ISSN 2219-9365

https://doi.org/10.31891/2219-9365-2026-85-45
УДК 004.52:004.75

ДРОЗД Андрій
Хмельницький національний університет
https://orcid.org/0009-0008-1049-1911
e-mail: andriydrozdit@gmail.com
МИКУЛЯК Дмитро
Хмельницький національний університет
e-mail: mykuliak.dmytro.ua@gmail.com
https://orcid.org/0009-0000-9437-8685

НТЕЛЕКТУАЛЬНА КОМП'ЮТЕРНА СИСТЕМА АВТОМАТИЧНОГО ВИЯВЛЕННЯ ВРАЗЛИВОСТЕЙ ВЕБ-ЗАСТОСУНКІВ ТА КЛАСИФІКАЦІЇ ЗАГРОЗ

У цій статті досліджується проблема автоматизованого виявлення та класифікації вразливостей веб-застосунків із використанням інтелектуальних комп'ютерних систем в умовах безперервного циклу розробки. Здійснено системний аналіз сучасних підходів до виявлення шкідливого програмного забезпечення та кібератак (SAST, DAST, SCA), зокрема архітектури систем децепції та методів машинного навчання, на основі фундаментальних досліджень вітчизняних та зарубіжних науковців.

Розглянуто актуальні загрози згідно з міжнародними стандартами OWASP Top 10:2021 та таксономією CWE. Особливу увагу приділено застосуванню великих мовних моделей (LLM) та архітектур на основі трансформерів (Transformers) для підвищення точності виявлення логічних вразливостей у вихідному коді, що є перспективним напрямком порівняно з традиційними статичними сканерами.

Авторами запропоновано концептуальну архітектуру інтелектуальної системи, яка базується на синергії графових нейронних мереж (GNN) та великих мовних моделей (LLM) для забезпечення семантичного аналізу та контекстної пріоритизації загроз із використанням розширених метрик CVSS. Обґрунтовано доцільність запровадження модуля нормалізації даних із гетерогенних сканерів у єдиний ознаковий простір. Експериментальні результати демонструють суттєве зниження рівня хибних спрацювань (False Positives) та підвищення метрики F1-score при використанні гібридної моделі. Робота становить практичний інтерес для фахівців з кібербезпеки, DevSecOps-інженерів та розробників засобів автоматизованого аудиту.

Ключові слова: веб-безпека, виявлення вразливостей, інтелектуальна система, машинне навчання, великі мовні моделі, LLM, трансформери, GNN, OWASP Top 10, CWE, DevSecOps.

DROZD Andriy, MYKULIAK Dmytro
Khmelnytskyi National University

INTELLIGENT COMPUTER SYSTEM FOR AUTOMATIC DETECTION OF WEB APPLICATION VULNERABILITIES AND THREAT CLASSIFICATION

The article investigates the problem of automated detection and classification of web application vulnerabilities using intelligent computer systems in a continuous development cycle. A systematic analysis of modern approaches to malware and cyberattack detection (SAST, DAST, SCA), including deception system architectures and machine learning methods, is carried out based on fundamental research by domestic and foreign scientists.

Current threats are reviewed according to the international standards OWASP Top 10:2021 and the CWE taxonomy. Special attention is paid to the application of Large Language Models (LLM) and Transformer-based architectures to improve the accuracy of detecting logical vulnerabilities in source code, representing a promising advancement over traditional static scanners.

The authors propose a conceptual architecture of an intelligent system based on the synergy of Graph Neural Networks (GNN) and LLMs to provide semantic analysis and context-aware threat prioritization using extended CVSS metrics. The feasibility of introducing a data normalization module from heterogeneous scanners into a unified feature space is substantiated. Experimental results demonstrate a significant reduction in the False Positive rate and an increase in the F1-score when using the hybrid model. The study is of practical interest to cybersecurity professionals, DevSecOps engineers, and developers of automated audit tools.

Keywords: web security, vulnerability detection, intelligent system, machine learning, Large Language Models, LLM, Transformers, GNN, OWASP Top 10, CWE, DevSecOps.

Стаття надійшла до редакції / Received 14.01.2026
Прийнята до друку / Accepted 13.02.2026
Опубліковано / Published 05.03.2026

ПОСТАНОВКА ПРОБЛЕМИ У ЗАГАЛЬНОМУ ВИГЛЯДІ ТА ЇЇ ЗВ'ЯЗОК ІЗ ВАЖЛИВИМИ НАУКОВИМИ ЧИ ПРАКТИЧНИМИ ЗАВДАННЯМИ

Сучасний етап розвитку інформаційних технологій невідривно пов'язаний із глобальною цифровізацією бізнес-процесів та перенесенням критичної інфраструктури в онлайн-середовище. Основою такої цифровізації є веб-застосунки та хмарні архітектури. Водночас цей етап характеризується стрімким ускладненням архітектури програмного забезпечення (ПЗ). Перехід від монолітних додатків до мікросервісних архітектур, інтеграція великої кількості сторонніх компонентів, використання контейнеризації та API-орієнтованих підходів суттєво розширили поверхню для можливих кібератак.

Як наслідок, безпека веб-застосунків стала одним із найкритичніших викликів для індустрії. Попри наявність розвиненої міжнародної нормативної бази (стандарти серії ISO/IEC 27001, ISO/IEC 27002 щодо впровадження систем управління інформаційною безпекою, та ISO/IEC 27005 у контексті управління ризиками) і глобальних таксономій вразливостей (CVE – Common Vulnerabilities and Exposures, CWE – Common Weakness Enumeration, CAPEC – Common Attack Pattern Enumeration and Classification), практична реалізація процесів виявлення, класифікації та пріоритизації загроз у реальних конвеєрах безперервної розробки та розгортання (CI/CD і DevSecOps) залишається надзвичайно фрагментованою.

Згідно зі звітами міжнародних аналітичних агенцій та бази даних OWASP (Open Worldwide Application Security Project), понад 80% сучасних веб-застосунків містять щонайменше одну критичну вразливість на момент релізу. Традиційні підходи до забезпечення безпеки, які базуються на періодичних ручних аудитах або точковому застосуванні ізольованих інструментів перевірки, виявляються неефективними в умовах концепції Agile, де нові версії продукту можуть випускатися кілька разів на день.

Методи автоматизованого аналізу, такі як SAST (Static Application Security Testing), DAST (Dynamic Application Security Testing) та SCA (Software Composition Analysis), стали стандартом де-факто для індустрії. Однак їх розрізнене використання не здатне забезпечити цілісного бачення стану безпеки продукту. Аналіз поточної ситуації у сфері автоматизованого тестування безпеки дозволяє виділити наступні критичні проблеми, що потребують невідкладного наукового та інженерного вирішення:

Гетерогенність результатів аналізу та відсутність уніфікації. Однією з головних проблем є відсутність уніфікованої схеми інтеграції даних, отриманих від різних інструментів. Системи SAST аналізують вихідний код, виявляючи структурні недоліки (наприклад, жорстко закодовані паролі чи відсутність перевірки вводу). Системи DAST взаємодіють із працюючим застосунком через HTTP-запити, імітуючи поведінку зловмисника, тоді як SCA-сканери перевіряють сторонні бібліотеки (Open Source) на наявність відомих CVE. Кожен із цих інструментів генерує звіти у власному форматі (JSON, XML, SARIF), використовує власні алгоритми визначення критичності та власну термінологію. Це ускладнює агрегацію, кореляцію та дедуплікацію даних, створюючи хаос у процесі усунення знайдених проблем.

Проблема високого рівня хибних спрацювань (False Positives). Статичні аналізатори (SAST), маючи доступ до всього коду, часто використовують жорсткі регулярні вирази або прості лексичні аналізатори. Як наслідок, вони генерують величезну кількість сповіщень про потенційні вразливості, які в реальному контексті виконання не становлять загрози (наприклад, змінна, що не пройшла очищення (sanitization) у одному модулі, може бути очищена на рівні Middleware до потрапляння в базу даних). DAST-аналізатори, навпаки, можуть давати хибнонегативні результати (False Negatives) через неможливість доступу до прихованих ендпоінтів або через складні механізми автентифікації. Постійний потік хибних сповіщень призводить до ефекту «втоми від попереджень» (alert fatigue), через що фахівці з безпеки починають ігнорувати повідомлення сканерів, пропускаючи справді критичні загрози.

Існуюча парадигма визначення критичності здебільшого спирається на базовий скоринг CVSS (Common Vulnerability Scoring System). Проте базовий CVSS-скоринг оцінює лише технічну складність експлуатації вразливості та її теоретичний вплив на конфіденційність, цілісність і доступність системи (тріада CIA). Ця метрика є абсолютно статичною і не враховує бізнес-контекст застосунку, архітектурні особливості середовища розгортання, наявність компенсуючих механізмів (наприклад, Web Application Firewall) та цінність скомпрометованих активів. У результаті, вразливість із високим базовим балом у внутрішньому тестовому середовищі може розглядатися інструментами як пріоритетніша для виправлення, ніж вразливість із середнім балом на публічному платіжному шлюзі.

Незважаючи на існування детальних таксономій, на практиці спостерігається семантичний розрив. Відсутній автоматизований, формалізований зв’язок у ланцюгу формування загрози: від абстрактного типу помилки розробника (класифікація CWE) через конкретний екземпляр уразливості (CVE) до потенційного методу або патерну атаки (CAPEC). Аналізатори вказують на рядок коду, але не дають розуміння сценарію експлуатації, що критично важливо для побудови систем захисту (наприклад, систем децепції чи пасток).

Хоча профільні стандарти ISO/IEC 29147 (Розкриття інформації про вразливості) та ISO/IEC 30111 (Процеси обробки вразливостей) чітко регламентують організаційні та управлінські аспекти, вони не визначають конкретних математичних чи алгоритмічних механізмів для інтелектуальної агрегації результатів аналізу. У традиційних системах цей процес досі виконується мануально, що є неприйнятним для масштабних корпоративних середовищ.

Таким чином, постає гостра науково-практична потреба у створенні формалізованої інтегрованої моделі, що забезпечить: синхронізацію таксономічних знань (CWE, CAPEC) для глибокого семантичного розуміння природи загроз; адаптивну інтеграцію результатів мультиінструментального сканування із приведенням гетерогенних даних до єдиного ознакового простору: динамічний ризик-скоринг, що поєднує базові метрики CVSS із контекстними коефіцієнтами експлуатації згідно з рекомендаціями стандарту управління ризиками ISO/IEC 27005.

Розв'язання цієї багатовимірної проблеми класичними детермінованими алгоритмами є неможливим через високу варіативність вихідного коду та мінливість ландшафту кіберзагроз. Єдиним ефективним шляхом

є розроблення інтелектуальної комп'ютерної системи, здатної до автоматичного вилучення ознак, абстрагування та машинного навчання на великих масивах даних про вразливості. Залучення методів глибинного навчання (Deep Learning), зокрема графових нейронних мереж (GNN) для структурного аналізу коду та великих мовних моделей (LLM) для контекстної обробки природної мови та синтаксису програмування, відкриває нові перспективи у створенні систем, здатних не лише виявляти, але й інтелектуально класифікувати та пріоритизувати кіберзагрози з точністю, що наближається до людської експертизи.

АНАЛІЗ ДОСЛІДЖЕНЬ ТА ПУБЛІКАЦІЙ

Питання автоматизації виявлення вразливостей, зниження відсотка хибних спрацювань та інтелектуалізації систем кіберзахисту перебуває в центрі уваги багатьох міжнародних та вітчизняних дослідницьких інституцій. Детальний аналіз сучасної фахової літератури (2020–2026 рр.) дозволяє класифікувати існуючі підходи за декількома ключовими векторами.

Методологічна база, стандартизація та онтологія загроз. Фундаментом для будь-якої системи автоматичного виявлення є стандарти та бази знань, що підтримуються міжнародними спільнотами, такими як MITRE Corporation та OWASP Foundation. У роботі [18] детально описується застосування таксономій CWE (Common Weakness Enumeration) та CAPEC, які дозволяють формалізувати типи помилок у коді та методи їх експлуатації зловмисниками. Проте, як справедливо зазначають автори методології OWASP [3] та дослідник [4], статичні списки (наприклад, щорічні звіти OWASP Top 10) хоча й дають загальне розуміння трендів, але потребують постійної адаптації до динамічного мікросервісного середовища сучасних веб-застосунків. Питання ризик-менеджменту та оцінювання критичності інцидентів традиційно базуються на серії стандартів ISO/IEC 2700x. Вони задають чудові концептуальні рамки для побудови СУІБ (Систем управління інформаційною безпекою), але, як зазначено у загальних положеннях [1], залишають суттєві прогалини в інженерній та алгоритмічній реалізації процесів пріоритизації на рівні програмного коду.

Інтелектуальні методи аналізу на основі глибинного навчання (Deep Learning). Сучасні дослідження (2023–2026 рр.) демонструють стрімкий зсув від класичних методів машинного навчання (таких як Random Forest чи SVM) у бік використання архітектур глибинного навчання. Зокрема, у роботі [7] та колектив авторів роботи [19] проводять масштабний аналіз застосування архітектур на основі трансформерів (Transformers, BERT) для виявлення шкідливого програмного забезпечення та автоматичного аудиту вихідного коду. Вони доводять, що здатність трансформерів утримувати контекст на великих ділянках коду дозволяє значно знизити рівень хибних спрацювань (FPR). Більше того, в роботі номер [8] та номер [15] наголошують на надзвичайній ефективності використання великих мовних моделей (Large Language Models, LLM) для семантичного аналізу вихідного коду. Автори стверджують, що LLM здатні виявляти складні логічні вразливості (Business Logic Vulnerabilities) та помилки порушення контролю доступу (Broken Access Control), які концептуально недоступні для класичних детермінованих SAST-інструментів, що базуються на абстрактних синтаксичних деревах (AST) чи регулярних виразах. Практичне застосування API LLM-моделей (наприклад, ChatGPT) для пошуку вразливостей також досліджено вітчизняними науковцями в роботі номер [5].

Гібридні архітектури, системи децепції та нормалізація даних. Окремий потужний напрям становлять роботи української наукової школи у сфері кібербезпеки, зокрема дослідження в роботах [2, 6, 17]. Ці фундаментальні праці зосереджені на створенні багаторівневих систем захисту з елементами кібердецепції (Cyber Deception) та пасток (Honeypots), що дозволяє виявляти цілеспрямовані атаки та ботнети в режимі реального часу. Особливої уваги в контексті нашого дослідження заслуговують розроблені ними критерії оцінювання варіантів централізації в архітектурі мультикомп'ютерних систем [6] та методи синтезу систем виявлення [2]. Математичний апарат нормалізації та агрегації результатів у таких гетерогенних розподілених системах закладає міцну теоретичну основу для розв'язання задачі уніфікації даних від різних засобів сканування (SAST/DAST/SCA), з якою стикаються сучасні DevSecOps-інженери.

Автоматизація та практична інтеграція у CI/CD. Низка прикладних досліджень присвячена архітектурним питанням впровадження систем безпеки у життєвий цикл розробки ПЗ (SDLC). Роботи [14], [16] та [9] фокусуються на розробленні автоматизованих сканерів та веб-краулерів, здатних працювати в режимі реального часу (Real-Time Threat Detection). Однак, попри досягнення в швидкості сканування, автори відзначають, що залишається невирішеною ключова проблема ефективної агрегації даних. Відсутність механізмів дедуплікації часто призводить до конфліктів у звітності та багаторазового дублювання однієї і тієї ж інформації, що нівелює переваги автоматизації.

Виділення невирішеної частини загальної проблеми Незважаючи на значні успіхи наукової спільноти в окремих галузях (вдосконалення трансформерів, розвиток систем децепції, розробка нових метрик), аналіз публікацій дозволяє стверджувати, що більшість існуючих систем фокусуються фрагментарно: або на покращенні процесу виявлення (Detection), або виключно на процесі категоризації (Classification), не забезпечуючи замкненого циклу інтелектуальної обробки інцидентів безпеки.

На сьогоднішній день відсутні комплексні формалізовані моделі, які б одночасно:

Нормалізували гетерогенні дані від SAST, DAST та SCA до єдиного ознакового простору.
Використовували синергію графових нейронних мереж (GNN) для математично точного аналізу структури потоків даних застосунку та великих мовних моделей (LLM) для глибокого контекстуального та семантичного аналізу описів уразливостей.

Забезпечували на основі цього динамічний контекстно-орієнтований ризик-скоринг (Risk-Scoring), що доповнює стандартні оцінки CVSS.

Розроблення такої інтелектуальної комп'ютерної системи становить актуальну науково-прикладну задачу, якій і присвячено дане дослідження.

ФОРМУЛЮВАННЯ ЦІЛЕЙ СТАТТІ

Інтелектуальна система автоматичного виявлення вразливостей веб-застосунків (ІКС-АВВЗ) ґрунтується на концепції багаторівневої обробки даних. Вона охоплює збір гетерогенних результатів сканування, їхню нормалізацію, гібридну нейромережеву класифікацію та контекстуальний ризик-скоринг. Для формального обґрунтування запропоноване рішення описано математичною моделлю.

Формалізація предметної області та цільова функція

Розглянемо множину веб-застосунків $W = \{w_1, w_2, \dots, w_N\}$. Кожен застосунок $w_i \in W$складається з множини структурних або логічних артефактів:

$$A(w_i) = \{a_{i1}, a_{i2}, \dots, a_{iK}\}, \tag{1}$$

де a_{ik}може бути файлом вихідного коду, функцією, модулем, графом залежностей або HTTP-запитом.

Множина відомих класів загроз позначається $C = \{c_1, c_2, \dots, c_M\}$, де кожен c_jвідповідає категорії з таксономій CWE (наприклад, CWE-79 для XSS або CWE-89 для SQL-ін'єкцій).

Кожному артефакту a_{ik}ставиться у відповідність вектор ознак $x \in \mathbb{R}^d$через оператор Φ:

$$x = \Phi(a_{ik}), x \in \mathrm{R}^d, \tag{2}$$

де $\Phi(\cdot)$формує ембеддинги на основі AST, графів залежностей або поведінкових характеристик.

Класифікація виконується відображенням

$$f_\theta \colon \mathbb{R}^d \to \Delta^{M-1}, \tag{3}$$

де θ- параметри моделі, а Δ^{M-1}- симплекс, що задає розподіл імовірностей належності артефакту до кожного класу.

Навчання моделі зводиться до мінімізації очікуваної втрати:

$$L(f_\theta) = \mathbb{E}_{(x,y)\sim P}[l(f_\theta(x), y)] \to \min_\theta, \tag{4}$$

де P- невідомий розподіл даних і міток, а $l(\cdot,\cdot)$- функція втрат. На практиці мінімізується емпіричний ризик:

$$\hat{L}_n(f_\theta) = \frac{1}{n}\sum_{i=1}^{n} l(f_\theta(x_i), y_i), \tag{5}$$

як функцію втрат використовують крос-ентропію:

$$l(f_\theta(x), y) = -\sum_{j=1}^{M} y^{(j)} \log(f_\theta(x)^{(j)}), \tag{6}$$

де $y^{(j)}$- індикатор належності до класу j.

Інтеграція SAST/DAST/SCA та нормалізація ознак

Основним недоліком традиційних конвеєрів безпеки є ізольованість різних аналізаторів. Позначимо множини сирих сповіщень від статичного, динамічного та компоновочного аналізу як $S_{\text{SAST}}, S_{\text{DAST}}, S_{\text{SCA}}$. Об'єднаний простір інцидентів:

$$S_{\text{total}} = S_{\text{SAST}} \cup S_{\text{DAST}} \cup S_{\text{SCA}}. \tag{6}$$

Для перетворення різнорідних форматів (JSON, XML, SARIF) у спільний векторний простір вводиться оператор нормалізації:

$$N \colon S_{\text{total}} \to \mathrm{R}^d, \tag{7}$$

який проводить дедуплікацію та об'єднує дані з різних джерел. Наприклад, якщо статичний аналізатор знаходить підозрілий рядок, а динамічний підтверджує його експлуатацію, оператор Nзливає ці сповіщення в один вектор для подальшої обробки.

Гібридна модель (GNN + LLM)

Для зменшення хибних спрацьовувань запропоновано ансамблеву модель, що поєднує:

Графовий компонент (GNN). Код моделюється як орієнтований граф $G = (V, E)$. Вузли Vвідповідають операторам, змінним та API-ендпоінтам, а ребра E- залежностям потоків даних і керування. Оновлення embedding-вузлів на k-му шарі:

$$h_v^{(k)} = \sigma(W^{(k)} \cdot \text{AGGREGATE}\{h_u^{(k-1)} : u \in \mathcal{N}(v)\}), \tag{8}$$

де $W^{(k)}$- матриця ваг, σ- нелінійність, $\mathcal{N}(v)$- сусіди.

Мовний компонент (LLM). У той же час вектор xподається на донавчену модель Transformer, яка аналізує текстову семантику коду, коментарів та HTTP-відповідей, даючи розподіл $P_{\text{LLM}}(y \mid x)$.

Фінальний розподіл комбінується:

$$P_{\text{final}}(y \mid x) = \alpha \cdot P_{\text{GNN}}(y \mid x) + (1 - \alpha)\, P_{\text{LLM}}(y \mid x), \tag{9}$$

де $\alpha \in [0,1]$підбирається на валідаційній вибірці.

Межі узагальнюючої похибки оцінюють через нерівність Хефдінга: для випадкової вибірки розміром nз імовірністю не менше $1 - \delta$:

$$| L(f_\theta) - \hat{L}_n(f_\theta) | \leq \sqrt{\frac{\ln(2/\delta)}{2n}}. \tag{10}$$

Отже, за достатньо великого nгібридна модель узгоджена й не перенавчається. Для гладких функцій втрат із константою Lградієнтний спуск збігається до локального мінімуму за умов кроку $\eta \leq 1/L$.

Контекстуальний ризик-скоринг

Запропоновано динамічний індекс ризику:

$$\text{Risk}_{\text{total}} = \text{BaseScore}_{\text{CVSS}} \cdot K_{\text{asset}} \cdot K_{\text{exposure}}, \tag{11}$$

де:

$\text{BaseScore}_{\text{CVSS}}$- базова оцінка з CVSS (0–10)

K_{asset}- коефіцієнт бізнес-важливості активу (максимальний для критичних модулів)

K_{exposure}- коефіцієнт експозиції (близький до 1 для публічних API, менший для внутрішніх)

Цей підхід дозволяє не лише знаходити вразливості, а й генерувати пріоритетний план усунення, орієнтований на бізнес-критичні загрози. Ми сформували формальну математичну модель інтелектуальної системи автоматичного виявлення вразливостей вебзастосунків (ІКС АВВЗ), що базується на концепції багаторівневої обробки даних та інтеграції різнорідних джерел безпекової інформації. Запропонований підхід забезпечує перехід від фрагментарного аналізу до цілісної моделі прийняття рішень, яка поєднує структурний, поведінковий і семантичний контексти.

Формалізація предметної області через множину вебзастосунків, їхніх артефактів та простір класів загроз дозволила звести задачу виявлення вразливостей до задачі багатокласової класифікації у векторному просторі ознак. Введення операторів відображення Ф та N забезпечує узгоджене представлення гетерогенних даних (SAST, DAST, SCA) у спільному ознаковому просторі, що усуває проблему ізольованості традиційних аналізаторів та зменшує кількість дублікатів і суперечливих сповіщень.

Гібридна архітектура (GNN + LLM) дозволяє одночасно враховувати топологічну структуру програмного коду (граф залежностей потоків даних і керування) та його семантичний зміст. Ансамблеве поєднання ймовірнісних розподілів підвищує точність класифікації та знижує рівень хибнопозитивних і хибнонегативних спрацьовувань. Теоретичне обґрунтування через мінімізацію емпіричного ризику та оцінку меж узагальнюючої похибки (нерівність Хефдінга) підтверджує статистичну узгодженість моделі за умови достатнього обсягу навчальної вибірки.

Запропонований механізм контекстуального ризик-скорингу розширює класичний підхід CVSS за рахунок врахування бізнес-критичності активів і рівня їх експозиції. Це забезпечує перехід від суто технічної оцінки вразливості до управлінсько-орієнтованої моделі пріоритезації, що підвищує ефективність планування заходів з усунення загроз.

Таким чином, розроблена математична модель ІКС АВВЗ забезпечує: формальну узгодженість задачі виявлення вразливостей; інтеграцію різнорідних безпекових джерел у єдиний аналітичний простір; підвищення точності класифікації за рахунок гібридної нейромережевої архітектури; адаптивну бізнес-орієнтовану пріоритезацію ризиків.

Отримані результати створюють теоретичну основу для подальшої реалізації системи в межах архітектури та експериментальної валідації її ефективності на реальних наборах даних.

ЕКСПЕРИМЕНТ

Мета експериментальної частини - емпірично перевірити ефективність інтелектуальної комп'ютерної системи автоматичного виявлення вразливостей веб-застосунків (ІКС-АВВЗ) та довести статистично значущу перевагу гібридної моделі (GNN + LLM) над базовими підходами. Порівняльне тестування проводилось одночасно для чотирьох архітектур:

1.Базова модель - Random Forest (RF), класичний ансамблевий метод, типовий для традиційних сканерів.

2.Структурна модель - графова нейронна мережа (GNN), навчена лише на графах потоків даних і керування (DFG/CFG).

3.Семантична модель - велика мовна модель (LLM) на базі Transformer, донавчена на текстових фрагментах коду та звітах про вразливості.

4.Гібридна архітектура - поєднання GNN та LLM із зваженим об'єднанням передбачень.

Ефективність оцінювали за точністю, повнотою, F1-оцінкою, площею під ROC-кривою (ROC-AUC), часткою хибних спрацювань (FPR) та обчислювальною складністю інференсу.

Формування датасету

Для валідації було сформовано комбінований датасет обсягом $N = 14{,}042$події безпеки з двох груп:

1.Реальні дані (62,3 %) - 8 742 підтверджені вразливості з 12 open-source веб-проєктів (Python, Java, Node.js), розмічені за таксономією CWE та базами NVD, OWASP Benchmark і ExploitDB.

2.Синтетичні дані (37,7 %) - 5 300 згенерованих фрагментів коду з мутаціями (SQL Injection, XSS, RCE), створених за допомогою фаззингу та синтаксичних мутацій.

3.Вибірку випадково поділено на тренувальну (70 %), валідаційну (15 %) та тестову (15 %) частини, щоб уникнути перенавчання і забезпечити коректну оцінку.

Метрики та статистичні гіпотези

Якість оцінювали за матрицею помилок, обчислюючи:

Точність: $P = \frac{TP}{TP+FP}$

Повнота: $R = \frac{TP}{TP+FN}$

F1-оцінка: $F_1 = 2\frac{P \cdot R}{P+R}$

Частка хибних спрацювань (FPR): $\frac{FP}{FP+TN}$

Інтегральним показником слугувала площа під ROC-кривою:

$$\text{AUC} = \int_0^1 TPR(FPR)\, d(FPR). \tag{12}$$

Формульовано гіпотези:

H_0: середня точність гібридної та найкращої ізольованої моделі однакові ($\mu_{\text{Hybrid}} = \mu_{\text{LLM}}$).

H_1: гібридна модель має вищу точність ($\mu_{\text{Hybrid}} > \mu_{\text{LLM}}$).

Перевірка виконувалась однобічним t-критерієм Стьюдента на рівні $\alpha = 0{,}05$.

4. Результати тестування

Таблиця 1

Порівняльний аналіз ефективності

Архітектура	Precision	Recall	F1-score	ROC-AUC
Random Forest	0.81	0.74	0.77	0.84
GNN	0.86	0.82	0.84	0.89
LLM	0.88	0.85	0.86	0.91
GNN + LLM	0.92	0.89	0.90	0.95

Таблиця 2

Аналіз хибних спрацювань

Архітектура	FPR	Відносне зниження
Random Forest	0,18 (18 %)	–
GNN	0,13 (13 %)	−27,7 %
LLM	0,11 (11 %)	−38,8 %
GNN + LLM	0,08 (8 %)	−55,5 %

Гібридна архітектура показала найкращі показники: F1-оцінка підвищилась на 13 % порівняно з Random Forest, ROC-AUC - на 11 %. Статистика $t = 4{,}12$, $p < 0{,}001$дозволяє відхилити H_0на користь H_1.

Зниження FPR до 8 % на 55,5 % менше базового рівня, що істотно зменшує “втому від попереджень”.

Оцінка узагальнюючої похибки та складність
Для $n = 14\,042$та довіри 95 %:

$$\epsilon \leq \sqrt{\frac{\ln(2/0{,}05)}{2 \cdot 14\,042}} \approx 0{,}011, \tag{13}$$

що відповідає максимальному відхиленню не більше 1,1 %. Складність інференсу GNN - $\mathcal{O}(|V| + |E|)$, LLM - $\mathcal{O}(L^2)$. Завдяки кешуванню і батчингу середній час аналізу одного комміту становив ~1,2 с.

Оцінювання економічної ефективності впровадження ІКС-АВВЗ

Наша мета - зменшити прямі й непрямі витрати підприємства (ліквідація інцидентів, ручний аудит, простій сервісів, штрафи). Оцінка проєкту базується на метриках TCO, ROI, NPV та строку окупності.

1. Модель витрат (TCO)

Сукупна вартість першого року:

$$\text{TCO} = C_{\text{dev}} + C_{\text{infra}} + C_{\text{support}} + C_{\text{staff}}, \tag{14}$$

де компоненти представляють розробку, інфраструктуру, підтримку та персонал. Для типової IT-компанії (2026 р.):

Таблиця 3

Оцінювання економічної ефективності

Стаття	Опис	Вартість (USD/рік)
C_{dev}	Адаптація GNN + LLM, інтеграція	45 000
C_{infra}	Хмарні потужності	18 000
C_{support}	Оновлення баз, моніторинг	12 000
C_{staff}	Робота фахівців	40 000
Всього (TCO)	-	115 000

2. Модель економічного ефекту й ROI

Економія:

$$\text{Savings} = (N_{\text{before}} - N_{\text{after}}) \times C_{\text{inc}}, \tag{15}$$

де $C_{\text{inc}} = 120{,}000$ USD, зниження інцидентів із 3 до 1 на рік дає 240 000 USD економії. Рентабельність:

$$\text{ROI} = \frac{\text{Savings} - \text{TCO}}{\text{TCO}} \times 100\% \approx 108{,}7\%. \tag{16}$$

3. Чиста приведена вартість (NPV) й строк окупності

Для періоду $T = 3$роки, ставки дисконту $r = 0{,}15$:

$$\text{NPV} = \sum_{t=1}^{T} \frac{CF_t}{(1+r)^t} - I_0 \approx 432{,}968 \text{ USD}, \tag{17}$$

$NPV > 0$- проєкт вигідний. Строк окупності:

$$\text{PP} = \frac{\text{TCO}}{\text{Savings}/12} \approx 5{,}75 \text{ місяців}.$$

4. Сценарний аналіз і порівняння з ручним аудитом

Таблиця 4

Сценарний аналіз і порівняння з ручним аудитом

Сценарій	Інциденти/рік	Економія (USD)	Очікуваний ROI
Оптимістичний	4 → 1	360 000	213 %
Базовий	3 → 1	240 000	108 %
Песимістичний	2 → 1	120 000	4 %

Навіть у песимістичному сценарії проєкт не збитковий. Для порівняння: ручний аудит коштує ~150 000 USD/рік, FPR 25-35 %, перевірка займає тижні, тоді як ІКС-АВВЗ зменшує витрати до 115 000 USD/рік, FPR - 8 %, час перевірки - до кількох годин.

Експериментально доведено статистично значущу перевагу гібридної архітектури GNN + LLM над класичними методами. Підвищення F1-score до 0,90, ROC-AUC до 0,95 й зниження FPR до 0,08 пояснюються синергією структурного аналізу (AST, CFG, PDG) та контекстуального розуміння логіки. Гібридна модель формально описується як ансамбль $\hat{y} = \alpha y_{\text{GNN}} + (1-\alpha) y_{\text{LLM}}$, що зменшує дисперсію оцінки й забезпечує узагальнюючу похибку < 1,1 %.

ВИСНОВКИ З ДАНОГО ДОСЛІДЖЕННЯ І ПЕРСПЕКТИВИ ПОДАЛЬШИХ РОЗВІДОК У ДАНОМУ НАПРЯМІ

У роботі вирішено практичну задачу підвищення ефективності автоматизованого виявлення вразливостей веб-застосунків та класифікації загроз шляхом створення інтегрованої інтелектуальної системи.

Системний аналіз методів і таксономій. Проаналізовано підходи SAST, DAST, SCA та таксономії CWE, CVE, CAPEC. Показано, що фрагментованість класичних інструментів і статичність метрик CVSS спричиняють перевантаження DevSecOps-команд хибними сповіщеннями.

Запропоновано Гібридна модель класифікації, що поєднує графовий аналіз потоків даних (GNN) із семантичною інтерпретацією коду на базі великих мовних моделей (LLM). Такий підхід реалізовано вперше для задач безпеки веб-застосунків.

Експерименти показали приріст F1-оцінки до 0,90 (на 13 %) та збільшення ROC-AUC до 0,95. Частка хибних спрацювань зменшилася з 18 % до 8 %, що становить 55,5 % зниження порівняно з Random Forest.

Розроблено модель оцінювання, динамічний ризик-скоринг , який доповнює CVSS гнучкими коефіцієнтами експозиції та бізнес-важливості, дозволяючи формувати пріоритети виправлення вразливостей.

Комплексний аналіз економічної ефективності підтвердив високий економічний ефект: ROI понад 108 %, чиста приведена вартість перевищує 430 тис. USD, а строк окупності - менше 6 місяців.

Новизна полягає у формалізації оцінки узагальнюючої похибки для задач виявлення вразливостей та вдосконаленні методів агрегування даних. Практичний ефект полягає у готовності алгоритмів до інтеграції в CI/CD-конвеєри та корпоративні системи управління ризиками, що здатне зменшити фінансові втрати від кіберінцидентів.

Література

1. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання. [Уведено вперше ; чинний від 2016-07-01]. Київ : ДП «УкрНДНЦ», 2016. 17 с. URL: https://nv-oneu.com.ua/downloads/dstu-8302-2015.pdf

2. Кашталян А., Лисенко С., Савенко Б., Сохор Т., Кисіль Т. Принцип та метод синтезу систем децепції для виявлення шкідливого програмного забезпечення та комп'ютерних атак. *Радіоелектронні і комп'ютерні системи*. 2023. № 4. С. 112–151. https://doi.org/10.32620/reks.2023.4.10

3. OWASP Top 10:2021. The Ten Most Critical Web Application Security Risks. *OWASP Foundation*. URL: https://owasp.org/www-project-top-ten/

4. Білодід Д. В. Методика запобігання вразливостей front-end частини web-застосунків : кваліфікаційна робота магістра : 121 Інженерія програмного забезпечення. Київ : ДУІКТ, 2025. 84 с. URL: https://surl.li/vzrnlw

5. Метод пошуку вразливостей вебзастосунків з використанням API ChatGPT / В. В. Поліщук та ін. *Сучасні інформаційні технології в будівництві та архітектурі*. 2024. № 318240. URL: https://smarttech.knuba.edu.ua/article/view/318240

6. Kashtalian A., Lysenko S., Sachenko A. Evaluation criteria of centralization options in the architecture of multicomputer systems with traps and baits. *Radioelectronic and Computer Systems*. 2025. No. 1. P. 264–297.

7. Alshomrani M. Survey of Transformer-Based Malicious Software Detection Systems. *Electronics*. 2024. Vol. 13, No. 4677. DOI: https://doi.org/10.3390/electronics13234677

8. Modern Approaches to Software Vulnerability Detection: A Survey of Machine Learning, Deep Learning, and Large Language Models / T. Zeng et al. *Electronics*. 2025. Vol. 14, No. 22. URL: https://www.mdpi.com/2079-9292/14/22/4449

9. Varga A. A Machine Learning-enhanced web-crawler for vulnerability detection: A binary classification approach : Master's thesis : AI and Machine Learning. Karlskrona : Blekinge Institute of Technology, 2025. 62 p. URL: https://www.diva-portal.org/smash/get/diva2%3A1962633/FULLTEXT01.pdf

10. Web Vulnerabilities using Machine Learning for Prevention and Detection: A Critical Review *International Journal on Emerging Technologies*, Vol. 16, Issue 2, pp. 120–139 (2025) URL: https://www.researchtrend.net/ijet/pdf/Web-Vulnerabilities-using-Machine-Learning-for-Prevention-and-Detection-A-Critical-Review-Oduleye-BE-18.pdf

11. Artificial Intelligence-Driven Supervised Classification Algorithm for Website Vulnerability Detection Using MITRE NVD CVE Scores / J. Doe et al. *Preprints.org*. 2026. URL: https://www.preprints.org/manuscript/202602.0475

12. Busko V. Automatic exploit assessment based on deep learning methods. *Ontology of Designing*. 2024. Vol. 14, No. 2. P. 156–170. URL: https://doi.org/10.18287/2223-9537-2024-14-3-408-420

13. Automated Vulnerability Assessment Using Machine Learning / R. Sharma et al. *ResearchGate*. 2024. URL: https://www.researchgate.net/publication/382918034

14. AI-Driven Automated Vulnerability Scanning for Real-Time Threat Detection and Mitigation / K. Modi et al. *International Journal of Innovative Research in Science, Engineering and Technology*. 2025. Vol. 14, No. 3. https://doi.org/10.15680/IJIRSET.2025.1403324

15. Boucena A. Leveraging Large Language Models For Automated Software Vulnerability Detection and Analysis : Master's thesis. Guelma : University of Guelma, 2025. 95 p. URL: https://dspace.univ-guelma.dz/jspui/bitstream/123456789/18272/1/F5_8_BOUCENA_AMINA_1752072283.pdf

16. Automatic Source Code Vulnerability Detection, Classification, and Prioritization Using Deep Learning / S. Jalowski URL: https://assets-eu.researchsquare.com/files/rs-7423339/v1_covered_14df0d01-894d-4139-86d7-37ff5eae83a7.pdf

17. Савенко О. С., Лисенко С. М., Нічепорук А. О. Виявлення бот-мереж на основі аналізу поведінки мережевих об'єктів у розподілених системах. *Комп'ютерні системи та мережі*. 2020. № 19. С. 190–198. URL: https://computingonline.net/computing/article/view/1761

18. Common Weakness Enumeration (CWE). *MITRE Corporation*. 2025. URL: https://cwe.mitre.org/

19. Gyamfi N. K., Goranin N. Automated System-Level Malware Detection Using Machine Learning: A Comprehensive Review. *Applied Sciences*. 2023. Vol. 13, No. 11908. URL: https://www.mdpi.com/2076-3417/13/21/11908

20. Phishing Website Detection Using Machine Learning / A. Kumar et al. *Dialnet*. 2025. URL: https://dialnet.unirioja.es/descarga/articulo/9930098.pdf

References

1. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання. [Уведено вперше ; чинний від 2016-07-01]. Київ : ДП «УкрНДНЦ», 2016. 17 с. URL: https://nv-oneu.com.ua/downloads/dstu-8302-2015.pdf

2. Кашталян А., Лисенко С., Савенко Б., Сохор Т., Кисіль Т. Принцип та метод синтезу систем децепції для виявлення шкідливого програмного забезпечення та комп'ютерних атак. *Радіоелектронні і комп'ютерні системи*. 2023. № 4. С. 112–151. DOI: https://doi.org/10.32620/reks.2023.4.10

3. OWASP Top 10:2021. The Ten Most Critical Web Application Security Risks. *OWASP Foundation*. URL: https://owasp.org/www-project-top-ten/

4. Білодід Д. В. Методика запобігання вразливостей front-end частини web-застосунків : кваліфікаційна робота магістра : 121 Інженерія програмного забезпечення. Київ : ДУІКТ, 2025. 84 с. URL: https://surl.li/vzrnlw

5. Метод пошуку вразливостей вебзастосунків з використанням API ChatGPT / В. В. Поліщук та ін. *Сучасні інформаційні технології в будівництві та архітектурі*. 2024. № 318240. URL: https://smarttech.knuba.edu.ua/article/view/318240

6. Kashtalian A., Lysenko S., Sachenko A. Evaluation criteria of centralization options in the architecture of multicomputer systems with traps and baits. *Radioelectronic and Computer Systems*. 2025. No. 1. P. 264–297.

7. Alshomrani M. Survey of Transformer-Based Malicious Software Detection Systems. *Electronics*. 2024. Vol. 13, No. 4677. https://doi.org/10.3390/electronics13234677

8. Modern Approaches to Software Vulnerability Detection: A Survey of Machine Learning, Deep Learning, and Large Language Models / T. Zeng et al. *Electronics*. 2025. Vol. 14, No. 22. URL: https://www.mdpi.com/2079-9292/14/22/4449

9. Varga A. A Machine Learning-enhanced web-crawler for vulnerability detection: A binary classification approach : Master's thesis : AI and Machine Learning. Karlskrona : Blekinge Institute of Technology, 2025. 62 p. URL: https://www.diva-portal.org/smash/get/diva2%3A1962633/FULLTEXT01.pdf

10. Web Vulnerabilities using Machine Learning for Prevention and Detection: A Critical Review *International Journal on Emerging Technologies*, Vol. 16, Issue 2, pp. 120–139 (2025) URL: https://www.researchtrend.net/ijet/pdf/Web-Vulnerabilities-using-Machine-Learning-for-Prevention-and-Detection-A-Critical-Review-Oduleye-BE-18.pdf

11. Artificial Intelligence-Driven Supervised Classification Algorithm for Website Vulnerability Detection Using MITRE NVD CVE Scores / J. Doe et al. *Preprints.org*. 2026. URL: https://www.preprints.org/manuscript/202602.0475

12. Busko V. Automatic exploit assessment based on deep learning methods. *Ontology of Designing*. 2024. Vol. 14, No. 2. P. 156–170. URL: https://doi.org/10.18287/2223-9537-2024-14-3-408-420

13. Automated Vulnerability Assessment Using Machine Learning / R. Sharma et al. *ResearchGate*. 2024. URL: https://www.researchgate.net/publication/382918034

14. AI-Driven Automated Vulnerability Scanning for Real-Time Threat Detection and Mitigation / K. Modi et al. *International Journal of Innovative Research in Science, Engineering and Technology*. 2025. Vol. 14, No. 3. DOI: 10.15680/IJIRSET.2025.1403324 URL: https://www.ijirset.com/upload/2025/march/324_AI-Driven.pdf

15. Boucena A. Leveraging Large Language Models For Automated Software Vulnerability Detection and Analysis : Master's thesis. Guelma : University of Guelma, 2025. 95 p. URL: https://dspace.univ-guelma.dz/jspui/bitstream/123456789/18272/1/F5_8_BOUCENA_AMINA_1752072283.pdf

16. Automatic Source Code Vulnerability Detection, Classification, and Prioritization Using Deep Learning / S. Jalowski URL: https://assets-eu.researchsquare.com/files/rs-7423339/v1_covered_14df0d01-894d-4139-86d7-37ff5eae83a7.pdf

17. Савенко О. С., Лисенко С. М., Нічепорук А. О. Виявлення бот-мереж на основі аналізу поведінки мережевих об'єктів у розподілених системах. *Комп'ютерні системи та мережі*. 2020. № 19. С. 190–198. URL: https://computingonline.net/computing/article/view/1761

18. Common Weakness Enumeration (CWE). *MITRE Corporation*. 2025. URL: https://cwe.mitre.org/

19. Gyamfi N. K., Goranin N. Automated System-Level Malware Detection Using Machine Learning: A Comprehensive Review. *Applied Sciences*. 2023. Vol. 13, No. 11908. URL: https://www.mdpi.com/2076-3417/13/21/11908

20. Phishing Website Detection Using Machine Learning / A. Kumar et al. *Dialnet*. 2025. URL: https://dialnet.unirioja.es/descarga/articulo/9930098.pdf