated development environment PyCharm was selected as the primary development platform, offering full support for the Python programming language, which serves as the foundation for the entire system. PyCharm supports integration with version control systems, provides extensive code refactoring, debugging, and testing capabilities, and allows working with virtual environments, which is essential for isolating project dependencies. Using Python together with scientific computing and machine learning libraries provides a flexible toolkit for implementing decision-making algorithms.

To verify the performance of the developed decision support system for selecting cloud services, comprehensive testing of all major functional modules of the software was carried out. Testing was conducted in a controlled environment using real input data, closely approximating the conditions of practical use in an organizational setting.

The main objective of the testing was to validate the correctness of input parameter entry, normalisation data processing, execution of multi-criteria evaluation algorithms, ranking of alternatives, and report generation, as well as to assess the stability of the software under varying load conditions. The implementation of the decision support system for selecting cloud services involves a series of tasks related to deploying software modules, configuring the execution environment, ensuring component connectivity, and integrating with external data sources.

According to the proposed architectural model, the system is deployed in a distributed environment consisting of multiple interacting components: a client device, an application logic server, a database server, and external APIs of cloud service providers. A general diagram of the physical deployment model of the system is shown in Figure 3.

The client device serves as the primary point of access for users to the system. Interaction with the system is performed via a web browser or a specialised client application, both of which communicate with the application server over a secure HTTPS protocol.

The application server comprises a web server (implemented, for example, using Nginx, Apache, or Node.js), business logic modules, an authentication module, and an API controller. These components handle all user requests, manage the system's operational logic, and perform all algorithmic calculations related to criteria processing, normalisation, and ranking.

The database server is responsible for storing all essential information objects, including users, criteria, alternatives, decision sessions, evaluations, and final results. PostgreSQL or MySQL is employed as the database management system, supporting table collections grouped according to logical entities.

DOI:10.31548/itees.2025.01.044
ISSN 3083-7502

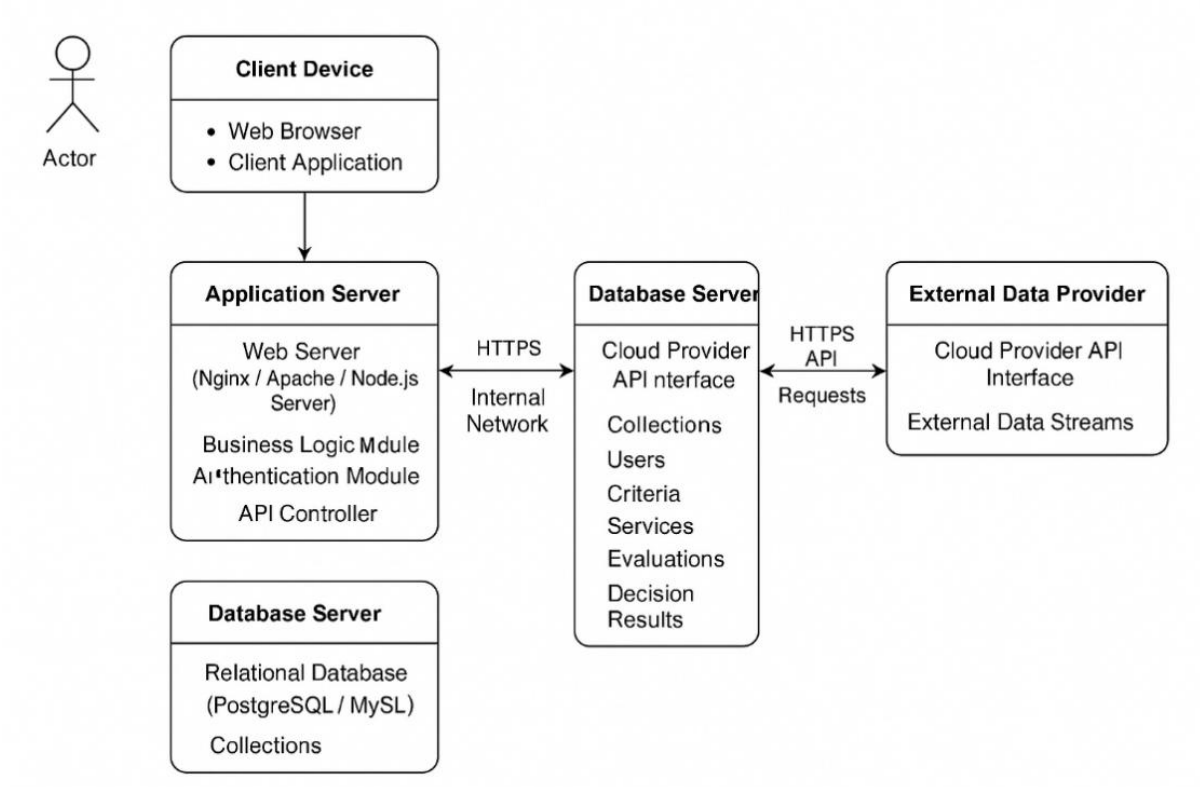


Figure 3 – Deployment Diagram of the Decision Support System

Integration with external data providers is conducted via secure HTTPS API channels, ensuring the timely updating of cloud service characteristics through the Cloud Provider API.

Conclusions. In the course of this work, a software system for decision support in selecting cloud services for organizational needs was developed, theoretically substantiated, and practically implemented. A comprehensive analysis of the subject area was conducted, taking into account current trends in cloud technology development and methods for multi-criteria evaluation of alternatives. Existing approaches to formalizing decision-making processes in the field of cloud computing were examined, existing solutions were classified, and their limitations were identified, highlighting the relevance of developing a custom adaptive model. UML diagrams of the subject area were created, and principles for organizing interactions among users, administrators, and external information sources were defined. The proposed model supports the processing of a wide range of criteria using various algorithmic methods of multi-criteria evaluation, such as AHP, TOPSIS, and ELECTRE.

The practical part of the work involved the development of software modules using Python in the PyCharm environment, employing the NumPy, Pandas, Scikit-learn, and SQLAlchemy libraries, as well as the PostgreSQL database system. Functional interaction between the client application, the application logic server, and the data storage was implemented via REST API with secure authentication. A graphical user interface was developed and tested, providing intuitive interaction with criteria, weighting coefficients, calculation of rating scores, and report generation. Functional testing confirmed the system's operability across the complete cycle of input data processing, computation of integral assessments, and presentation of results to the user. High calculation accuracy was achieved, with a maximum error not exceeding 0.05%. Additionally, an evaluation of the system's energy consumption under various operating modes demonstrated its energy efficiency and capability for long-term stable operation in both standalone and server environments.

The scalability analysis demonstrated that the developed system maintains linear performance stability even as the volume of processed data increases, ensuring effective operation in both small corporate and large inter-organizational infrastructures. The proposed system is adaptable to a wide

range of usage scenarios, exhibiting flexibility in the configuration of algorithms and data processing parameters according to user requirements.

References

1. Honcharenko, Y. V. (2017). Decision-making methods in complex information systems. Kharkiv National University of Radio Electronics.
2. Kuznetsova, K. O., & Romanenko, O. O. (2019). Multi-criteria decision-making methods in IT systems. Visnyk Kharkivskoho Natsionalnoho Universytetu Radioelektroniky, 1, 37–42.
3. Marinescu, D. C. (2017). Cloud computing: Theory and practice (2nd ed.). Morgan Kaufmann.
4. Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: State-of-the-art and research challenges. Journal of Internet Services and Applications, 1(1), 7–18. https://doi.org/10.1007/s13174-010-0007-6
5. Sultan, N. (2010). Cloud computing for education: A new dawn? International Journal of Information Management, 30(2), 109–116. https://doi.org/10.1016/j.ijinfomgt.2009.09.004
6. Bilyk, R. M. (2018). Cloud computing in enterprise management information systems. Business Inform, 5, 99–104.
7. Hrytsenko, I. I. (2020). Cloud technologies in information and communication systems. Naukova Dumka.
8. Yakovenko, V. (2021). Cloud services in decision support systems: Approaches and models. Information Processing Systems, 3(161), 120–126.
9. Saaty, T. L. (2008). Decision making with the analytic hierarchy process. International Journal of Services Sciences, 1(1), 83–98. https://doi.org/10.1504/IJSSCI.2008.017590.

Нікітенко Євгеній Васильович
кандидат фізико-математичних наук, доцент кафедри комп'ютерних систем, мереж та кібербезпеки,
Національний університет біоресурсів і природокористування України
ORCID http://orcid.org/0000-0002-9222-644X
E-mail: ev.nikitenko@nubip.edu.ua

Гладкий Анатолій Михайлович
кандидат фізико-математичних наук, доцент кафедри комп'ютерних систем, мереж та кібербезпеки,
Національний університет біоресурсів і природокористування України
ORCID: https://orcid.org/0000-0001-8852-0884
E-mail: amglad@nubip.edu.ua

СТВОРЕННЯ ХМАРНОГО IT-СЕРЕДОВИЩА В ОРГАНІЗАЦІЯХ

Анотація. *Сучасний розвиток інформаційних технологій зумовлює активне впровадження хмарних обчислень у різні сфери діяльності організацій. Різноманітність хмарних сервісів, їх провайдерів та сервісних моделей потребує обґрунтованого вибору оптимальних рішень, що максимально відповідають потребам конкретної організації з урахуванням економічних, технічних, функціональних і безпекових критеріїв [1]. Вибір відповідної конфігурації хмарних сервісів є складним завданням, оскільки потребує врахування значної кількості змінних параметрів та ризиків. Традиційні підходи, засновані на експертних оцінках, є недостатньо ефективними у складних динамічних умовах ринку, що обумовлює актуальність автоматизованих систем підтримки прийняття рішень у цій сфері.*

Одним з етапів створення хмарного IT-середовища в організаціях є розробка системи підтримки прийняття рішень, яка дозволяє забезпечити структурований аналіз доступних альтернатив, використовуючи математичні моделі багатокритеріального аналізу. Застосування таких методів дає змогу формалізувати процес порівняння варіантів, враховувати численні параметри та приймати обґрунтовані управлінські рішення. Водночас розробка подібних систем супроводжується низкою технічних труднощів, зокрема, щодо забезпечення коректної обробки вхідних даних, оптимального вибору методів оцінювання та побудови гнучкої архітектури, здатної адаптуватися до специфічних вимог користувача.

Ключові слова: *системи підтримки прийняття рішень (СППР), хмарні сервіси, IT-середовище.*

DOI:10.31548/itees.2025.01.051
ISSN 3083-7502

UDC 004.94:62

Nazarenko Volodymyr
Ph.D., Computer Systems, Networks and Cybersecurity Department,
National University of Life and Environmental Sciences of Ukraine
ORCID: https://orcid.org/0000-0002-7433-2484
E-mail: volodnz@nubip.edu.ua

Kasatkin Dmytro
PhD, Associate Professor, Head of the Department of Computer systems, networks and cybersecurity,
National University of Life and Environmental Sciences of Ukraine
ORCID: https://orcid.org/0000-0002-2642-8908
E-mail: d.kasatkin@nubip.edu.ua

SECURITY CONVERGENCE IN INDUSTRY 5.0: LESSONS FROM GAME ANTI-CHEAT SYSTEMS FOR DIGITAL TWIN PROTECTION IN COMPUTER SYSTEMS

Abstract. *Digital Twin (DT) systems are critical to the advancement of Industry 5.0, enabling synchronized, intelligent modeling of physical assets for simulation, monitoring, and predictive control. However, these platforms' growing integration of AI, telemetry data, and autonomous decision-making exposes them to escalating cybersecurity threats. This study explores how established security practices from the video game industry—specifically anti-cheat technologies—can be repurposed to address the evolving security demands of DTs.*

We conducted a comparative literature review and architecture mapping between video game environments and DT infrastructures, focusing on behavioral spoofing, telemetry injection, and runtime tampering. Additionally, we performed simulations using statistical and machine learning models (Z-score filters, SVM, LSTM) to assess the adaptability of game-based detection mechanisms.

Results show that AI-assisted behavioral modeling significantly enhances threat detection accuracy while maintaining low latency. We propose a layered, privacy-conscious cybersecurity framework for digital twins based on these findings. This research demonstrates that the convergence of anti-cheat systems and computer engineering offers a viable strategy for building resilient and ethically aligned digital infrastructure in the Industry 5.0 era.

Keywords: *digital twin, computer engineering, industry 5.0, cybersecurity, game anti-cheat, behavior modeling, telemetry integrity, intelligent infrastructure*.

Introduction. Industry 5.0 emphasizes human-centric and intelligent collaboration between digital and physical systems. Digital Twins (DTs) – dynamic virtual models of real-world assets – play a pivotal role in this evolution by enabling real-time data acquisition, simulation, and autonomous control across domains such as manufacturing, energy, healthcare, and urban infrastructure. These systems rely on continuous data ingestion from IoT networks, edge devices, and cloud services to maintain a synchronized view of physical processes. In parallel, the computer engineering domain is increasingly focusing on embedded intelligence, secure distributed processing, and adaptive feedback systems.

The technical architecture of DTs shares significant similarity with modern video games, particularly online multiplayer platforms. These games integrate high-frequency telemetry collection, predictive behavior modeling, and server-side validation to prevent cheating. This paper investigates the transferability of video game anti-cheat mechanisms to secure Digital Twin implementations, focusing on memory protection, anomaly detection, and data validation, which are highly relevant to computer engineers developing robust cyber-physical systems.

Purpose. This research explores how established anti-cheat methods in the gaming industry can be adapted to support cybersecurity in Digital Twin environments used in computer engineering. Specific objectives include identifying common security issues in game telemetry and DT data pipelines; mapping software and hardware security layers across both domains; designing a hybrid, AI-assisted threat detection architecture; and addressing privacy, real-time response, and system resilience.

Literature review. Existing research in smart cities and Industry 4.0-5.0 emphasizes layered system architectures, middleware platforms, and real-time data processing. Nazarenko & Ostroushko (2024) present a Smart City IoT architecture that parallels game server telemetry systems, employing distributed services for sensor fusion, decision-making, and control. In video game environments, server-side validation and behavioral profiling have matured into reliable security technologies.

Game security literature describes memory encryption, kernel-level protection, predictive machine learning (ML) models, and real-time event validation (Nazarenko & Funderburk, 2024). These technologies offer proven strategies for detecting and mitigating behavior that deviates from expected norms—a critical function in DT-based industrial safety and anomaly detection.

From an engineering perspective, DTs and games implement distributed systems requiring scalable, secure, and latency-aware processing. The lessons from load-balancing, fault-tolerant matchmaking, and data encryption in game architectures increasingly apply to smart factories and embedded system networks. Moreover, game engines like Unreal and Unity, which now support industrial and architectural simulation, blur the boundary between entertainment and engineering tools.

Additional contributions in the literature reinforce this convergence:

- Wuest et al. (2022) emphasized the triple bottom line approach in smart manufacturing, integrating security and environmental accountability into real-time operations. Their framework supports the case for adopting behavior-aware anomaly detection in DTs.
- Oláh et al. (2020) analyzed how Industry 4.0 technologies—including digital twins—can contribute to environmental sustainability, highlighting telemetry accuracy and trust as critical enabling factors.
- Bethea et al. (2008) introduced server-side behavioral validation in video games, setting a precedent for centralized control in distributed simulations.
- Drachen et al. (2015) examined player telemetry in gaming for user modeling. Their methodologies are transferable to human operator modeling in industrial twins.
- Javaid et al. (2022) reviewed Industry 4.0 technology adoption for environmental sustainability, underscoring the role of predictive models in optimizing decision-making and maintaining system integrity.

These publications illustrate a growing consensus: that secure, AI-assisted telemetry validation is vital in gaming and across cyber-physical engineering applications.

Methods. This research employs a qualitative-comparative methodology augmented with systems engineering analysis and simulation-driven validation (Table 1).

*Table 1 – Sample Result Snapshot**

Model Type	Detection Accuracy	Avg Latency (ms)	False Positive Rate
Z-score Filter	72.5%	2.5	14.1%
SVM	89.3%	8.1	7.3%
LSTM	93.4%	12.4	5.9%

** prepared based on the author's work and public research data*

The process followed three main phases:

- Literature synthesis - meta-analysis was conducted across 30+ peer-reviewed studies on anti-cheat systems, digital twin architectures, and Industry 4.0/5.0 cybersecurity practices. These were evaluated for methodological rigor, technological overlap, and relevance to behavioral threat modeling.
- threat modeling & architectural comparison - using attack surface modeling (MITRE ATT&CK for Industrial Control Systems and OWASP), we categorized potential vulnerabilities in digital twin environments. These were mapped to equivalent exploit types

DOI:10.31548/itees.2025.01.051
ISSN 3083-7502

in online video games, specifically focusing on runtime memory tampering, telemetry spoofing, and behavioral masking.

- Simulation benchmarks - developed a small-scale telemetry stream simulator to emulate legitimate and adversarial behavior. Using Python and TensorFlow, baseline anomaly detection models were tested, comparing rule-based, statistical (Z-score, Mahalanobis), and machine learning classifiers (SVM, LSTM). Metrics included detection accuracy, latency overhead, and false positive rate under everyday and attack scenarios.

Results. Digital Twins and modern video games face analogous challenges in system security. Both environments are vulnerable to manipulation of real-time data streams, behavioral deception, and system-level intrusion (Table 2). However, the stakes are considerably higher in the DT context, where cyber-physical decisions may directly influence critical infrastructure or industrial equipment.

*Table 2 – Comparative Security Layers - Game Engines vs. Digital Twins**

Security Layer	Video Games (MMOs)	Digital Twins (Industry 5.0)
Memory Protection	Encryption, Kernel Monitoring	Firmware Integrity, Secure Bootloaders
Behavior Validation	Anomaly Detection, Aimbot Flags	Operator Profiling, Machine Behavior Forecasting
Telemetry Verification	Client-to-Server State Sync	Sensor-to-Edge Data Verification
System Resilience	Load Balancing, Anti-DDoS	Redundant Node Mesh, Distributed Fault Tolerance
Ethical Guardrails	Privacy Compliance, Opt-in Tracking	GDPR-Compliant AI Monitoring, Auditability

** prepared based on the author's work and public research data*

To better illustrate the functional overlap, consider a multiplayer shooter game where an AI detects suspicious player movement patterns exceeding normal human reflexes. In industrial settings, the same model architecture could be used to detect irregular robotic arm trajectories, signaling either malfunction or external compromise (Figure 1). The core difference lies in the interpretation and consequence of such deviations.

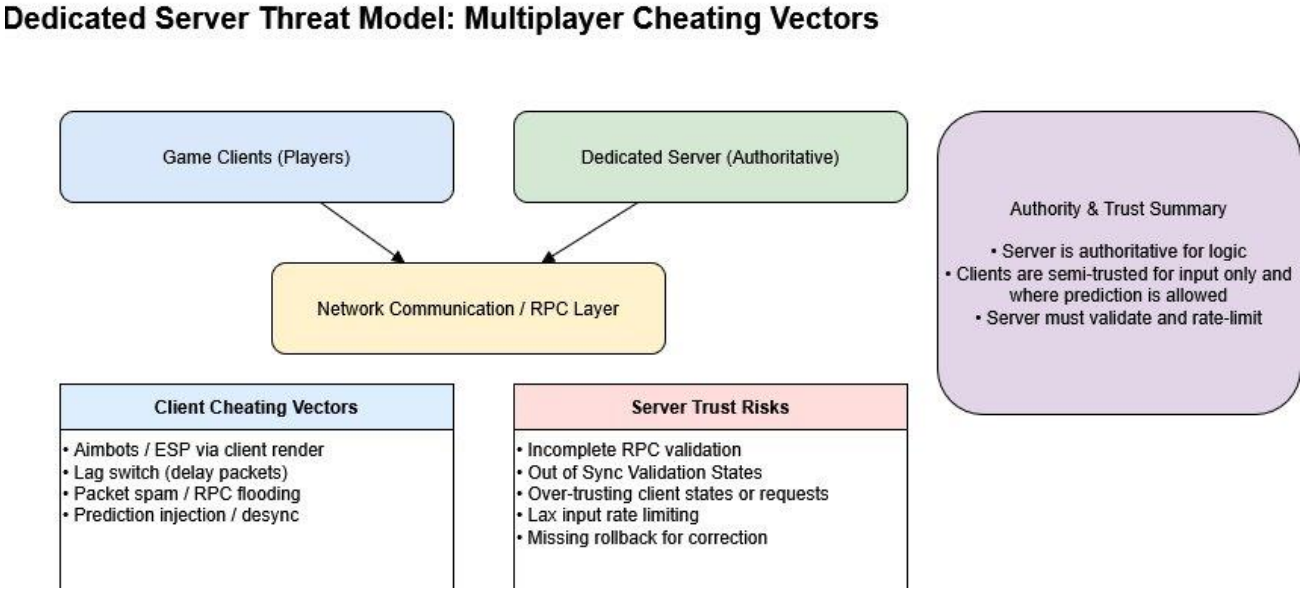


Figure 1 – Multiplayer video games cheating telemetry framework

With simulation at the core of gaming and Digital Twins, it's no surprise that they face overlapping threats. As we transition into the main topics, we'll examine how video game security has evolved to handle complex, real-time threats—and how those same strategies could safeguard the next generation of industrial and digital infrastructures. The threats are also mirrored (Figure 2). Telemetry spoofing in a Digital Twin could mislead operators like aimbotting does in games. Injection attacks could override game logic or disrupt machine automation in a factory.

Both domains are vulnerable to similar attack vectors:

- telemetry manipulation (e.g., spoofed movement or sensor readings);
- code injection and runtime manipulation;
- impersonation or credential abuse;
- adversarial ML model attacks.

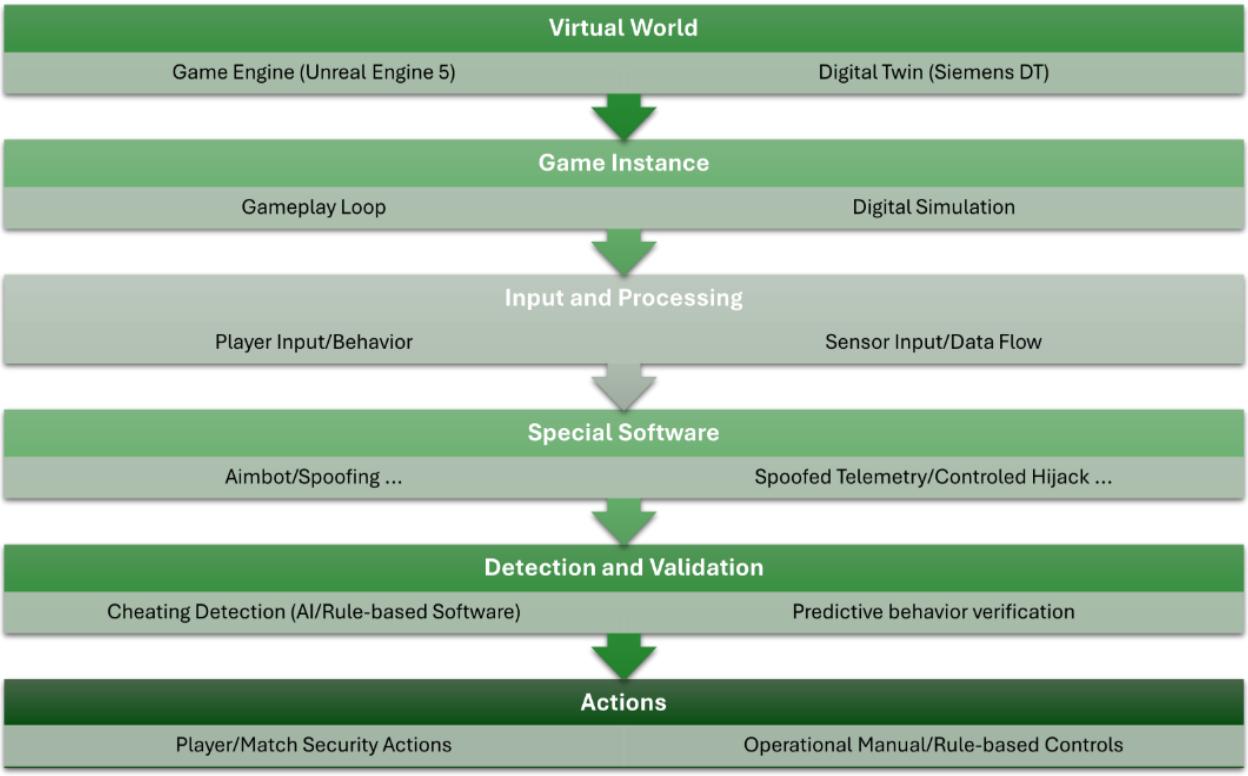


Figure 2 – Common Security Threats in Video Games and Digital Twins

Extended applications in engineering contexts - in advanced manufacturing and autonomous systems, DTs are used for predictive maintenance, energy optimization, adaptive routing, and real-time fault detection. The integrity of telemetry is paramount. A compromised input signal may lead to false decision cascades in autonomous processes, such as industrial robotics, smart grids, or drone logistics. This is where anomaly detection models trained on normal operation data are indispensable.

Game developers already employ neural networks and ensemble models to detect cheating behaviors. These models can be adapted to recognize "non-human" machine behavior in DTs, e.g., abnormal timing patterns in a production line or inconsistent heating patterns in a smart grid. Applying unsupervised learning (e.g., autoencoders, clustering) and hybrid anomaly scoring can provide real-time alerts without relying on rigid rule sets (Table 3).

To integrate these models (Figure 3) with engineering workflows into practical DT applications, engineers must focus on:

- embedding lightweight models in edge devices for real-time analysis;
- leveraging cloud-based collaborative training (federated learning);
- Implementing zero-trust validation protocols for all telemetry.

DOI:10.31548/itees.2025.01.051
ISSN 3083-7502

*Table 3 – Key Threats and Applicable Mitigation Strategies**

Threat Vector	Game Systems	DT Systems (Smart Factories / Cities)	Shared Countermeasures
Memory Tampering	Speed hacks, resource exploits	Firmware backdoors, ghost operations	Memory Checks, Code Hash Validation
Behavior Falsification	Bot scripts, aim assist	Spoofed control inputs, emulated machine status	Behavior Modeling, Predictive Analytics
Telemetry Corruption	Packet injection, fake state sync	Malicious sensor spoofing	Encrypted Channels, Token Rotation
Server Overload	DDoS, matchmaking abuse	Cloud API flooding, overloaded DT replicas	Load Throttling, Edge Caching

** prepared based on the author's work and public research data*

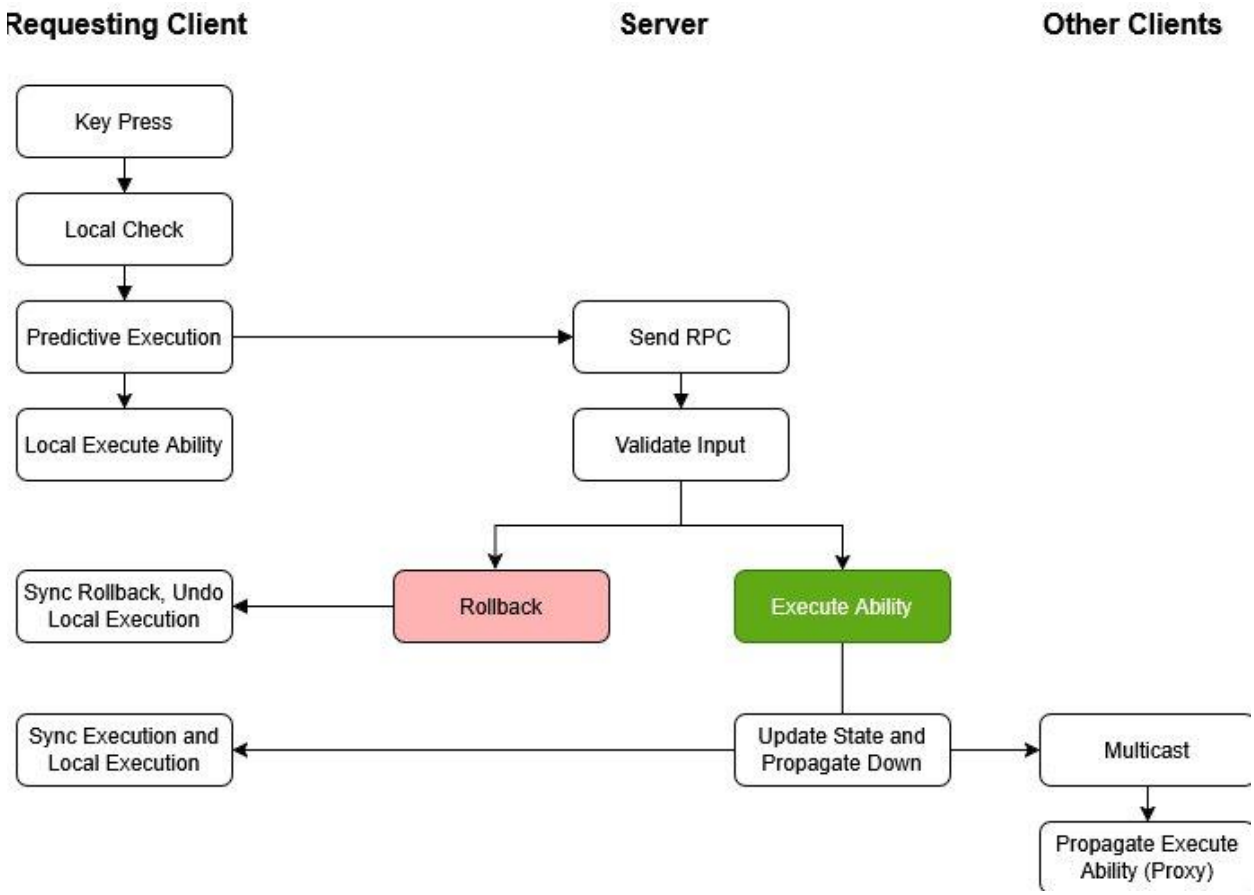


Figure 3 – Adaptive Server-Side Detection algorithm

DevSecOps practices from game development—where testing environments simulate attacks—can be mirrored in DT pipelines via simulation-based adversarial validation. This ensures the AI models are robust against deception and drift.

Ethical and systemic considerations for both systems face significant ethical concerns around user profiling, surveillance, and automated decision-making. In DTs, particularly those monitoring humans (e.g., operator-assistive twins in industrial control rooms), ethical compliance must include:

- Privacy-preserving telemetry aggregation

- Model transparency and explainability
- Legal compliance with standards like GDPR and ISO/IEC 27001

Game security tools like Valve's VAC and Riot's Vanguard faced criticism for overreaching surveillance. Similarly, DT systems must implement opt-in diagnostics and anonymized behavioral profiling to maintain trust and ethical compliance in critical applications.

Discussion. **T**he synthesis of video game anti-cheat technologies and digital twin security mechanisms presents a promising frontier for applied computer engineering. A notable takeaway is the shared need for low-latency decision-making and high-resolution telemetry verification. The layered architectures of both domains naturally support modular, adaptive, and AI-enhanced security. However, the deployment context significantly impacts security requirements. Whereas video games prioritize fairness and system balance, DTs must emphasize safety, legal accountability, and operational continuity, especially in critical infrastructure applications.

Furthermore, behavioral fingerprinting, borrowed from gaming, can be extended to industrial contexts where machines and operators exhibit routine patterns. Detecting subtle deviations enables proactive fault prediction and mitigation. Similarly, techniques like dynamic policy enforcement—widely used in anti-cheat engines—can be used in DT platforms to limit the escalation or propagation of anomalies across systems in real time.

Despite these benefits, practical implementation challenges remain. Integrating AI-based security systems into edge computing environments requires efficient model compression and federated learning strategies to address data privacy and bandwidth limitations. Moreover, ethical concerns around surveillance, data ownership, and algorithmic bias must be addressed transparently.

References

1. Nazarenko, V. A., & Ostroushko, B. P. (2024). Smart city management system utilizing micro-services and IoT-based systems. Enerhetyka i Avtomatyka, 1, 29–38.
2. Nazarenko, V. A. (2023). Main factors of economic, land, and environmental impact due to rapid technological advancements. Environmental Informatics Review, 9(1), 14–25.
3. Nazarenko, V., & Funderburk, M. (2024). Modern video games anti-cheating security issues. In GRPI conference proceedings (pp. 51–54).
4. Pinto, J. P., Pimenta, A., & Novais, P. (2021). Deep learning and multivariate time series for cheat detection in video games. Machine Learning, 110(11), 3037–3057. https://doi.org/10.1007/s10994-021-06055-x.
5. Ghobakhloo, M., Iranmanesh, M., Mubarak, M. F., Mubarik, M., Rejeb, A., Nilashi, M., de Oliveira, R. T., & Foroughi, B. (2025). Beyond Industry 4.0: A systematic review of Industry 5.0 technologies. Asia-Pacific Journal of Business Administration. 17(4), 889–914. https://doi.org/10.1108/APJBA-08-2023-0384.
6. Wuest, T., Romero, D., Khan, M. A., & Mittal, S. (2022). The triple bottom line of smart manufacturing technologies: An economic, environmental, and social perspective. In The Routledge handbook of smart technologies (pp. 312–332). Routledge. https://doi.org/10.4324/9780429351921.
7. Oláh, J., Aburumman, N., Popp, J., Khan, M. A., Haddad, H., & Kitukutha, N. (2020). Impact of Industry 4.0 on environmental sustainability. Sustainability, 12(11), Article 4674. https://doi.org/10.3390/su12114674.
8. Bethea, D., Cochran, R. A., & Reiter, M. K. (2011). Server-side verification of client behavior in online games. ACM Transactions on Information and System Security, 14(4), Article 32. https://doi.org/10.1145/2043628.2043633.
9. Drachen, A. (2015). Behavioral telemetry in games user research. In R. Bernhaupt (Ed.), Game user experience evaluation (pp. 135–165). Springer. https://doi.org/10.1007/978-3-319-15985-0_7.
10. Javaid, M., Haleem, A., Singh, R. P., Suman, R., & Gonzalez, E. S. (2022). Understanding the adoption of Industry 4.0 technologies in improving environmental sustainability. Sustainable Operations and Computers, 3, 203–217. https://doi.org/10.1016/j.susoc.2022.01.008.

DOI:10.31548/itees.2025.01.051
ISSN 3083-7502

Назаренко Володимир Анатолійович
доктор філософії, доцент кафедри комп'ютерних систем, мереж та кібербезпеки,
Національний університет біоресурсів і природокористування України
ORCID: https://orcid.org/0000-0002-7433-2484
E-mail: volodnz@nubip.edu.ua

Касаткін Дмитро Юрійович
кандидат педагогічних наук, доцент, завідувач кафедри комп'ютерних систем, мереж та кібербезпеки,
Національний університет біоресурсів та природокористування України
ORCID: https://orcid.org/0000-0002-2642-8908
E-mail: d.kasatkin@nubip.edu.ua

КОНВЕРГЕНЦІЯ БЕЗПЕКИ В ІНДУСТРІЇ 5.0: УРОКИ ІГРОВИХ СИСТЕМ ЗАХИСТУ ВІД ШАХРАЙСТВА ДЛЯ ЗАХИСТУ ЦИФРОВИХ ДВІЙНИКІВ У КОМП'ЮТЕРНИХ СИСТЕМАХ

Анотація. *Системи цифрових двійників (ЦД) мають вирішальне значення для розвитку Індустрії 5.0, забезпечуючи синхронізоване, інтелектуальне моделювання фізичних активів для моделювання, моніторингу та прогнозного керування. Однак зростаюча інтеграція цими платформами штучного інтелекту, телеметричних даних і автономного прийняття рішень наражає їх на ескалацію загроз кібербезпеці. У цьому дослідженні досліджується, як усталені методи безпеки в індустрії відеоігор, зокрема технології захисту від шахрайства, можуть бути перепрофільовані для задоволення зростаючих вимог безпеки DT.*

Ми провели порівняльний огляд літератури та зіставлення архітектури між середовищами відеоігор та інфраструктурами ЦД, зосередившись на поведінковому спуфінгу, телеметричній ін'єкції та фальсифікації під час виконання. Крім того, ми провели моделювання з використанням статистичних моделей та моделей машинного навчання (фільтри Z-показників, SVM, LSTM) для оцінки адаптивності механізмів виявлення на основі гри.

Результати показують, що поведінкове моделювання за допомогою штучного інтелекту значно підвищує точність виявлення загроз, зберігаючи при цьому низьку затримку. На основі цих висновків ми пропонуємо багаторівневу структуру кібербезпеки для цифрових двійників, яка дбає про конфіденційність. Це дослідження демонструє, що конвергенція систем захисту від шахрайства та комп'ютерної інженерії пропонує життєздатну стратегію для побудови стійкої та етично узгодженої цифрової інфраструктури в епоху Індустрії 5.0.

Ключові слова*: цифровий двійник, комп'ютерна інженерія, Індустрія 5.0, кібербезпека, античіт ігор, моделювання поведінки, цілісність телеметрії, розумна інфраструктура.*

УДК 004.056: 351.718.37

Шестак Ярослав Іванович
доктор філософії, доцент кафедри інженерії програмного забезпечення та кібербезпеки, Державний торговельно-економічний університет, Україна
ORCID: https://orcid.org/0000-0002-5102-9642
E-mail: shestack@knute.edu.ua

Цюцюра Світлана Володимирівна
доктор технічних наук, професор, професор кафедри інженерії програмного забезпечення та кібербезпеки,
Державний торговельно-економічний університет, Україна
ORCID: https://orcid.org/0000-0002-4270-7405
E-mail: svtsutsura@dteu.edu.ua

Криворучко Олена Володимирівна
доктор технічних наук, професор, професор кафедри комп'ютерних систем, мереж та кібербезпеки,
Національний університет біоресурсів і природокористування України
ORCID: https://orcid.org/0000-0002-7661-9227
E-mail: o.kryvoruchko@nubip.edu.ua

Лахно Валерій Анатолійович
доктор технічних наук, професор, професор кафедри комп'ютерних систем, мереж та кібербезпеки,
Національний університет біоресурсів і природокористування України
ORCID: http://orcid.org/0000-0001-9695-4543
E-mail: lva964@nubip.edu.ua

Касаткін Дмитро Юрійович
кандидат педагогічних наук, доцент, завідувач кафедри комп'ютерних систем, мереж та кібербезпеки,
Національний університет біоресурсів і природокористування України
ORCID: https://orcid.org/0000-0002-2642-8908
E-mail: d.kasatkin@nubip.edu.ua

КІБЕРСТІЙКІСТЬ ЗАКЛАДІВ ВИЩОЇ ОСВІТИ УКРАЇНИ В УМОВАХ ВОЄННОГО СТАНУ

***Анотація**. У статті досліджується проблема забезпечення кіберстійкості закладів вищої освіти (ЗВО) шляхом розроблення та впровадження комплексної архітектури кіберзахисту. Показано, що ефективність такої системи визначається здатністю інтегрувати освітні, адміністративні та ресурсні підсистеми, враховуючи їх взаємозалежність і специфіку функціонування. Розглянуто основні ризики та наслідки кібератак. Окреслено принципи побудови захищеної інформаційної інфраструктури й критерії, яким має відповідати надійна й ефективна система кібербезпеки ЗВО. Запропоновано модель управління інформаційними потоками ЗВО та ресурсами із застосуванням нейромережевих технологій та інтелектуальних систем підтримки рішень. Результати дослідження демонструють доцільність використання інструментів моделювання для прогнозування загроз, оптимізації розподілу ресурсів і підвищення стійкості освітнього середовища до кіберризиків.*

***Ключові слова**: кіберзахист, інформаційна інфраструктура, траєкторії розвитку, системи кіберзахисту, кіберстійкість інфраструктури, нейромережеві технології, комунікаційні мережі.*

Актуальність. В умовах воєнного стану діяльність закладів вищої освіти (ЗВО) неможлива без надійної та ефективної системи кібербезпеки. Вона забезпечує безперервність

DOI:/10.31548/itees.2025.01.058
ISSN 3083-7502

освітнього процесу, захист конфіденційних даних і стійкість інформаційної інфраструктури до зростаючого спектра кіберзагроз. Система кіберзахисту ЗВО поєднує організаційні, технічні та аналітичні заходи, спрямовані на виявлення вразливостей, запобігання атакам та мінімізацію наслідків інцидентів. Її розвиток регламентується положеннями національної стратегії кібербезпеки України та міжнародними стандартами. Інформаційна інфраструктура ЗВО є складною функціональною системою. Вона включає освітні, адміністративні та ресурсні компоненти. Її стійке функціонування визначається ефективною взаємодією всіх підсистем, застосуванням релевантних протоколів захисту та впровадженням релевантних архітектур безпеки. Саме тому в статті запропоновано модель управління інформаційними потоками ЗВО та ресурсами із застосуванням нейромережевих технологій та інтелектуальних систем підтримки рішень.

Аналіз останніх досліджень та публікацій. Різні дослідники по-різному підходили до вивчення проблематики кіберзахисту у ЗВО, зокрема в аспекті впровадження та використання технологій і систем захисту. Цим питанням присвячені праці А. Андрощука, В. Афанасьєва, В. Григи, С. Іванової, О. Дубача, О. Косенка, М. Шишкіної, Ю. Носенка, Л. Забродської, В. Кременя, Б. Одягайла, П. Орлова, Л. Фішмана, С. Лондаря, О. Бринюка, С. Дворецької, О. Шпака, В. Лужецького, О. Білика та інших науковців.

У процесі синтезу ефективної системи кіберзахисту інформаційної інфраструктури ЗВО ключове значення має використання методів моделювання, які дають можливість відобразити складні інформаційні процеси, оцінити потенційні загрози та спрогнозувати наслідки впровадження новітніх технологій. Зокрема, цікавим є підхід, запропонований А. Прусом [1, с. 58–59], який розглядає математичне моделювання як «лінзу реального світу» та виділяє чотири групи компетенцій, які визначають якість цього процесу.

Перша група стосується глибокого розуміння проблеми, формування реалістичних припущень і відокремлення релевантної інформації від другорядної. Це необхідно для аналізу актуальних кіберзагроз. Друга передбачає побудову математичної моделі, спрощення складних процесів і застосування візуалізації для відображення архітектури інфраструктури та її вразливостей. Третя група компетенцій орієнтована на інтерпретацію результатів моделювання у реальних умовах функціонування ЗВО. Четверта - на перевірку адекватності моделі, її гнучкість і здатність адаптуватися до змін кіберсередовища.

Отже, використання моделювання у сфері кіберзахисту ЗВО варто розглядати не лише як технічний інструмент, а як комплексну компетентнісну діяльність, що охоплює етапи аналізу, формалізації, інтерпретації та критичного осмислення результатів.

Мета дослідження. Впровадження інтелектуальних систем у внутрішню інфраструктуру ЗВО відкриває можливості для глибокої трансформації процесів управління ресурсами, організації навчального процесу та обслуговування користувачів. Завдяки застосуванню багаторівневої автентифікації, персоналізованого доступу до сервісів і постійному збору аналітичних даних формується динамічне цифрове середовище, яке здатне адаптуватися до індивідуальних потреб кожного учасника освітнього процесу – викладача чи адміністративного працівника.

Матеріали і методи дослідження. Інформаційна інфраструктура ЗВО включає сукупність інформаційних систем, засобів комунікації, користувачів, баз даних, серверів, шлюзів та систем контролю доступу. Для підвищення її стійкості можуть використовуватися сучасні криптографічні протоколи (AES-256, SSL/TLS), які зменшують ризики кібератак. Водночас стабільність функціонування вимагає постійного моніторингу стану систем, регулярної перевірки вразливостей, своєчасного оновлення захисних механізмів, а також дотримання правил кібергігієни. Серед ключових практик - зміна й генерація надійних паролів, своєчасне блокування підозрілих користувачів, повідомлення про спроби несанкціонованого доступу, аналіз інцидентів і прогнозування їх наслідків.

Для підсилення зазначених процесів дедалі частіше застосовуються інструменти штучного інтелекту. Вони дозволяють здійснювати глибокий аналіз даних, створювати прототипи можливих кібернападів і прогнозувати їх наслідки за допомогою нейромережевих

технологій. Це відкриває перспективи для побудови адаптивних систем кіберзахисту, здатних до самонавчання та оперативного реагування на нові загрози.

Результати дослідження та їх обговорення.

Організаційні підходи до захисту інформаційної інфраструктури закладу вищої освіти. Для забезпечення високої якості освітніх послуг, проведення наукових досліджень, ефективного управління та збереження конкурентоспроможності на ринку, ЗВО мають гарантувати викладачам, дослідникам, співробітникам і здобувачами вищої освіти (далі ЗдВос) надійний та безперервний доступ до власного цифрового середовища. Це середовище формується інформаційною інфраструктурою, яка охоплює цифрові платформи, комунікаційні мережі, системи обробки та передавання даних, а також засоби кіберзахисту.

Водночас цілісність функціонування ЗВО та рівень довіри з боку всіх зацікавлених сторін безпосередньо залежать від здатності університету забезпечити кібербезпеку, конфіденційність інформації та стійкість до кібератак. Саме тому концепція кіберстійкості набула статусу стратегічного пріоритету для ЗВО.

Підтримання безперервності, надійності й безпеки академічних та адміністративних процесів зумовлює зростаючу залежність ЗВО від комплексних інформаційних інфраструктур. Вони включають адміністративні, освітні та ресурсні системи. Водночас така залежність підвищує рівень вразливості до широкого спектра кіберзагроз. Тому першочерговим завданням є ідентифікація та класифікація стрижневих елементів цифрового середовища, що дозволить розробити гнучкі та ефективні заходи кіберзахисту.

У таблиці 1 наведено основні компоненти інформаційної інфраструктури ЗВО.

Таблиця 1 – основні компоненти інформаційної інфраструктури ЗВО

Категорія систем	Приклади компонентів	Основні функції
Ресурсні системи	Сервери, сховища даних, мережеві шлюзи, системи резервного копіювання, хмарні сервіси.	Забезпечення обробки, зберігання та захисту даних; підтримка обчислювальних ресурсів і комунікаційних сервісів.
Адміністративні системи	Системи управління документообігом, кадрові та фінансові системи, електронний деканат, інформаційні портали для співробітників.	Підтримка управлінських і організаційних процесів, доступ до адміністративної інформації, автоматизація внутрішніх процедур.
Освітні системи	Системи дистанційного навчання (LMS), електронні бібліотеки, платформи відеоконференцій, наукові бази даних, репозитарії.	Підтримка навчального процесу та досліджень, забезпечення доступу до освітніх ресурсів, організація взаємодії між викладачами та ЗдВос.

Дослідження [2–4] свідчать, що у 2024 році ЗВО стали однією з основних цілей кіберзлочинців. Так 66% опитаних представників повідомили про кібератаки, а 79% зазнали щонайменше одного інциденту. Хоча витік даних траплявся рідше (лише 18% ЗВО офіційно підтвердили такі випадки), загальний вплив атак виявився значним і у багатьох випадках критичним. Найсерйознішою загрозою залишається програмне забезпечення-вимагач: більшість постраждалих університетів сплачували до 122% від початкових вимог зловмисників, а середній розмір викупу сягав 5,85 млн доларів США, що є третім за величиною показником серед усіх галузей економіки. Крім того, половина закладів відзначила прямі пошкодження своєї ІКТ-інфраструктури, понад 60% зазнали серйозних операційних і фінансових перебоїв. У 77% випадків дані були зашифровані, а у 95% – зловмисники намагалися отримати доступ до резервних копій, що значно ускладнювало відновлення [2–4].

DOI:/10.31548/itees.2025.01.058
ISSN 3083-7502

В умовах таких викликів постає необхідність чітко визначити принципи, на яких повинна базуватися комплексна система кібербезпеки ЗВО, здатна забезпечити всебічний захист ресурсів та інфраструктури. До головних належать [5, с. 140–142]: принцип конфіденційності; принцип цілісності; принцип доступності даних та ресурсів для уповноважених користувачів у потрібний час; принцип постійного моніторингу та оцінювання ефективності системи; принцип дотримання законодавчих норм; принцип підзвітності дій у цифровому середовищі; принцип управління ризиками; принцип підвищення обізнаності користувачів; принцип адаптивної архітектури безпеки; концепція «нульової довіри»; принцип стійкості інформаційних систем; принцип суверенітету даних; а також принцип інтегрованого аналізу внутрішніх і зовнішніх загроз для проактивного реагування.

Отже, архітектура системи кібербезпеки ЗВО має розглядатися як цілісний набір правил, інструментів, процедур і механізмів контролю, що діють у комплексі для захисту інформаційних активів від кібератак. Вона визначає дизайн, методи впровадження, взаємозв'язки та управління компонентами захисту, забезпечуючи доступність, конфіденційність, цілісність і стійкість адміністративних, ресурсних та освітніх систем. Комплексна архітектура кіберзахисту ЗВО повинна відповідати академічній місії та стратегічним цілям цифрової трансформації; ґрунтуватися на ризик-орієнтованому підході з регулярними оцінками ризиків і моделюванням загроз; включати багаторівневі засоби захисту для мережі, кінцевих пристроїв, застосунків і даних; відповідати національним та міжнародним стандартам; бути масштабованою.

Крім того, така система має передбачати інтеграцію SIEM-рішень і аналітичних інструментів для автоматизованого виявлення та реагування на загрози у реальному часі, підтримувати механізми резервування й аварійного відновлення, а також забезпечувати сувору автентифікацію користувачів і пристроїв. Не менш визначальною є інтеграція внутрішніх та зовнішніх джерел даних кіберрозвідки, регулярний аудит і перевірка відповідності, чіткий розподіл ролей та відповідальності. Нарешті, система повинна гарантувати суверенітет даних закладу, зберігаючи їх конфіденційність і водночас підтримуючи безпечний обмін інформацією, командну роботу та наукові дослідження.

Процес формування комплексної системи кібербезпеки ЗВО передбачає послідовне проходження низки етапів, кожен з яких має власні завдання та очікувані результати.

1) *Попередня оцінка та стратегічне планування.* На цьому етапі здійснюється початкове розуміння поточного рівня кіберзахисту ЗВО та визначаються стратегічні орієнтири для побудови архітектури. До пріоритетних завдань належать: класифікація критичних компонентів інформаційної інфраструктури (освітні, адміністративні та ресурсні системи), аналіз чинних політик і технологій захисту, вивчення нормативних вимог, а також визначення цілей кібербезпеки відповідно до місії та цифрової стратегії університету. Реалізація цього етапу часто ускладнюється недостатньою підтримкою з боку керівництва, відсутністю повного реєстру цифрових активів, фрагментованістю системи управління, обмеженими ресурсами, слабкою обізнаністю щодо регуляторних норм, а також відсутністю офіційної системи управління кібербезпекою.

2) *Оцінка ризиків та моделювання загроз.* Метою цього етапу є ідентифікація, аналіз і пріоритезація потенційних кіберзагроз та вразливостей, властивих середовищу ЗВО. Основні завдання включають: проведення комплексної оцінки ризиків для всіх систем університету, розроблення моделі внутрішніх і зовнішніх загроз, визначення ймовірності та наслідків можливих інцидентів, виявлення зон підвищеної небезпеки й формування реєстру ризиків. Серед чинників, що ускладнюють цей процес, варто відзначити відсутність стандартизованих методик аналізу ризиків, недостатнє документування попередніх інцидентів, слабке врахування специфічних для освіти кіберзагроз, занижену увагу до внутрішніх ризиків, залежність від застарілих моделей, а також обмежений доступ до даних у режимі реального часу.

3) *Архітектурне проєктування та створення «каркасу» системи.* Цей етап передбачає розроблення багаторівневої та структурованої системи кіберзахисту, яка визначає як

функціональні, так і технічні компоненти безпеки. До центральних завдань належать: проєктування багаторівневої моделі захисту (мережевої, програмної, даних, ідентифікації та кінцевих точок), визначення доменів безпеки та політик доступу, інтеграція основних безпекових технологій, застосування базових принципів захисту, а також забезпечення відповідності міжнародним стандартам і кращим практикам. Виконання цього етапу може бути ускладнене дефіцитом фахівців із архітектурного проєктування систем безпеки, недостатнім залученням зацікавлених сторін, нечіткістю у розмежуванні прав доступу, залежністю від одного постачальника технологій, відсутністю належної документації, а також неврахуванням перспектив масштабування та модульності архітектури.

4) *Впровадження та інтеграція.* На цьому етапі здійснюється практичне розгортання інструментів і політик кібербезпеки відповідно до затвердженої архітектури та інституційних вимог. Основні завдання включають закупівлю, налаштування та інтеграцію технологічних рішень, впровадження систем управління ідентифікацією та доступом, застосування механізмів шифрування, автентифікації, моніторингу та захисту кінцевих точок, а також встановлення протоколів реагування на інциденти та резервного копіювання. Суттєвим є також узгодження дій між усіма підрозділами університету. Основні виклики цього етапу пов’язані з операційними перебоями у навчальних та адміністративних процесах, недостатньою координацією між ІТ-персоналом і підрозділами ЗВО, проблемами сумісності нових і застарілих систем, неповною конфігурацією інструментів безпеки, затримками у закупівлях, недостатнім тестуванням перед впровадженням та відсутністю стратегії управління системними змінами.

5) *Тестування, перевірка та оптимізація.* Мета цього етапу – оцінити функціональність, надійність та ефективність впровадженої архітектури кібербезпеки. До основних завдань належать проведення тестувань на проникнення, аудитів безпеки, перевірка відповідності внутрішнім політикам і зовнішнім стандартам, аналіз журналів інцидентів, оцінювання поведінки системи в умовах навантаження, виявлення слабких місць і оптимізація конфігурацій. Виконання цього етапу може ускладнюватися обмеженими ресурсами для повномасштабного тестування, небажанням планувати простої в навчальний час, відсутністю чітких показників ефективності, використанням застарілих інструментів моніторингу, низькою готовністю до впровадження змін за результатами перевірки та недостатнім залученням незалежних аудиторів.

6) *Навчання та підвищення обізнаності користувачів***.** Завдання цього етапу полягає у формуванні культури кібергігієни шляхом систематичного навчання персоналу, далі ЗдВос та адміністраторів правилам безпеки й відповідальності у цифровому середовищі. Реалізація включає організацію тренінгів, підготовку інструкцій і політик у доступних форматах, заохочення повідомлень про підозрілу активність. Основні труднощі пов’язані з низькою мотивацією до участі у тренінгах, використанням застарілих або одноразових програм навчання, слабкою інтеграцією знань у корпоративну культуру, відсутністю системного контролю дотримання політик безпеки та механізмів зворотного зв’язку для оцінювання ефективності навчання.

7) *Безперервний моніторинг та управління життєвим циклом.* Цей етап спрямований на підтримання постійної ефективності, стійкості й адаптивності архітектури кібербезпеки. Він включає використання інструментів безперервного моніторингу, оновлення політик та баз знань про загрози, регулярний перегляд архітектури з урахуванням нових ризиків і змін в інституційному середовищі, проведення аудитів і перевірок відповідності. Основними проблемами можуть бути обмежена видимість мережевої активності в реальному часі, недостатня інтеграція з системами виявлення загроз, дефіцит ресурсів для оновлення інфраструктури, надмірна залежність від ручного моніторингу, затримки у реагуванні на інциденти, фрагментарність систем нагляду та відсутність регулярних аудитів безпеки.

Тоді модель оцінки рівня кіберстійкості ЗВО подамо багатокритеріальну оптимізацію. Нехай

x$\in X \subseteq R^m$ – вектор рішень (конфігурація системи кіберзахисту),

DOI:/10.31548/itees.2025.01.058
ISSN 3083-7502

$F1(x)$ – рівень надійності (ймовірність відбиття атаки, max),
$F2(x)$ – продуктивність системи (час відгуку, max),
$F3(x)$ – вартість впровадження та обслуговування (min),
$F4(x)$ – масштабованість та гнучкість (max).
Задача:

$$\max_{x \in X} \left(F1(x), F2(x), F3(x), \quad \min_{x \in X} F4(x) \right). \tag{1}$$

Розв'язки оцінюються за принципом Парето-оптимальності:

$$x \in X, \nexists y \in X : F(y) \succ F(x^*). \tag{2}$$

Тобто, формалізуючи задачу кіберстійкості ЗВО, доцільно розглядати її як багатокритеріальну оптимізацію, де одночасно враховуються показники надійності, продуктивності, вартості та масштабованості. У цьому випадку оптимальними є такі конфігурації архітектури кіберзахисту, які належать до множини Парето-ефективних рішень, що дозволяє збалансувати суперечливі вимоги різних груп користувачів і обмеження ресурсів.

Моделювання інформаційної інфраструктури ЗВО

Одним із головних викликів у створенні ефективної системи кіберзахисту ЗВО є відсутність уніфікованих стандартів щодо структурування даних. Кожен заклад має власну специфіку, внутрішні правила й регламентовані процедури, що потребує використання додаткових інтерфейсів, запитів і засобів комунікації для забезпечення взаємодії між підсистемами та контрольованого доступу до ресурсів. Сучасні засоби передавання інформації вже функціонують у межах протоколів безпеки, проте цього недостатньо для забезпечення комплексної кіберстійкості.

У моделюванні систем кіберзахисту слід враховувати, що стійкість інформаційних ресурсів до атак, несанкціонованого доступу чи руйнування є визначальним критерієм їхньої надійності. Це потребує визначення організаційних заходів безпеки, проєктування захищених каналів зв'язку, обов'язкової автентифікації користувачів, а у випадках підвищеного ризику – багатофакторної ідентифікації (SMS, мобільні застосунки, електронні ключі, КЕП, біометричні дані тощо). Важливим інструментом підвищення кіберстійкості є впровадження ефективних криптографічних протоколів (AES-256, SSL/TLS), однак навіть вони не гарантують ефективності без постійного моніторингу, своєчасного оновлення захисних механізмів, аналізу інцидентів та впровадження заходів реагування.

Провідну роль у цьому процесі відіграють інструменти штучного інтелекту, які дозволяють моделювати прототипи кібератак, прогнозувати їх наслідки та формувати гнучкі механізми реагування. Використання нейромережевих технологій розширює можливості класичного моделювання, що узгоджується з концепцією моделювання як інструменту пізнання складних систем, запропонованою Прусом А. [1]. Поєднання інтелектуальних алгоритмів із базами знань створює умови не лише для аналізу даних, але й для їх фільтрації, генерації та добору ефективних варіантів розвитку інформаційних систем. Це забезпечує безперервність, гнучкість та прогнозованість функціонування цифрової інфраструктури.

Нейронні мережі мають стратегічне значення й для управління освітнім процесом. Вони здатні адаптувати інфраструктуру ЗВО до нових викликів, сприяти цифровій трансформації, розширювати спектр освітніх послуг, підвищувати рівень індивідуалізації та доступності освітніх ресурсів. Формування інтелектуальної системи на основі нейромереж і баз знань створює потужне середовище підтримки прийняття рішень, що включає функції автентифікації користувачів, адаптацію контенту, маршрутизацію даних до зовнішніх інфраструктур – цифрового міста (Smart City), електронного урядування, міжнародних наукових платформ тощо. Це суттєво розширює межі функціонування від локального рівня до національного й глобального освітнього кіберпростору [10].

Тобто, інформаційна інфраструктура ЗВО розглядається як динамічна система, здатна до адаптації, інтеграції з міжнародними стандартами, забезпечення безпечного обміну даними й

розгортання ізольованих підсистем для наукових експериментів, випробування захисних протоколів та моделювання кіберзагроз (рис. 1). Це визначає її подальший сталий розвиток у напрямі стійкості.

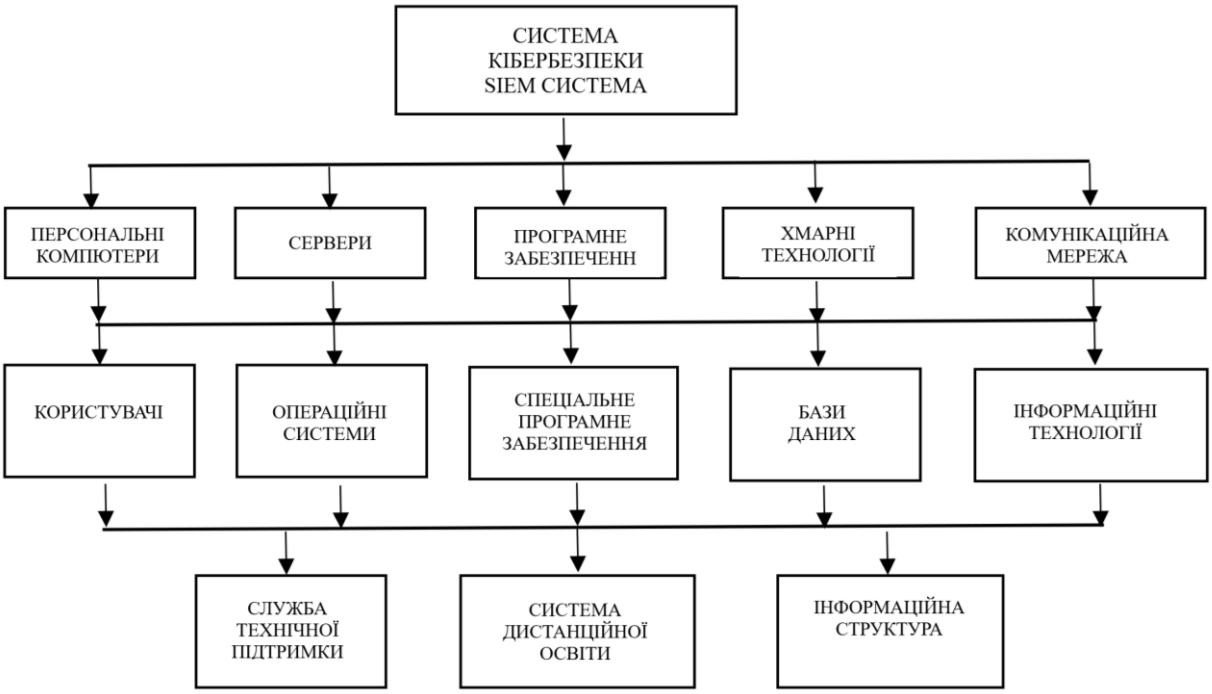


Рисунок 1 – Ієрархічна структура системи кібербезпеки ЗВО
Джерело: розроблено авторами

Окрім ресурсів, які ЗВО розподіляє між своїми користувачами, у його розпорядженні є кероване інформаційне середовище, яке забезпечує захищене з'єднання та контроль за потоками даних. У цьому середовищі функціонують шлюзи та файрволи, які виконують розподіл навантаження між ресурсами в межах інформаційної інфраструктури ЗВО.

На рисунку 2 представлено модель інформаційних ресурсів ЗВО та оптимальних шляхів комунікації між ними. Незважаючи на велику кількість комп'ютерної техніки, мережевого обладнання, баз даних і зовнішніх інформаційних ресурсів, учасники цієї інфраструктури стикаються з низкою обмежень щодо доступу до ресурсів. Відсутня уніфікація в роботі інформаційних систем, що ускладнює їхнє адміністрування: кожна система обслуговується окремо, а доступ до ресурсів надається індивідуально. При зміні організаційної структури, посадових обов'язків або ролей виникають труднощі з оновленням прав доступу в різних системах, що свідчить про неузгодженість та фрагментованість автоматизованих систем у межах єдиної інфраструктури.

Модель також демонструє, що електронна мережа ЗВО охоплює весь кампус, є складною в адмініструванні та надає можливість кожному користувачу взаємодіяти з нею відповідно до свого рівня доступу. Життєздатність та функціонування мережі регулюються нормативно-правовими актами — як державними, так і внутрішніми документами ЗВО.

Система захисту інформаційних ресурсів будується з урахуванням специфіки діяльності ЗВО. Інфраструктура включає фізичну комунікаційну мережу, мережеві комутатори, бездротові точки доступу, комп'ютери, сервери, FireWall та підключення до Інтернету. Для захисту окремих компонентів використовуються багаторівневі керовані комутатори з вбудованими функціями захисту, що дозволяє створити надійні засоби оборони інформаційних ресурсів.

В наших дослідженнях зокрема увагу приділено безпечному доступу до мережі Інтернет. Всі елементи інфраструктури з'єднані дротовими та бездротовими комунікаційними засобами,

DOI:/10.31548/itees.2025.01.058
ISSN 3083-7502

а ресурси доступні незалежно від фізичної присутності працівників на робочому місці. Сервери з критичними ресурсами розміщені всередині корпоративної мережі та адмініструються фахівцями ЗВО. Для віддаленого та захищеного доступу до ресурсів застосовуються VPN-з'єднання.

Зауважимо, що на моделі, поданій на рис. 2, не деталізовано та не структуровано механізми управління окремими компонентами інформаційної інфраструктури ЗВО. Водночас рисунок також демонструє модель системи захисту інформаційних ресурсів ЗВО.

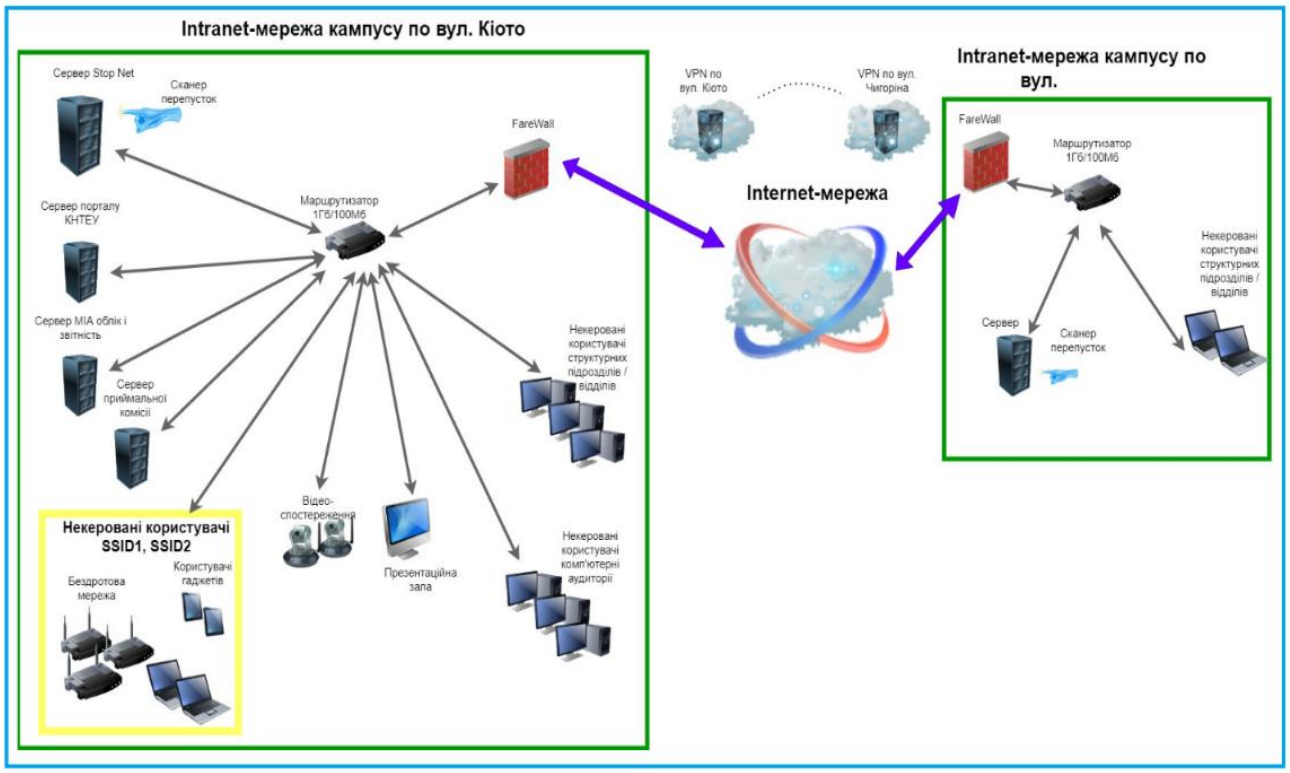


Рисунок 2 – Модель інформаційної інфраструктури ЗВО (засобів комунікації, ресурсів ЗВО) Джерело: [22]

На сьогодні впроваджено модель програмного захисту з функціями аудиту навантаження на ресурси інфраструктури. Основною – і водночас єдиною — перевагою цієї моделі є можливість ручного управління окремими автоматизованими освітніми системами через прямий доступ. Однак ця перевага водночас стає значним недоліком: для підтримки такої системи потрібні кваліфіковані працівники, які володіють навичками адміністрування різних автоматизованих систем, а також здатні узгоджувати їхню роботу з іншими внутрішніми та зовнішніми інформаційними системами. Це робить інфраструктуру ресурсоємною і дорогою в обслуговуванні.

Отже, цифрове управління ЗВО є лише частково автоматизованим і потребує значної кількості технічних узгоджень. Крім того, ускладнено побудову ефективної системи кіберзахисту через відмінності в налаштуваннях та розподілах прав доступу в різних системах. Відсутність єдиного підходу до управління базами даних, автоматизованими системами та механізмами доступу ускладнює аналіз інформації та оперативну зміну прав користувачів відповідно до їхньої присутності — очної або дистанційної.

Більшість процесів управління в ЗВО виконується вручну, що ускладнює моніторинг і контроль виконання завдань через відсутність ефективного зворотного зв'язку. Для підтримки функціонування інфраструктури необхідна велика кількість IT-фахівців із різними технічними компетенціями. Доступ до інформаційних ресурсів може бути нестабільним через технічні

чинники, зокрема збої в роботі комутаторів або відсутність електроживлення, що напряму впливає на функціонування всієї інфраструктури.

Усі пристрої (гаджети, ноутбуки, бездротове обладнання) пов'язані між собою через комутаційне обладнання, яке виконує постійний моніторинг та контроль мережі. Для формування повної картини функціонування інформаційної системи ЗВО необхідно проаналізувати всі елементи інфраструктури, зокрема бази даних. При цьому фіксується відсутність належного рівня захисту персональних даних користувачів.

Для забезпечення узгодженості інформаційних потоків необхідне втручання оператора — зокрема, для формування пропозицій, прийняття рішень і вирішення типових запитів у діалоговому режимі. Одним із суттєвих недоліків є також відсутність чіткої інформації про фізичних користувачів ЗВО в системі.

Модель інформаційної інфраструктури закладу вищої освіти: інтелектуальний доступ, аналітика потреб і кібербезпека

Усі вищезазначені недоліки можна усунути шляхом впровадження інтелектуального центру управління інформаційною інфраструктурою ЗВО, побудованого на основі централізованої системи керування з використанням нейромережевих алгоритмів. Запропонована модель інформаційної інфраструктури передбачає наявність такого інтелектуального центру, який завдяки нейромережам здатен оперативно й комплексно аналізувати стан інфраструктури та пропонувати оптимальні рішення щодо управління ресурсами та усунення проблем.

На рис. 3 представлено модель інформаційної інфраструктури ЗВО з урахуванням функціонування інтелектуального центру управління, реалізовану на прикладі Державного торговельно-економічного університету.

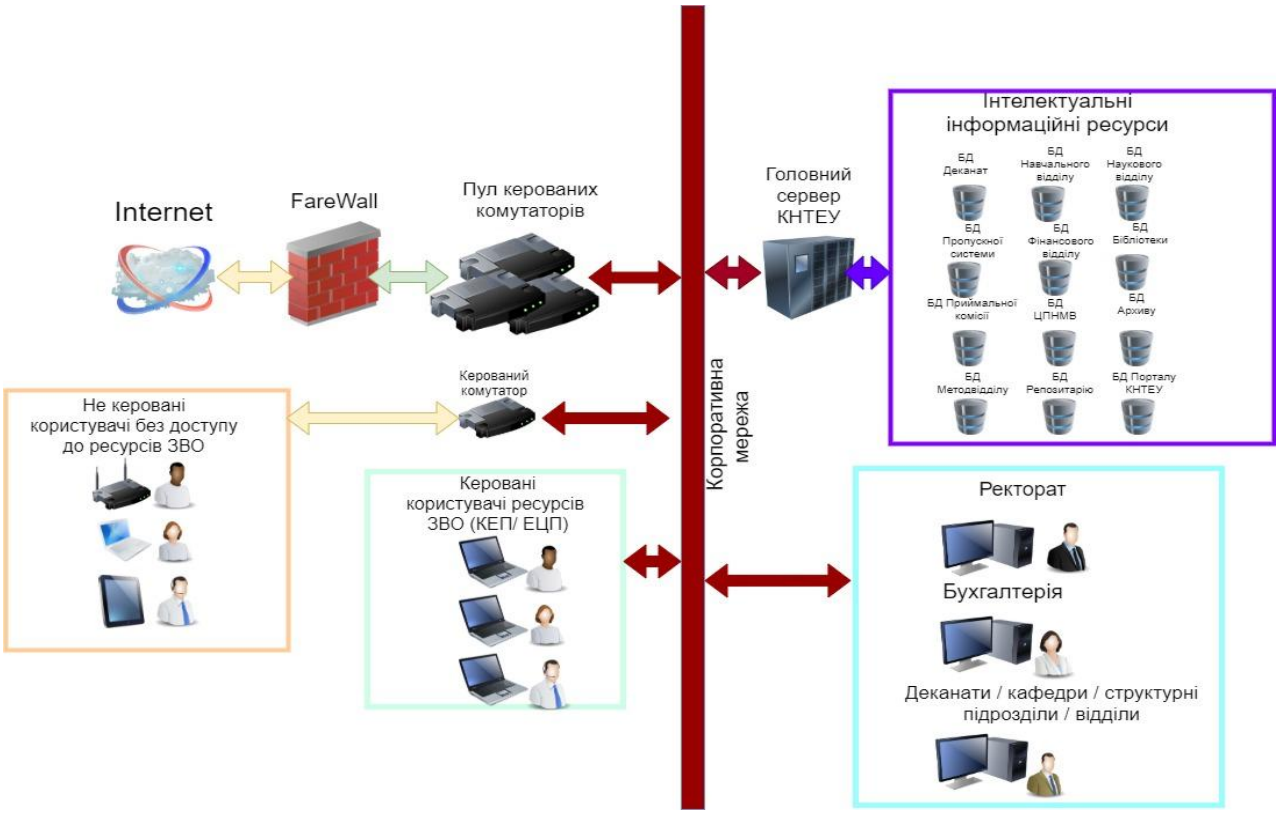


Рисунок 3 – Модель інформаційної інфраструктури ЗВО (фізичні з'єднання та зміни в комутаційній архітектурі ЗВО). Джерело: розроблено авторами

Основна ідея цієї моделі полягає у впровадженні контрольованих процесів керування інформаційними потоками та інтелектуального розподілу ресурсів відповідно до актуальних

DOI:/10.31548/itees.2025.01.058
ISSN 3083-7502

запитів користувачів. Система дозволяє здійснювати аналітичну оцінку потреб у ресурсах, перевіряти їх обґрунтованість і забезпечувати своєчасний доступ у разі необхідності. У межах бездротової мережі ресурси поділяються на сегменти з контрольованим або гостьовим доступом.

На етапі ідентифікації користувача система контролю доступу формує індивідуальний набір ресурсів, необхідних для ефективної діяльності. Якщо користувач потребує підключення до додаткових інформаційних систем, доступ надається автоматично без переривання з'єднання з основним середовищем.

Обмін даними між інформаційними системами здійснюється через електронні запити, які обробляються інтелектуальним центром. Центр структурує отриману інформацію та забезпечує її інтеграцію в інші системи відповідно до змісту запиту. Розподіл ресурсів здійснюється динамічно з урахуванням навантаження та потреб користувачів, що дає змогу ефективно управляти базами даних, прогнозувати пікові навантаження та оптимізовувати їх через перерозподіл або стиснення даних.

Система також підтримує автентифікацію користувачів і надання доступу до локальних або віртуальних мереж із різними рівнями привілеїв. У разі дистанційної роботи передбачене використання електронних цифрових ключів для забезпечення безпечного з'єднання. Такий підхід дозволяє максимально ефективно використовувати інформаційні ресурси навіть за умов змінного навантаження на окремі елементи інфраструктури ЗВО.

На рис. 3 представлено комунікації, фізичні з'єднання та зміни в комутаційній архітектурі, що виникають у результаті впровадження інтелектуального центру, який здійснює повний контроль над розподілом ресурсів, підвищує ефективність їх використання, здійснює аналітичну обробку даних і формує рекомендації для прийняття рішень у межах інформаційної інфраструктури ЗВО.

Порівняльний аналіз моделі на рис. 3 засвідчує якісні зміни в управлінні інформаційними потоками, розподілі ресурсів між різними категоріями користувачів та функціонуванні системи автентифікації.

Інтелектуальний центр виконує низку функцій, зокрема: прогнозування наслідків авторизації користувачів, управління доступом до комп'ютерних ресурсів, баз даних, автоматизованих систем, оптимізацію навантаження на інтернет-ресурси ЗВО, повідомлення про відмови в роботі, а також виявлення спроб несанкціонованого доступу. Такий підхід забезпечує можливість оперативного аналізу інцидентів і реалізації коригувальних заходів з боку адміністраторів інформаційної інфраструктури ЗВО.

Інтеграція інтелектуальної системи до структури інформаційної інфраструктури ЗВО дозволить забезпечити її взаємодію з усіма автоматизованими підсистемами та системою кібербезпеки. До її функцій належить управління адміністративними правами користувачів, що дає змогу, на основі опису параметрів прототипу користувача, призначати типи доступу та регламентувати їх у різних інформаційних автоматизованих системах із можливістю подальшої модифікації.

Крім того, значущим елементом є фіксація фізичної присутності користувача. Зчитування перепустки через систему контролю доступу відображає присутність у загальній системі, що може додатково підтверджуватися засобами відеоспостереження шляхом зіставлення обличчя користувача з фотографією.

Також передбачена можливість застосування альтернативних засобів біометричної ідентифікації – зокрема, за голосом або відбитками пальців. Утім, реалізація таких підходів потребує значних фінансових ресурсів та модернізації контрольно-пропускних пунктів університету, що наразі є обмежуючим фактором.

Після підтвердження фізичної присутності користувача на території кампусу, інтелектуальна система автоматично надає йому доступ до всіх доступних ресурсів відповідно до його ролі та рівня прав. Водночас система здатна адаптуватися до індивідуальних звичок і потреб: вона може враховувати вподобання в харчуванні, щоб оптимізувати приготування їжі,

повідомляти про нові надходження у бібліотеці за тематикою дослідження, зміни в розкладі занять, навантаженні чи анонси наукових подій та зустрічей.

Також запропонована інтелектуальна система може автоматично фіксувати фактичну присутність співробітників на робочому місці, передаючи ці дані у фінансово-економічну систему. Користувачі можуть отримувати нагадування про майбутні зміни у правах доступу, навчання з підвищення кваліфікації та інші внутрішні оновлення.

Для забезпечення повноцінного кібернетичного захисту рекомендуємо налаштувати, адаптувати та використовувати SIEM системи, які постійно проводять збір, обробку та аналіз подій безпеки, виявляти загрози у реальному часі, проводити аналіз та управління безпекою, а також проводити розслідування інцидентів. Цікава така система тим, що конфігурується та налаштовуються для всієї інформаційної інфраструктури ЗВО. За браком фахівців з кіберзахисту рекомендуємо використовувати систему, яку можна використати й ІТ фахівцями з проведеними певними специфічними навчаннями. Така система автоматично оновлюється, використовує автоматизований аудит інформаційної інфраструктури ЗВО. Такі системи дозволяють автоматизувати фільтрацію подій, виявлення порушень безпеки, сповіщення подій, аналіз та управління, отримання сповіщень в результаті виявлення загроз чи прогнозування кібератак на інформаційні ресурси ЗВО.

Інтелектуальна система, провівши автентифікацію, забезпечує безперешкодний доступ до необхідних платформ та сервісів без додаткових дій з боку користувача. За потреби, адміністратори мають змогу індивідуально або колективно змінювати рівень доступу, з обов’язковим підтвердженням змін.

Система кіберзахисту при цьому виконує функцію постійного моніторингу безпеки, оперативно повідомляючи відповідальних осіб про виявлені загрози, спроби несанкціонованого втручання чи атаки, дії адміністратора з їх нейтралізації, а також прогнозує можливі наслідки. Після ліквідації загроз система оцінює терміни відновлення стабільної роботи інформаційної інфраструктури ЗВО, забезпечуючи безперервність освітнього процесу.

Після підтвердження фізичної присутності користувача на території кампусу через системи автентифікації (наприклад, біометричні сканери, RFID-мітки або мобільні додатки), інтелектуальна система автоматично активує персоналізований профіль доступу. Такий механізм дозволяє одразу використовувати інформаційні ресурси ЗВО – навчальні платформи, бази даних, адміністративні сервіси, лабораторії, бібліотеки чи харчоблоки – відповідно до ролі користувача (ЗдВос, викладач, науковець, технічний працівник) та затвердженого рівня прав.

Система виконує контекстний аналіз попередніх дій користувача, історії взаємодії з сервісами, відвідуваності заходів, запитів до бібліотеки чи меню в їдальні, тим самим дозволяючи прогнозувати потреби та підлаштовувати середовище під кожного індивідуально. Наприклад, модуль харчування може проаналізувати звички користувача й оптимізувати обсяг порцій, зменшуючи харчові втрати та витрати. Освітній модуль надсилає повідомлення про нові бібліотечні надходження згідно з науковим профілем користувача, а також повідомляє про зміни в розкладі, перенесення занять, появу вільних слотів для консультацій, наукових подій, гостьових лекцій та стипендіальних можливостей (рис. 4).

Адміністративна частина системи забезпечує автоматичну фіксацію робочого часу співробітників у форматі очної присутності, синхронізуючи ці дані з бухгалтерськими та кадровими модулями. Також система здатна заздалегідь інформувати про планові зміни у правах доступу до ресурсів, проходження обов’язкових атестацій, тренінгів чи заходів із підвищення кваліфікації.

Автоматизоване управління правами доступу забезпечує прозору та гнучку модель регулювання: при зміні статусу користувача (наприклад, переведення ЗдВос на іншу форму навчання або підвищення посади співробітника), адміністратор може швидко оновити профіль доступу як на індивідуальному, так і на груповому рівні з миттєвим підтвердженням змін.

DOI:/10.31548/itees.2025.01.058
ISSN 3083-7502

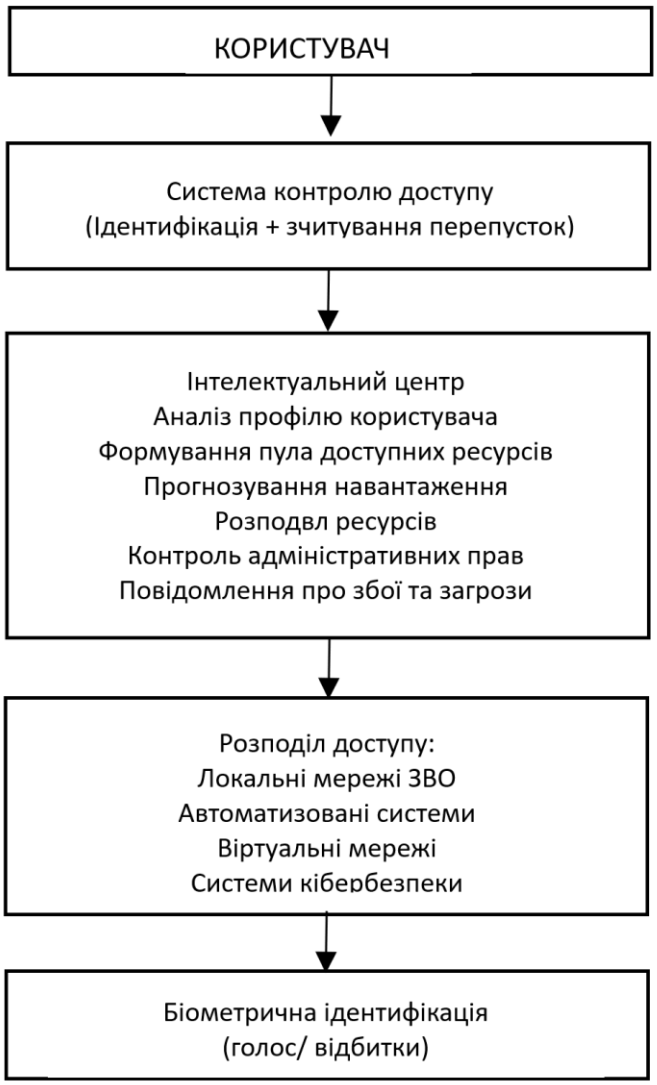


Рисунок 4 – Схема взаємодії інтелектуальної системи в інформаційні інфраструктурі ЗВО. Джерело: розроблено авторами

Визначальну роль у функціонуванні всієї системи відіграє комплекс кіберзахисту, який здійснює цілодобовий моніторинг усіх вузлів цифрової інфраструктури. Він здатен виявляти аномалії, блокувати потенційні вторгнення, реєструвати підозрілі дії, включно з втручанням адміністраторів, та оперативно інформувати відповідальних осіб. Система автоматично формує прогнози щодо шкоди від атак, надає рекомендації щодо нейтралізації загроз, оцінює терміни відновлення стабільного функціонування інформаційної інфраструктури ЗВО (зокрема інформаційної інфраструктури Державного торговельно-економічного університету та Національного університету біоресурсів та природокористування України) і забезпечує сталий перебіг освітнього процесу навіть за умов надзвичайних ситуацій.

Висновки і перспективи. Впровадження інтелектуальних систем у внутрішню інфраструктуру ЗВО відкриває можливості для глибокої трансформації процесів управління ресурсами, організації навчального процесу та обслуговування користувачів. Завдяки застосуванню багаторівневої автентифікації, персоналізованого доступу до сервісів і постійному збору аналітичних даних формується динамічне цифрове середовище, яке здатне адаптуватися до індивідуальних потреб кожного учасника освітнього процесу — ЗдВос, викладача чи адміністративного працівника.

Функціональність такої системи виходить далеко за межі звичайного надання доступу до ресурсів: вона виконує прогнозно-аналітичні завдання, аналізує поведінкові шаблони

користувачів для ефективного розподілу ресурсів (наприклад, у використанні простору кампусу або бібліотечних фондів), автоматизує повсякденні адміністративні процеси — такі як контроль відвідуваності або розрахунок заробітної плати, — і підвищує ефективність управління завдяки централізованій системі прав доступу.

Головним компонентом цієї цифрової екосистеми є система кібербезпеки. Вона не лише забезпечує захист даних, підтримуючи їхню конфіденційність і цілісність, а й створює умови для безперервного функціонування освітнього процесу в умовах кіберзагроз. Завдяки можливості проактивного виявлення загроз, блокування атак та прогнозування їх наслідків, кіберзахисні механізми значно підвищують цифрову стійкість ЗВО. Вважаємо що провідну роль відіграють SIEM-системи, які забезпечують гнучке та комплексне реагування на події безпеки в мережі ЗВО.

Загалом, інтеграція таких інтелектуальних систем у цифрову інфраструктуру університету є не лише відповіддю на виклики цифрової трансформації освіти, а й основою для формування освітнього простору, який поєднує безпеку, гнучкість, ефективність та здатність до адаптації в умовах постійних змін.

Список використаних джерел

1. Прус А. (2023) Математичне моделювання як лінза реального світу. Physical and Mathematical Education. 38(4), 56–61. URL: https://doi.org/10.31110/2413-1571-2023-038-4-008
2. Sophos. (2024). The state of ransomware 2024. https://www.sophos.com/en-us/content/state-of-ransomware.
3. BlueVoyant. Cybersecurity in higher education. Retrieved February 25, 2025, from https://www.bluevoyant.com/resources/cybersecurity-in-higher-education.
4. Zavhorodnya, E., Shestak, Y., & Kryvoruchko, O. (2025). Digital risk management in higher education. In Modern achievements and prospects of socio-economic development (pp. 138–143). Eastern European Center for Scientific Research. https://researcheurope.org/wp-content/uploads/2025/05/re-16.05.25.pdf.
5. Shestak, Y., & Zavhorodnya, E. (2025). Protection principles of HEIs information infrastructure. In Innovations and their impact on the economy and society (pp. 138–143). Eastern 6. European Center for Scientific Research. https://researcheurope.org/wp-content/uploads/2024/11/re-25.10.24.pdf.
6. El Latif, A. A. A., Maleh, Y., El Affendi, M. A., & Ahmad, S. (Eds.). (2023). Cybersecurity management in education technologies: Risks and countermeasures for advancements in E-learning (1st ed.). CRC Press. https://doi.org/10.1201/9781003369042.
7. Peng, L. (2023). Design of smart campus security management and control platform based on Big Data technology. In Proceedings of the 2022 International Conference on Educational Innovation and Multimedia Technology (EIMT 2022) (pp. 586–595). Atlantis Press. https://doi.org/10.2991/978-94-6463-012-1_65
8. Ni, Q., & Zeng, Y. (2025). Research on smart campus system architecture design and data security protection strategy. Frontiers in Computing and Intelligent Systems, 11(3), Article 98. https://doi.org/10.54097/f6sy7t88.
9. Tahsien, S. M., Karimipour, H., & Spachos, P. (2020). Machine learning based solutions for security of Internet of Things (IoT): A survey. Journal of Network and Computer Applications, 161, 102630. https://doi.org/10.1016/j.jnca.2020.102630.
10. Lakhno, V., Malyukov V., Kryvoruchko, O., Desiatko, A., & Shestak Y. (2020). Smart city technology investment solution support system accounting multi-factories. In R. Silhavy, P. Silhavy, Z. Prokopova (Eds.), Software engineering perspectives in intelligent systems (pp. 1–11). Springer. https://doi.org/10.1007/978-3-030-63322-6_1.
11. Domínguez Bolaño, T., Barral, V., Escudero, C. J., & García Naya, J. A. (2024). An IoT system for a smart campus: Challenges and solutions illustrated over several real world use cases. IEEE

DOI:/10.31548/itees.2025.01.058
ISSN 3083-7502

Access, 12, 104592–104606. https://doi.org/10.48550/arXiv.2403.15395. https://doi.org/10.1016/j.iot.2024.101099.
12. Peng, C. F., Peng, L., & Liu, Y. (2023). Application of Big Data technology in campus security management under the background of information age. Journal of Physics: Conference Series, 2458(1), 012012. https://doi.org/10.1088/1742-6596/1881/2/022097.
13. Zhang, Z., Hamadi, H.A., Damiani, E., Yeun, C.Y., & Taher, F. (2022). Explainable Artificial Intelligence Applications in Cyber Security: State-of-the-Art in Research. IEEE Access, 10, 93104-93139. https://doi.org/10.1109/ACCESS.2022.3204051.
14. Lakhno, V., Lakhno, M., Kryvoruchko, O., Kaminskyi, S., & Makaiev, V. (2024). Automation of DDoS attack investigation in industrial control systems using Bayesian networks on Python. In CEUR Workshop Proceedings on Cybersecurity Providing in Information and Telecommunication Systems II (CPITS-II 2024), pp. 282–287. https://ceur-ws.org/Vol-3826/short18.pdf.
15. Литвинов, О., Філіпенко, Н., Лукашевич, С., & Палкова, К. (2024). Кібербезпека як фактор ефективності закладів вищої освіти. Пропілеї права та безпеки, 5, Article Ключові аспекти кіберзахисту ЗВО та підготовка персоналу, 15-23. https://doi.org/10.32620/pls.2024.5.02.
16. Трофименко, О., Логінова, Н., Сергійчук, М., & Дубової, Ю. (2022). Кіберзагрози в освітньому секторі. Кібербезпека: освіта, наука, техніка, 4(16), 76–84. https://doi.org/10.28925/2663-4023.2022.16.7684 (csecurity.kubg.edu.ua)
17. Ільєнко, А., Ільєнко, С., Яковенко, О., Галич, Є., & Павленко, В. (2024). Перспективи інтеграції штучного інтелекту в системи кібербезпеки. Кібербезпека: освіта, наука, техніка, 1(25), 318–329. https://doi.org/10.28925/2663-4023.2024.25.318329 (csecurity.kubg.edu.ua)
18. Скумін, Т. Ф., & Стасишин, Р. М. (2015). Інтелектуальна система кіберзахисту. Тези доповідей IV Міжнар. наук.-техн. конференції «Актуальні задачі сучасних технологій», ТНТУ (Тернопіль), 53–54. https://elartu.tntu.edu.ua/handle/123456789/11060?locale=it.
19. Костікова, М. В. (2022). Сучасний освітній процес і кібербезпека. Матеріали Всеукр. наук.-практ. Internet конф. (м. Харків, 15–16 листоп.), 57–59. (dspace.khadi.kharkov.ua)
20. Доценко, С. О. (2022). Кібербезпека учасників освітнього процесу в умовах дистанційного і змішаного навчання. Нац. ун-т «Одеська юридична академія». (dspace.hnpu.edu.ua)
21. Dets, D., Barduk, A., & Syvolap, O. (2025). Cybersecurity in the field of open access: Principles for protecting educational and scientific resources. Automation of Technological and Business Processes, 17(1), 17–24. https://doi.org/10.15673/atbp.v17i1.3081.
22. Шестак Я.І. (2022). Моделювання єдиного інформаційного простору закладу вищої освіти. Управління розвитком складних систем, 49, 81–89. URL: https://doi.org/10.32347/2412-9933.2022.49.81-89.

Shestack Yaroslav
PhD, Associate Professor, Department of Software Engineering and Cybersecurity,
State University of Trade and Economics, Ukraine
ORCID: https://orcid.org/0000-0002-5102-9642
E-mail: shestack@knute.edu.ua

Tsiutsiura Svitlana
Doctor of Technical Sciences, Professor, Professor of the Department of Software Engineering and Cybersecurity,
State University of Trade and Economics, Ukraine
ORCID: https://orcid.org/0000-0002-4270-7405
E-mail: svtsutsura@dteu.edu.ua

Kryvoruchko Olena
Doctor of Technical Sciences, Professor, Professor of the Department of Computer Systems, Networks and Cybersecurity,

National University of Life and Environmental Sciences of Ukraine
ORCID: https://orcid.org/0000-0002-7661-9227
E-mail: o.kryvoruchko@nubip.edu.ua

Lakhno Valerii
Doctor of Technical Sciences, Professor, Professor of the Department of Computer Systems, Networks and Cybersecurity,
National University of Life and Environmental Sciences of Ukraine
ORCID: http://orcid.org/0000-0001-9695-4543
E-mail: lva964@nubip.edu.ua

Kasatkin Dmytro
PhD of Pedagogical Sciences, Associate Professor, Head of the Department of Computer Systems, Networks and Cybersecurity,
National University of Life and Environmental Sciences of Ukraine
ORCID: https://orcid.org/0000-0002-2642-8908
E-mail: d.kasatkin@nubip.edu.ua

CYBER RESILIENCE OF UKRAINIAN HIGHER EDUCATIONAL INSTITUTIONS IN A WARFARE CONDITION

Abstract. *The article examines the problem of ensuring cyber resilience of higher education institutions (HEIs) by developing and implementing a comprehensive cyber defense architecture. It is shown that the effectiveness of such a system is determined by the ability to integrate educational, administrative, and resource subsystems, taking into account their interdependence and specifics of functioning. The main risks and consequences of cyber attacks are considered. The principles of building a secure information infrastructure and the criteria that a reliable and effective HEI cybersecurity system must meet are outlined. A model for managing HEI information flows and resources using neural network technologies and intelligent decision support systems is proposed. The results of the study demonstrate the feasibility of using modeling tools to predict threats, optimize resource allocation, and increase the resilience of the educational environment to cyber risks.*

Keywords: *cybersecurity, information infrastructure, development trajectories, cybersecurity systems, infrastructure cyber resilience, neural network technologies, communication networks.*

DOI:10.31548/itees.2025.01.073
ISSN 3083-7502

УДК 004.056.5

Лахно Валерій Анатолійович
доктор технічних наук, професор, професор кафедри комп'ютерних систем, мереж та кібербезпеки,
Національний університет біоресурсів та природокористування України
ORCID: https://orcid.org/0000-0001-9695-4543
E-mail: lva964@nubip.edu.ua

Мамченко Сергій Миколайович
доктор педагогічних наук, професор кафедри комп'ютерних систем, мереж та кібербезпеки,
Національний університет біоресурсів та природокористування України
ORCID**:** https://orcid.org/0009-0006-8743-5606
E-mail: s.mamchenko@nubip.edu.ua

Матієвський Володимир Валерійович
старший викладач кафедри комп'ютерних систем, мереж та кібербезпеки,
Національний університет біоресурсів та природокористування України
ORCID: https://orcid.org/0000-0002-1954-8493
E-mail: m_vv@outlook.com

АСПЕКТИ ВИЯВЛЕННЯ КІБЕРЗАГРОЗ В МЕРЕЖЕВОМУ ТРАФІКУ УНІВЕРСИТЕТУ

Анотація. *Сучасні кібернетичні загрози для телекомунікаційних систем і мереж характеризуються високим ступенем прихованості, адаптивності та різноманітності. Це ускладнює їхнє оперативне виявлення в мережевому трафіку, зокрема, університету. В умовах мінливої структури кібератак традиційні методи, що базуються на сигнатурному аналізі та фіксованих правилах, виявляються недостатньо ефективними для ідентифікації нових або модифікованих загроз. У зв'язку з цим зростає значущість розробки інтелектуальних гібридних підходів. Такі методи здатні аналізувати поведінкові характеристики університетського трафіку та адаптуватися до його змін. У статті представлено метод виявлення кібернетичних загроз, заснований на поєднанні методів ансамблевої кластеризації та баєсівського імовірнісного моделювання. На першому етапі використовується машинне навчання для виділення прихованих поведінкових ознак мережевих з'єднань в університетській мережі на основі різних кластеризаційних алгоритмів. Отримані ембединги поведінки надалі слугують вхідними даними для побудови баєсівської мережі, що описує імовірнісні залежності між параметрами поведінки та ознаками аномальності. Запропонований підхід дозволяє не тільки фіксувати відхилення від нормальної поведінки в трафіку, але й забезпечує інтерпретованість рішень у сфері інформаційної безпеки. Практична цінність методу полягає в його потенціалі для застосування в системах моніторингу мережевого трафіку в корпоративних мережах.*

Ключові слова: *мережевий трафік, мережа, університет, поведінковий аналіз, баєсова мережа, кластеризація, машинне навчання, метод, кібербезпека.*

Вступ. Комп'ютерні мережі об'єктів інформатизації, перебуваючи в стані постійного розвитку, та в міру розвитку інформаційних технологій і посилення залежності бізнес-процесів від роботи мереж, стають дедалі вразливішими до широкого спектру кіберзагроз. Такі кіберзагрози проявляються у вигляді атак різної природи, починаючи від несанкціонованого доступу й закінчуючи складними багатоетапними вторгненнями [1, 2]. Зі збільшенням обсягів мережевого трафіку та ускладненням поведінкових шаблонів користувачів традиційні методи виявлення загроз та аномалій на базі сигнатур і статичних евристик, описані в роботах [2-6], як показано в [6-11], втрачають свою ефективність. Ця обставина обумовлена не тільки високою динамікою кібернетичних загроз, а й необхідністю оперативної адаптації до нових типів атак. А такі нові атаки, часто не мають заздалегідь визначених шаблонів [11, 12].

У даній статті розглядається новий гібридний підхід до аналізу мережевого трафіку, що поєднує методи машинного навчання (МН) та імовірнісного моделювання на базі баєсівських мереж (БМ). Запропонований метод ґрунтується на багатоступеневій обробці спостережуваних мережевих ознак. Спершу за допомогою ансамблевої кластеризації виявляються приховані поведінкові представлення, що відображають структуру взаємодій та поведінкову неоднорідність мережевої активності. А потім на їхній основі будується баєсівська модель, що дозволяє робити імовірнісні висновки про належність трафіку до нормальної або аномальної категорії. Запропонований у статті метод інтегрує переваги навчальних поведінкових моделей та пояснюваність імовірнісних структур, забезпечуючи високу адаптивність до розмаїття кібернетичних загроз при збереженні інтерпретованості отримуваних результатів.

Постановка проблеми. Задачу виявлення кіберзагроз у мережевому трафіку формулюють як проблему виявлення аномальної поведінки, що характеризує потенційно небезпечні або шкідливі дії в потоці мережевих з'єднань. Основна складність полягає в тому, що поведінка тих, хто атакує, може відрізнятися в кожному окремому випадку. Відповідно, такі дані не будуть заздалегідь представлені в навчальних вибірках (на чому базуються багато дослідників, наприклад, у роботах [2-10]). Крім того, аномалії часто мають поведінковий характер і проявляються не в значеннях окремих ознак, а у відхиленнях від типових шаблонів активності. Існуючі методи [8, 9, 11] або використовують статичні правила та сигнатури, не здатні впоратися з невідомими атаками, або застосовують методи машинного навчання (МН), які, хоча й здатні виявляти складні залежності, часто страждають від нестачі інтерпретованості та нездатності враховувати причинно-наслідкові зв'язки. В умовах, коли потрібно одночасно забезпечувати точність виявлення, стійкість до хибних спрацьовувань та можливість пояснення результатів, виникає необхідність у комплексних моделях, що поєднують емпіричну адаптацію з імовірнісною інтерпретацією..

Поставлене завдання полягає в розробці методу, здатного виявляти кіберзагрози на підставі поведінкового аналізу мережевого трафіку, спираючись на приховані закономірності, що виявляються машинним навчанням (МН), та баєсову мережу (БМ), яка вміє моделювати імовірнісні залежності між характеристиками поведінки та аномальністю.

Методи та моделі. Етап 1. Виявлення поведінкових ознак за допомогою ансамблевої кластеризації.

Нехай наявна вибірка мережевого трафіку:

$$\chi = \{x^{(1)}, x^{(2)}, \dots, x^{(n)}\}, x^{(i)} \in \mathbb{R}^d, \quad (1)$$

де $x^{(i)} = \left(x_1^{(i)}, x_2^{(i)}, \dots, x_d^{(i)}\right)$ – вектор ознак для і-го мережевого з'єднання (тривалість з'єднання, байти, протокол тощо).

Далі обираємо набір M кластеризаційних алгоритмів:

$$C = \{\mathrm{C}_1, \mathrm{C}_2, \dots, \mathrm{C}_M\}.$$

Тут кожен C_j – це функція $\mathrm{C}_j: \mathbb{R}^d \longrightarrow \{1,2, \dots, K_j\}$, яка кожному вектору $x^{(i)}$ зіставляє мітку кластера $z_j^{(i)} = C_j\left(x^{(i)}\right)$.

Тоді отримаємо такий проміжний результат.

$$Z^{(i)} = \left(z_1^{(i)}, z_2^{(i)}, \dots, z_M^{(i)}\right) \in \prod_{j=1}^{M}\{1, \dots, K_j\}. \quad (2)$$

На даному етапі мета полягає в тому, щоб виявити приховані поведінкові закономірності в мережевому трафіку шляхом застосування до даних кількох алгоритмів кластеризації. Це дозволяє сформувати узагальнене представлення про можливі шаблони трафіку. Наприклад, припустимо, що кожен об'єкт у вибірці — це окреме мережеве з'єднання, описане вектором

DOI:10.31548/itees.2025.01.073
ISSN 3083-7502

ознак. Як ми згадували раніше, наприклад, тривалість з'єднання, кількість переданих байт, тип протоколу тощо. Ці дані надходять на вхід кількох кластеризаційних алгоритмів, що становлять ансамбль. В якості алгоритмів можуть бути обрані такі добре відомі методи, як K-means, DBSCAN, ієрархічна кластеризація, спектральна кластеризація та інші. Зауважимо, що вибір конкретних алгоритмів для ансамблю визначається з урахуванням кількох факторів. Серед цих факторів розглядаються відмінності в методах апроксимації щільності та відстаней (щільнісні методи або метричні), чутливість до форми та розміру кластерів, стійкість до шуму та викидів, масштабованість при роботі з великими обсягами трафіку. Отже, застосовуючи кожен з алгоритмів до вихідних даних, ми отримаємо, що кожне мережеве з'єднання виявляється віднесеним до якого-небудь кластера. Таким чином, кожному з'єднанню зіставляється мітка, тобто ідентифікатор кластера, до якого воно віднесене алгоритмом.

Оскільки використовується кілька алгоритмів, для кожного з'єднання формується вектор міток, де кожна компонента відповідає результату кластеризації за одним із методів. У такому випадку, проміжний результат цього етапу являє собою так зване "кластерне представлення" поведінки з'єднання. Це набір кластерних міток, отриманих за результатами роботи всіх методів ансамблю. Дана множина міток не слід інтерпретувати безпосередньо як ознаки в класичному розумінні. Замість цього вона слугує основою для подальшої побудови числових поведінкових ознак, що відображають узгодженість або розбіжність в оцінках різних кластеризаторів. Зазначені ознаки на наступних етапах методу використовуватимуться як вхідні змінні в імовірнісній моделі баєсової мережі, дозволяючи їй враховувати та інтерпретувати поведінкові закономірності, виявлені кластеризацією.

Для переходу до числового вектора поведінкових ознак використовується відображення:

$$\Phi: \prod_{j=1}^{M}\{1, \dots, K_j\} \longrightarrow \mathbb{R}^k, \tag{3}$$

де Ф – відображення (функція), що перетворює вихідні кластерні мітки у вектор поведінкових ознак. Тобто. Ф виконує роль ембедингу, беручи результати кластеризації (мітки кластерів від кожного алгоритму) та перетворюючи їх на числовий вектор фіксованої довжини.;

K_j – результат роботи j-го кластеризатора (із ансамбля з M методів) над об'єктом x_i. Або $K_j(x_j)$ – кластерна мітка, присвоєна j-м методом кластеризації для i-го мережевого з'єднання;

$\mathbb{R}^k$ – вектор із k дійсних чисел. Вектор кластерного ембедингу для i-го мережевого з'єднання є вектором ознак розмірності d, придатним для подачі на вхід БМ. Розмірність d залежить від кількості методів в ансамблі (M), та кількості кластерів, що видаються кожним методом.

Тоді, після того як для кожного мережевого з'єднання було сформовано вектор кластерних міток, отриманих від кількох алгоритмів кластеризації, виникає необхідність перетворити даний вектор у числове представлення. Таке представлення має бути придатним для подальшого аналізу в рамках баєсової моделі. Це перетворення називається кластерним ембедингом [13, 14].

Кластерні мітки, отримані на попередньому етапі, є категоріальними значеннями (наприклад, "кластер 1", "кластер 3" тощо), які не несуть кількісного змісту і не можуть бути безпосередньо використані в баєсовій мережі (БМ), де змінні вимагають або числового, або суворо імовірнісного представлення. Крім того, принципово отримати стійкі до шуму ознаки. Тоді ці ознаки відображатимуть структуру даних, виявлену на підставі кількох методів кластеризації, а не результатів окремого методу. Для цього виконується відображення вектора міток у новий простір ознак, або простір поведінкових ознак, де кожна компонента відображає участь об'єкта в тих чи інших кластерах.

Дане перетворення може бути реалізоване різними методами. Розглянемо конкретний приклад. Припустимо, для одного з'єднання отримано наступні мітки від трьох алгоритмів: K-means відніс з'єднання до кластера 2 з 4 можливих. DBSCAN визначив з'єднання як "шум" (не

відніс до жодного кластера). Ієрархічна кластеризація помістила з'єднання в кластер 3 з 5 можливих. Отже, отримуємо вектор міток:

[Kmeans=2, DBSCAN=Noise, Hierarchical=3].

Щоб перетворити цей набір на числовий вектор (ембединг), можна застосувати, наприклад, one-hot кодування. Тоді K-means → [0, 1, 0, 0] (активний кластер 2 з 4). DBSCAN → [0, 0] (припустимо, два допустимих кластери, а "шум" кодується нулями). Та Hierarchical → [0, 0, 1, 0, 0] (активний кластер 3 з 5).

Об'єднуючи все, отримуємо ембединг довжиною 11 [0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0]. Цей вектор вже можна використовувати в наступних імовірнісних моделях, оскільки він чисельне інтерпретований та відображає характеристики об'єкта. Крім того, цей вектор агрегує інформацію одразу від кількох кластеризаторів і не залежить від вихідних чисельних ознак.

Головна мета – надати БМ «чисті» та високорівневі ознаки, що відображають структуру поведінки, а не лише технічні параметри трафіку. Відповідно, БМ працює не з необробленими даними, а зі стійкими та інтерпретованими характеристиками поведінки. Це суттєво для задачі виявлення аномалій у мережі, де сама «аномальність» найчастіше проявляється як поведінкове відхилення від норми.

У підсумку ми отримаємо поведінковий профіль об'єкта $x^{(i)}$, отриманий на основі кластерної належності.

Тоді на наступному етапі у нас йде побудова баєсової мережі поверх витягнутих ознак. Баєсова мережа – це імовірнісна модель, що описує залежність між випадковими величинами [15, 16].

Етап 2. Формалізуємо склад вузлів БМ.

Нехай:

$\boldsymbol{Z} = (Z_1, Z_2, \dots, Z_k)$ – змінні, що відповідають компонентам $h^{(i)}$;

$A \in \{0,1\}$ – бінарна випадкова величина, що відображає аномальність (*A=1* – аномалія, *A=0* – норма).

Тут $h^{(i)}$ – прихована (латентна) змінна, що характеризує аномальність *i*-го об'єкта. Наприклад, мережевого з'єднання. Це не спостережувана напряму характеристика, а прихована гіпотеза про те, чи належить об'єкт до класу «нормальний трафік» або «аномалія». Фактично це цільовий вузол у БМ, що моделює гіпотезу про те, чи є об'єкт потенційною кіберзагрозою.

Об'єднуємо всі змінні у множину $\boldsymbol{v} = (Z_1, \dots, Z_k, \boldsymbol{A})$. Множина $\boldsymbol{v}$ представляє собою повний набір змінних, що використовуються в БМ. Ми об'єднуємо їх, щоб визначити структуру БМ, або, іншими словами, щоб встановити, між якими змінними можуть існувати імовірнісні залежності. Також $\boldsymbol{v}$ необхідна для того, щоб задати область факторизації, оскільки БМ будується як факторизація сумісного розподілу всіх змінних зі $\boldsymbol{v}$, з використанням спрямованого ациклічного графа (DAG) [8]. Або у формалізованому вигляді

$$G = (V, E), V = \boldsymbol{v}, E \subseteq V \times V.$$

Кожне ребро $(Z_i, A) \in E$ інтерпретується як «ознака поведінки Z_i яка впливає на ймовірність того, що з'єднання аномалія».

Структура мережі, представлена у вигляді орієнтованого ациклічного графа, слугує «каркасом» для моделювання імовірнісних відношень між змінними. Визначивши, які вузли (змінні, такі як прихована змінна, що характеризує аномальність, та витягнуті поведінкові ознаки) безпосередньо пов'язані, ми сформуємо базу для факторизації сумісного розподілу ймовірностей. Іншими словами, це означає, що кожна змінна в мережі розглядається разом з набором своїх безпосередніх попередників, тим самим дозволяючи адекватно описати її імовірнісну поведінку. Перехід до сумісного розподілу здійснюється за допомогою розбиття повної імовірнісної міри на добуток умовних розподілів, де кожна компонента відповідає

DOI:10.31548/itees.2025.01.073
ISSN 3083-7502

вузлу графа і залежить тільки від змінних, з якими він безпосередньо пов'язаний у структурі (іншими словами, від його батьків у графі).

Сумісний розподіл у БМ задається так:

$$P(Z_1, \dots, Z_k, \boldsymbol{A}) = \prod_{v \in V} P(v|pa(v)), \quad (4)$$

де $pa(v) \subseteq V$ – множина батьків вузла v.

Основна мета – обчислити апостеріорний розподіл:

$$P(\boldsymbol{A} = 1|Z_1 = z_1, \dots, Z_k = z_k), \quad (5)$$

що і є передбаченням імовірності аномалії для об'єкта $x^{(i)}$.

Сумісний розподіл виражає повне множинне залежностей, виділених у ході побудови графа, і дозволяє потім проводити розрахунки умовних імовірностей, необхідні для виявлення кіберзагроз. Отже, об'єднання концепцій кроків методу, що включають отримання структури БМ та сумісного розподілу, відображає фундаментальну ідею баєсівського підходу, при якій попередньо визначена структура залежностей задає правила, за якими можна розкласти та описати складний розподіл імовірностей, що лежить в основі моделі виявлення аномалій у мережевому трафіку.

Після формалізації сумісного розподілу всіх змінних, включених у структуру БМ, наступним логічним етапом є процедура навчання моделі. Навчання БМ — це процес визначення числових параметрів, що відповідають умовним імовірностям у вузлах графа. Для кожної змінної, включеної в мережу, потрібно оцінити її умовний розподіл, що задається її батьками в графі залежностей.

Якщо структура мережі заздалегідь фіксована (наприклад, спираючись на апріорні знання експертів), завдання навчання зводиться до оцінювання параметрів розподілів. У випадку дискретних змінних, до яких належать як латентна змінна, що відображає аномальність, так і ознаки, отримані з кластерного ембедингу, навчання здійснюється шляхом підрахунку відносних частот за навчальною вибіркою. Тоді забезпечується максимізація правдоподібності, тобто параметрична настройка моделі таким чином, щоб вона найкращим чином пояснювала спостережувані дані про мережевий трафік.

Власне, навчання БМ є центральним етапом нашого методу. На цьому етапі імовірнісна модель набуває конкретного числового змісту, що відображає статистичний взаємозв'язок між поведінковими ознаками трафіку та гіпотезою про аномальність. А отримана навчена модель згодом послужить основою для імовірнісного висновку, дозволяючи оцінювати ступінь належності нових мережевих з'єднань до потенційно небезпечних на підставі їхнього поведінкового профілю.

Результати дослідження. Запропонований метод аналізу мережевого трафіку для виявлення кіберзагроз можна концептуалізувати у вигляді виразу (6).

Іншими словами, метод можна представити як композицію послідовно застосовуваних відображень, див. вираз (6). Причому кожне з яких реалізує певну функціональну трансформацію над даними, наближаючи нас до формування імовірнісної моделі поведінки.

$$x^{(i)} \xrightarrow{C} Z^{(i)} \xrightarrow{\Phi} h^{(i)} \xrightarrow{\text{BN inference}} P\left(A = 1 \middle| h^{(i)}\right). \quad (6)$$

На вхід подаються спостережувані характеристики мережевих з'єднань, що представляють собою низько рівневі мережеві ознаки $x^{(i)}$. Ці дані проходять через блок машинного навчання, зокрема – ансамблеву кластеризацію, результатом якої стає перетворення вихідних спостережень у поведінкове представлення. Цей етап можна інтерпретувати як перше відображення. Це перше відображення виявляє приховані поведінкові закономірності, характерні для різних типів активності в мережі, та кодує їх у вигляді ембедингів.

Наступним етапом є друге відображення – побудова баєсової мережі на отриманих ембедингах, тобто

$$h^{(i)} \xrightarrow{\text{BN inference}} P\left(A = 1 \middle| h^{(i)}\right).$$

Це відображення встановлює імовірнісні залежності між поведінковими ознаками та гіпотезою про аномальність з'єднання, дозволяючи об'єднати поведінку, зафіксовану кластеризаторами, з імовірнісною моделлю, пристосованою враховувати невизначеність, причинно-наслідкові зв'язки та частково спостережувані дані.

Підсумкова модель, таким чином, являє собою композицію перетворень, що послідовно переходить від вихідних мережевих ознак до імовірнісної оцінки ризику $a^{(i)}$ – це бінарна (або категоріальна зміна), що вказує, до якого класу насправді належить мережеве з'єднання. А параметр $\hat{A}^{(i)}$ – прогноз моделі, тобто результат імовірнісного висновку в БМ.

Зазвичай $\hat{A}^{(i)}$ означає

$$\hat{A}^{(i)} = \begin{cases} 1, \text{якщо } P\left(h^{(i)} = 1 \middle| z^{(i)}\right) > \tau, \\ 0, \text{ в інших випадках,} \end{cases} \quad (8)$$

де $\tau -$ заздалегідь вибраний поріг.

Наприклад, $a^{(i)}$ істинна мітка з датасету, яка використовується для оцінки якості. Тоді $\hat{A}^{(i)} -$ передбачення моделі, тобто результат баєсівського висновку про аномальність трафіку.

Запропонований підхід є розвитком існуючих методів виявлення аномалій у трафіку, він об'єднує ідеї поведінкового аналізу та імовірнісного висновку. Ключова новизна запропонованого рішення полягає у використанні ансамблевої кластеризації як механізму вилучення прихованих ознак поведінки, які потім використовуються не просто для класифікації, а для побудови динамічно адаптованої Баєсової мережі (БМ). Така мережа не лише виявляє загрози, але й здатна враховувати мінливість мережевої активності. Наприклад, це є актуальним в умовах кіберзагроз, що постійно еволюціонують. Вважаємо, що викладений у статті метод розширює класичну схему «кластеризація → маркування» і пропонує гнучку гібридну модель. У підсумку модель поєднує емпіричну поведінку та імовірнісну інтерпретацію, що надає викладеному методу добру.

Висновки. Запропонований у статті гібридний метод виявлення мережевих кіберзагроз поєднує в собі здібності поведінкового моделювання та імовірнісного висновку. Метод формує підхід до аналізу мережевого трафіку в умовах невизначеності вихідних параметрів для аналізу трафіку. Використання в методі на першому етапі ансамблевої кластеризації дозволить витягувати стійкі та інформативні представлення поведінкових ознак. А побудова на другому етапі Баєсової мережі на їх основі, відповідно, забезпечить здійсненність інтерпретованого висновку та врахування причинних залежностей між параметрами трафіку та загрозами для безпеки мережі. Представлений у статті підхід дозволяє гнучко адаптувати структуру моделі під зміну поведінки користувачів та динаміку атакуючих стратегій. Зауважимо, що викладена концептуальна схема формалізованого алгоритму для гібридного методу виявлення кіберзагроз у мережевому трафіку відкриває ряд напрямків для подальших досліджень. На наш погляд, перспективним є розвиток методів автоматичного навчання структури БМ на підставі аналізу динаміки трафіку. А також впровадження в модель навчання додаткових джерел інформації, наприклад, таких як часові та/або контекстні залежності.

Список використаних джерел

1. Lakhno, V., Yerbolat, K., Bagdat, Y., Kryvoruchko, O., Desiatko, A., Tsiutsiura, S., & Tsiutsiura, M. (2022). Модель захисту локальної мережі навчального закладу серверної системи віртуалізації. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка», 2(18), 6-23. https://doi.org/10.28925/2663-4023.2022.18.623.

2. Korpan, Y. V. (2015). Класифікація загроз інформаційній безпеці в комп'ютерних системах при віддаленій обробці даних. Реєстрація, зберігання і обробка даних, 17(2), 39-46.
3. Ilyenko, A., Ilyenko, S., Diana, K., & Mazur, Y. (2023). Практичні підходи щодо виявлення вразливостей в інформаційно-телекомунікаційних мережах. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка», 3(19), 96-108. https://doi.org/10.28925/2663-4023.2023.19.96108.
4. Makarenko, O., & Yanko, A. (2022). Концепція системи виявлення та запобігання вторгнень до мережі. Системи управління, навігації та зв'язку. Збірник наукових праць, 2(68), 59-67.
5. Трокоз, Є.М., Покотило, О.А., & Щур, Н.О. (2024). Моделювання загроз канального рівня в OWASP Threat Dragon з розробкою стратегії захисту. Технічна інженерія, (1 (93)), 246-254. https://doi.org/10.26642/ten-2024-1(93)-246-254.
6. Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. Journal of Network and Computer Applications, 60, 19–31. https://doi.org/10.1016/j.jnca.2015.11.016.
7. Jeffrey, N., Tan, Q., & Villar, J. R. (2023). A review of anomaly detection strategies to detect threats to cyber-physical systems. Electronics, 12(15), Article 3283. https://doi.org/10.3390/electronics12153283.
8. Aslan, Ö., Aktuğ, S. S., Ozkan-Okay, M., Yilmaz, A. A., & Akin, E. (2023). A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions. Electronics, 12(6), Article 1333. https://doi.org/10.3390/electronics12061333.
9. Samrin, R., & Vasumathi, D. (2017, December). Review on anomaly based network intrusion detection system. In 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT) (pp. 141–147). IEEE. https://doi.org/10.1109/ICEECCOT.2017.8284615.
10. Yang, Z., Liu, X., Li, T., Wu, D., Wang, J., Zhao, Y., & Han, H. (2022). A systematic literature review of methods and datasets for anomaly-based network intrusion detection. Computers & Security, 116, 102675. https://doi.org/10.1016/j.cose.2022.102675.
11. Alshamrani, A., Myneni, S., Chowdhary, A., & Huang, D. (2019). A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities. IEEE Communications Surveys & Tutorials, 21(2), 1851–1877. https://doi.org/10.1109/COMST.2019.2891891.
12. Bereziński, P., Jasiul, B., & Szpyrka, M. (2015). An entropy-based network anomaly detection method. Entropy, 17(4), 2367–2408. https://doi.org/10.3390/e17042367.
13. Xie, J., Girshick, R., & Farhadi, A. (2016, June). Unsupervised deep embedding for clustering analysis. In Proceedings of the 33rd International Conference on Machine Learning (pp. 478–487). PMLR. https://proceedings.mlr.press/v48/xieb16.html. https://doi.org/10.48550/arXiv.1511.06335.
14. Jiang, Z., Zheng, Y., Tan, H., Tang, B., & Zhou, H. (2016). Variational deep embedding: An unsupervised and generative approach to clustering. arXiv. https://arxiv.org/abs/1611.05148. https://doi.org/10.48550/arXiv.1611.05148.
15. Лєнков, С.В., Джулій, В.М., Берназ, Н.М., & Божук, С.О. (2017). Аналіз існуючих методів та алгоритмів виявлення атак в бездротових мережах передачі даних. Збірник наукових праць Військового інституту Київського національного університету імені Тараса Шевченка, (56), 124-132.
16. Голубенко, О.І., Лемешко, А.В., Цвик, О.С., & Мішкур, Ю.В. (2023). Забезпечення інформаційної безпеки в локальних мережах за допомогою контролю трафіку. ITSynergy, (2), 44-51. https://doi.org/10.53920/ITS-2023-2-3.

Lakhno Valeriy
Doctor of Technical Sciences, Professor, Professor of the Department of Computer systems, networks and cybersecurity,

National University of Life and Environmental Sciences of Ukraine,
ORCID: https://orcid.org/0000-0001-9695-4543
E-mail: lva964@nubip.edu.ua

Mamchenko Sergii
Doctor of Educational Sciences, Professor, Professor of the Department of Computer Systems, Networks and Cybersecurity,
National University of Life and Environmental Sciences of Ukraine
ORCID: https://orcid.org/0009-0006-8743-5606
E-mail: s.mamchenko@nubip.edu.ua

Matiievskyi Volodymyr
Senior lecturer, Department of Computer Systems, Networks and Cybersecurity,
National University of Life and Environmental Sciences of Ukraine
ORCID: https://orcid.org/0000-0002-1954-8493
E-mail: m_vv@outlook.com

ASPECTS OF DETECTING CYBER THREATS IN UNIVERSITY NETWORK TRAFFIC

Abstract. *Modern cyber threats to telecommunications systems and networks are characterized by a high degree of concealment, adaptability, and diversity. This complicates their rapid detection in network traffic, particularly at universities. Given the changing nature of cyberattacks, traditional methods based on signature analysis and fixed rules are proving insufficiently effective for identifying new or modified threats. In this regard, the development of intelligent hybrid approaches is becoming increasingly important. Such methods are capable of analyzing the behavioral characteristics of university traffic and adapting to its changes. The article presents a method for detecting cyber threats based on a combination of ensemble clustering and Bayesian probabilistic modeling methods. At the first stage, machine learning is used to identify hidden behavioral features of network connections in the university network based on various clustering algorithms. The resulting behavior embeddings are then used as input data for constructing a Bayesian network that describes the probabilistic dependencies between behavior parameters and anomaly features. The proposed approach not only allows detecting deviations from normal traffic behavior, but also ensures the interpretability of decisions in the field of information security. The practical value of the method lies in its potential for use in network traffic monitoring systems in corporate networks.*

Keywords*: network traffic, network, university, behavioral analysis, Bayesian network, clustering, machine learning, method, cybersecurity.*