

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Кваліфікаційна наукова
праця на правах рукопису

ШАХМАТОВ ІВАН ОЛЕКСАНДРОВИЧ

УДК 004.42: 004.9: 004.8: 004.65: 004.67

ДИСЕРТАЦІЯ
МОДЕЛІ ТА МЕТОДИ ЗАБЕЗПЕЧЕННЯ ДОВІРИ Й
ЦІЛІСНОСТІ У ВЕБСИСТЕМАХ

Спеціальність 121 «Інженерія програмного забезпечення»

Галузь знань 12 «Інформаційні технології»

Подається на здобуття наукового ступеня доктора філософії.

Дисертація містить результати власних досліджень. Використання ідей,
результатів і текстів інших авторів мають посилання на відповідне джерело.

_____Іван ШАХМАТОВ

Науковий керівник:

ЗАМРІЙ Ірина Вікторівна, доктор технічних наук, професор

Київ - 2026

АНОТАЦІЯ

Шахматов І.О. Моделі та методи забезпечення довіри й цілісності у вебсистемах - Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 121 Інженерія програмного забезпечення. - Державний університет інформаційно-комунікаційних технологій Міністерства освіти і науки України, Київ, 2026.

Дисертаційну роботу присвячено дослідженню та розробленню моделей і методів підвищення захищеності вебсистем шляхом забезпечення довіри до критичних подій і цілісності даних під час їх обробки, зберігання та передачі. Актуальність теми зумовлена широким використанням вебсервісів у сферах із підвищеними вимогами до надійності та захищеності, а також зростанням кількості кіберзагроз, що призводять до порушення цілісності інформації, неправомірного доступу, витоків даних і зниження доступності сервісів. Окремою практично значущою проблемою є вебспам у формах зворотного зв'язку та інших каналах введення даних, який перевантажує інфраструктуру, погіршує якість сервісу і може використовуватися як супутній інструмент для подальших атак.

Метою дослідження є підвищення рівня довіри, цілісності та захищеності архітектури вебсистем, за рахунок обґрунтування та побудови моделей і методів, що забезпечують контроль цілісності даних, визначення критичних подій і адаптивне виявлення підозрілої активності у вебсередовищі на основі технологій блокчейну та машинного навчання.

Для досягнення мети здійснено аналіз загроз і вимог до довіри й цілісності у вебсистемах, розглянуто підходи до фіксації та перевірки критичних змін і подій, сформовано вимоги до запропонованого рішення, а також розроблено й оцінено методи, орієнтовані на протидію порушенню цілісності даних, неправомірному доступу та атакам, що знижують надійність і керованість вебсервісів.

У результаті дослідження *вперше* розроблено модель інтегрованого контуру довіри й цілісності (ІКДЦ) у вебсистемі, що ґрунтується на кортежно-графовому поданні критичних подій і криптографічних принципах їх верифікації та за рахунок

поєднання незмінного журналювання критичних подій, формалізованого подання зв'язків між вебформами, SQL-операціями, рішеннями аналітичного модуля та політиками реагування забезпечує єдине інформаційне середовище для контролю цілісності даних, простежуваності подій, аудитної перевірки та відтворюваності рішень у вебсистемі.

Вперше розроблено метод блокчейн-верифікованого журналювання критичних подій і контролю доступу у вебсистемах, що ґрунтується на розробленій моделі ІКДЦ та теорії криптографічно зв'язаного ланцюга подій із хешуванням, цифровим підписом і пороговим правилом прийняття рішення щодо доступу, який дозволяє зменшити ризик прихованої модифікації інформації, підвищити доказовість журналів і контроль цілісності даних під час розслідування інцидентів.

Вперше розроблено метод графово-нейромережевого виявлення вебспаму та підозрілої активності у вебсистемах, що ґрунтується на моделі ІКДЦ та багатопредставленому графовому описі подій, поданих через систему ознак технічного, змістовного, часово-поведінкового, контекстного характеру з урахуванням зв'язків між подіями й результатами аналітичного оцінювання, що забезпечує розрізнення легітимних, підозрілих і шкідливих звернень, підвищує точність виявлення вебспаму та зменшує частку хибних спрацювань.

Вперше розроблено метод інтегрованого забезпечення довіри й цілісності у вебсистемах, що ґрунтується на моделі ІКДЦ, методі блокчейн-верифікованого журналювання критичних подій і контролю доступу та методі графово-нейромережевого виявлення вебспаму й підозрілої активності, а також на теорії композиції функціональних відображень критичних подій у клас рішень, що забезпечує цілісність системи, простежуваність та точність прийняття рішень.

Отримані наукові результати показали, що застосування розробленої моделі ІКДЦ і запропонованих методів дозволяє підвищити F1-міру виявлення підозрілої активності на 17.9% для класу SUBMIT і на 21.6% для домену TX, зменшити частку хибних спрацювань відповідно на 65.4% і 57.9%, а також підтвердити ефективність виявлення окремих сценаріїв порушення цілісності та неправомірного втручання.

Практичне значення отриманих результатів полягає у можливості використання розробленої моделі ІКДЦ, методу забезпечення довіри й цілісності критичних подій і даних, методу виявлення вебспау та підозрілої активності у вебсистемах та методу інтегрованого забезпечення довіри й цілісності у вебсистемах як основи для побудови або модернізації вебсистем із підвищеними вимогами до безпеки, простежуваності та аудиту. Запропоновані рішення використовуються для захисту вебформ, контролю критичних SQL-операцій, виявлення підозрілої активності у вебтрафіку, зменшення навантаження на адміністраторів у задачах модерації та аналізу інцидентів, а також під час проєктування програмних модулів для систем електронної комерції, корпоративних вебзастосунків, інформаційних сервісів і вебплатформ, що працюють із критичними даними.

У *першому розділі* дисертації проведено аналіз сучасного стану забезпечення довіри й цілісності у вебсистемах. Обґрунтовано актуальність теми, розглянуто принципи побудови безпечних вебсистем і підходи до оцінювання рівня їх безпеки. Встановлено обмеження традиційних механізмів захисту та журналювання щодо забезпечення незмінності критичних подій і доведення коректності виконаних дій. Проаналізовано блокчейн-підходи до фіксації подій, контролю цілісності та аудиту змін, а також методи машинного навчання для виявлення атак, аномалій, вебспау й підозрілої активності. За результатами аналізу обґрунтовано необхідність інтегрованого підходу, що поєднує криптографічну фіксацію подій, контроль цілісності, аудит змін та інтелектуальне виявлення загроз. Сформульовано наукове завдання, мету та завдання дослідження.

У *другому розділі* дисертації розроблено модель інтегрованого контуру довіри й цілісності у вебсистемах та метод блокчейн-верифікованого журналювання критичних подій і контролю доступу. Запропонована модель ґрунтується на кортежно-графовому поданні критичних подій і криптографічних принципах їх верифікації, що дозволяє формалізувати зв'язки між вебформами, SQL-операціями, рішеннями аналітичного модуля, політиками реагування та аудитними записами. Розроблений метод забезпечує криптографічне зв'язування подій із використанням хешування, цифрового підпису та порогового правила прийняття рішень щодо

доступу. Проведено адаптацію методу для перевірки SQL-операцій, подій доступу та аудитного аналізу активності користувачів із використанням фільтра Блума, метрики Жаккара, кривої Лоренца та коефіцієнта Джині.

У *третьому розділі* дисертації розроблено метод графово-нейромережевого виявлення вебспаму та підозрілої активності у вебсистемах. Сформовано підхід до підготовки вхідних даних для аналізу вебконтенту та поведінки користувачів, що передбачає очищення, кодування, нормалізацію, масштабування даних і врахування фінансових, географічних, часових та поведінкових характеристик подій. Формалізовано багатопредставлений графовий опис подій вебсистеми, у якому повідомлення, дії користувачів, технічні параметри та зв'язки між об'єктами взаємодії розглядаються як елементи єдиного потоку подій.

У *четвертому розділі* дисертації розроблено метод інтегрованого забезпечення довіри й цілісності у вебсистемах, який поєднує модель ІКДЦ, метод блокчейн-верифікованого журналювання критичних подій і метод графово-нейромережевого виявлення вебспаму та підозрілої активності. Розроблено адаптацію моделі ІКДЦ для прототипу захисту платіжного контуру, а також архітектуру і програмну реалізацію інтегрованого контуру довіри й цілісності. Проведено експериментальну перевірку на потоках подій типу SUBMIT і TX, та отримано результати, які показали підвищення F1-міри, зменшення частки хибних спрацювань і підтвердили практичну придатність запропонованого методу для вебсистем із підвищеними вимогами до безпеки, простежуваності та аудитної перевірки.

Результати дисертаційної роботи реалізовано у вигляді моделі підсистеми довіри й цілісності у вебсистемах, методів забезпечення цілісності критичних подій, виявлення вебспаму та інтегрованого забезпечення довіри й цілісності, а також програмного прототипу для їх експериментальної перевірки. Основні положення та результати дисертаційного дослідження впроваджено у ТОВ «ШЛІФАРБ», ТОВ «АРМА МОТОРС КИЇВ», Інституті програмних систем НАН України, Державному університеті інформаційно-комунікаційних технологій, що підтверджується відповідними актами впровадження.

Авторський внесок полягає у формалізації задачі забезпечення довіри й цілісності у вебсистемах, побудові моделі ІКДЦ, створенні методу забезпечення довіри й цілісності критичних подій і даних, методу виявлення вебспаму та підозрілої активності у вебсистемах на основі ІКДЦ і методу інтегрованого забезпечення довіри й цілісності у вебсистемах, що ґрунтується на комбінації моделі ІКДЦ, блокчейн-верифікованого журналювання критичних подій і графово-нейромережевого аналізу підозрілої активності, а також у реалізації прототипу програмного рішення та проведенні експериментальної перевірки запропонованих рішень з аналізом отриманих результатів.

Дисертаційну роботу виконано відповідно до планів наукової і науково-технічної діяльності Державного університету інформаційно-комунікаційних технологій у межах науково-дослідної роботи «Забезпечення функціональної стійкості інформаційних систем підприємства в умовах впливу дестабілізуючих факторів із застосуванням нейронних мереж» (державний реєстраційний номер 0226U000249) та науково-дослідної роботи «Методи побудови функціонально стійких захищених інформаційних систем з централізованим управлінням» (державний реєстраційний номер 0125U002823).

Основні результати дослідження доповідалися та обговорювалися на наукових і науково-практичних конференціях різного рівня.

Ключові слова: цілісність даних, вебсистеми, архітектура вебзастосунків, інтегрований контур довіри, підсистема забезпечення довіри, незмінне журналювання, аудит подій, блокчейн, SQL-ін'єкції, вебспам, нейронні мережі, програмне забезпечення, машинне навчання, якість програмного забезпечення.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА

Наукові праці, в яких опубліковані основні наукові результати дисертації

Статті в наукових фахових виданнях

1. Шахматов І.О. Технологія Blockchain як інструмент протидії неправомірному використанню доступу до вебсайтів. Зв'язок, № 1 2024. С.20-25. DOI: 10.31673/2412-9070.2024.012025.
2. Шахматов І.О., Замрій І.В. Потенціал блокчейну у покращенні безпеки вебсайтів. Сучасний захист інформації, №1(57) 2024. С.28-38. DOI: 10.31673/2409-7292.2024.010004.
3. Замрій І.В., Шахматов І.О. Підвищення безпеки вебзастосунків через інноваційні патерни інтеграції штучного інтелекту. Сучасний стан наукових досліджень та технологій в промисловості, № 1(27) 2024. С. 67-80 DOI: 10.30837/ITSSI.2024.27.067.
4. Замрій І. В., Шахматов І. О. Інтегрована система безпеки для захисту синхронізації платежів від MITM-атак. Проблеми програмування. № 2 2025. С. 28-39. DOI: 10.15407/pp2025.02.028.
5. Замрій І. В., Шахматов І. О. Автоматизація оцінки безпеки вебзастосунків засобами Python. Зв'язок. № 4 (176) 2025. С. 58-66. DOI: 10.31673/2412-9070.2025.045866.
6. Шахматов І. О. Інтегрований контур довіри у вебзастосунках на основі графового оцінювання ризику та незмінного журналювання рішень. Зв'язок. № 2 (180) 2026. С.72-78. DOI:10.31673/2412-9070.2026.024909.

Публікації в наукових фахових виданнях, що індексуються в міжнародних базах Scopus

7. Zhurakovskiy, B., Averichev, I., and Shakhmatov, I. Using the Latest Methods of Cluster Analysis to Identify Similar Profiles in Leading Social Networks. CEUR Workshop Proceedings; 10th International Scientific Conference Information Technology and Implementation, IT and I-WS 2023; Volume 3646, 2023. Pages 116-126. URL: https://ceur-ws.org/Vol-3646/Paper_12.pdf (SCOPUS).

8. Замрій І.В., Шахматов І.О., Яскевич В.О. Blockchainsqlsecure: Інтеграція блокчейн-технології для зміцнення захисту від SQL-ін'єкцій. Вісник Київського національного університету імені Тараса Шевченка. Фізико-математичні науки, № 1(78) 2024. С. 160-168. DOI: 10.17721/1812-5409.2024/1.29 (SCOPUS).

9. I. Zamrii, I. Shakhmatov, O. Yudin, Y. Diana, M. Tyshchenko and Y. Rudenko, Methods for Detecting DDoS Attacks in Web Traffic Using Autoencoders with an Adaptive Three-Level Approach. 2024 IEEE 5th International Conference on Advanced Trends in Information Theory (ATIT), Lviv, Ukraine, 2024, pp. 1-5, DOI: 10.1109/ATIT64324.2024.11222524 (SCOPUS).

10. Zamrii I., Shakhmatov I. Multi-View Graph Model with Representation Alignment and Adaptive Fusion for Better Spam Detection. Proceedings of the Workshop on Cryptology and Data Security (WCDS 2025), co-located with SMICS 2025, Lviv, Ukraine, October 16-18, 2025, CEUR Workshop Proceedings, 2026, Vol. 4191. P. 99-106. URL: <https://ceur-ws.org/Vol-4191/short8.pdf>.(SCOPUS).

Наукові праці, які засвідчують апробацію матеріалів дисертації

11. Шахматов І.О., Замрій І.В. Технологія блокчейн як інструмент протидії неправомірному використанню доступу до вебсайтів. V Міжнародна науково-практична конференція молодих вчених та студентів «Інженерія програмного забезпечення і передові інформаційні технології (Soft Tech-2023)», 19 - 21 грудня 2023 року, Київ, С. 360-364.

12. Замрій І.В., Шахматов І.А. Посилення безпеки вебідентифікації через технологію блокчейн. Всеукраїнська науково-технічна конференція «Застосування програмного забезпечення в інформаційно-комунікаційних технологіях», 24 квітня 2024 року, 2024 року, Київ, С. 249-253.

13. Шахматов І.О., Замрій І.В. Використання блокчейн-технології для підвищення безпеки від SQL ін'єкцій. XIII Міжнародна науково-технічна конференція «Безпека інформаційних технологій: ITSEC-2024», 9-11 травня 2024 року, Львів, С.98 - 101.

14. Шахматов І. О., Замрій І. В. Технології масштабування даних у боротьбі з DDOS-атаками. Науково-практична конференція «Штучний інтелект і безпека», 19- 21 листопада 2024 року, Київ, С. 27-30.
15. Білодід Д. В., Шахматов І. О. Ефективність CSRF-токенів у запобіганні міжсайтовим запитам у фронтенд-додатках. Всеукраїнська науково-технічна конференція «Виклики та рішення в програмній інженерії», 26 листопада 2024 року, Київ, С. 92-96.
16. Шахматов І.О., Замрій І.В. Адаптивні нейромережі у боротьбі з вебспамом. XIV Міжнародна науково-технічна конференція «Безпека інформаційних технологій: ITSEC-2025», 22-24 травня 2025 року, м. Тернопіль, С. 211-213.
17. Замрій І.В., Шахматов І.О. Мультирепрезентаційна GNN-модель з узгодженням і адаптивним злиттям для детекції спаму. «SMICS: Безпека сучасних інформаційно-комунікаційних систем», 16-18 жовтня 2025 року, м. Львів, С. 320-325.
18. Замрій І.В., Нестеренко К.С., Задонцев Ю.В., Глушкова О.І., Шахматов І.О. Методика підвищення функціональної стійкості інформаційної системи через виявлення вторгнень та реконфігурації мережі. XIV Міжнародна науково-практична конференція «Математика. Інформаційні технології. Освіта», 13-15 червня 2025 року, Луцьк - Світязь, С. 101-104.
19. Бовкун І.В., Шахматов І.О. Визначення вимог до вебсистеми для управління безконтактними замовленнями у ресторані. II Всеукраїнська науково-технічна конференція «Виклики та рішення в програмній інженерії», 26 листопада 2025року, Київ, С. 432-434.

ANNOTATION

Shakhmatov I.O. Models and Methods for Ensuring Trust and Integrity in Web Systems. - Qualifying scientific work submitted as a manuscript.

Dissertation for the degree of Doctor of Philosophy in specialty 121 Software Engineering. - State University of Information and Communication Technologies of the Ministry of Education and Science of Ukraine, Kyiv, 2026.

The dissertation is devoted to the study and development of models and methods for improving the security of web systems by ensuring trust in critical events and data integrity during their processing, storage, and transmission. The relevance of the topic is caused by the wide use of web services in areas with increased requirements for reliability and security, as well as by the growing number of cyber threats that lead to data integrity violations, unauthorized access, data leakage, and reduced service availability. A separate practically important problem is web spam in feedback forms and other data input channels, which overloads infrastructure, reduces service quality, and may be used as an additional tool for further attacks.

The aim of this research is to increase the level of trust, integrity, and security of web system architecture by substantiating and developing models and methods that ensure data integrity control, identification of critical events, and adaptive detection of suspicious activity in the web environment based on blockchain technologies and machine learning.

To achieve this aim, threats and requirements for trust and integrity in web systems were analyzed; approaches to recording and verifying critical changes and events were considered; requirements for the proposed solution were formed; and methods focused on counteracting data integrity violations, unauthorized access, and attacks that reduce the reliability and manageability of web services were developed and evaluated.

As a result of the research, *for the first time*, a model of the Integrated Trust and Integrity Circuit in a web system was developed. The model is based on a tuple-

graph representation of critical events and cryptographic principles of their verification. By combining immutable logging of critical events, a formalized representation of relations between web forms, SQL operations, analytical module decisions, and response policies, the model provides a unified information environment for data integrity control, event traceability, audit verification, and reproducibility of decisions in a web system.

For the first time, a method of blockchain-verified logging of critical events and access control in web systems was developed. The method is based on the developed ITIC model and on the theory of a cryptographically linked chain of events using hashing, digital signatures, and a threshold rule for making access decisions. The method reduces the risk of hidden information modification, improves the evidential value of logs, and strengthens data integrity control during incident investigation.

For the first time, a method of graph neural network-based detection of web spam and suspicious activity in web systems was developed. The method is based on the ITIC model and on a multi-representational graph description of events represented through a system of technical, content-related, temporal-behavioural, and contextual features, taking into account the relationships between events and the results of analytical evaluation. This makes it possible to distinguish legitimate, suspicious, and harmful requests, improve the accuracy of web spam detection, and reduce the false positive rate.

For the first time, a method of integrated trust and integrity assurance in web systems was developed. The method is based on the ITIC model, the method of blockchain-verified logging of critical events and access control, and the method of graph neural network-based detection of web spam and suspicious activity. It also relies on the theory of composition of functional mappings of critical events into a class of decisions, which ensures system integrity, event traceability, and the accuracy of decision-making.

The obtained scientific results showed that the use of the developed ITIC model and the proposed methods makes it possible to increase the F1-score of

suspicious activity detection by 17.9% for the SUBMIT class and by 21.6% for the TX domain, reduce the false positive rate by 65.4% and 57.9%, respectively, and confirm the effectiveness of detecting individual scenarios of integrity violation and unauthorized interference.

The practical value of the obtained results lies in the possibility of using the developed ITIC model, the method for ensuring trust and integrity of critical events and data, the method for detecting web spam and suspicious activity in web systems, and the method of integrated trust and integrity assurance in web systems as a basis for building or modernizing web systems with increased requirements for security, traceability, and audit. The proposed solutions are used for protecting web forms, controlling critical SQL operations, detecting suspicious activity in web traffic, reducing the workload of administrators in moderation and incident analysis tasks, and designing software modules for e-commerce systems, corporate web applications, information services, and web platforms that process critical data.

In the first chapter of the dissertation, an analysis of the current state of ensuring trust and integrity in web systems was carried out. The relevance of the topic was substantiated, the principles of building secure web systems were considered, and approaches to assessing their level of security were examined. The limitations of traditional protection and logging mechanisms in ensuring the immutability of critical events and proving the correctness of performed actions were identified. Blockchain-based approaches to event recording, integrity control, and change auditing were analysed, as well as machine learning methods for detecting attacks, anomalies, web spam, and suspicious activity. Based on the analysis, the need for an integrated approach was substantiated, combining cryptographic recording of events, integrity control, change auditing, and intelligent threat detection. The scientific problem, aim, and research objectives were formulated.

In the second chapter of the dissertation, a model of the Integrated Trust and Integrity Circuit for web systems and a method of blockchain-verified logging of critical events and access control were developed. The proposed model is based on

a tuple-graph representation of critical events and cryptographic principles of their verification, which makes it possible to formalize the relationships between web forms, SQL operations, decisions of the analytical module, response policies, and audit records. The developed method ensures the cryptographic linking of events using hashing, digital signatures, and a threshold-based decision rule for access control. The method was adapted for the verification of SQL operations, access events, and audit analysis of user activity using a Bloom filter, the Jaccard metric, the Lorenz curve, and the Gini coefficient.

In the third chapter of the dissertation, a method for graph-neural-network-based detection of web spam and suspicious activity in web systems was developed. An approach to preparing input data for the analysis of web content and user behaviour was formed. It includes data cleaning, encoding, normalization, scaling, and consideration of the financial, geographical, temporal, and behavioural characteristics of events. A multi-representational graph description of web system events was formalized, in which messages, user actions, technical parameters, and relationships between interaction objects are considered as elements of a unified event flow.

In the fourth chapter of the dissertation, a method for integrated assurance of trust and integrity in web systems was developed. This method combines the ITIC model, the method of blockchain-verified logging of critical events, and the method of graph-neural-network-based detection of web spam and suspicious activity. The ITIC model was adapted for a prototype of payment framework protection, and the architecture and software implementation of the integrated trust and integrity framework were developed. An experimental evaluation was carried out on event flows of the SUBMIT and TX types. The obtained results showed an increase in the F1-score, a reduction in the false positive rate, and confirmed the practical suitability of the proposed method for web systems with increased requirements for security, traceability, and audit verification.

The results of the dissertation were implemented as a model of a trust and integrity subsystem in web systems, methods for ensuring the integrity of critical

events, methods for detecting web spam, a method of integrated trust and integrity assurance, and a software prototype for their experimental verification. The main provisions and results of the dissertation research were implemented at SHLIFARB LLC, ARMA MOTORS KYIV LLC, the Institute of Software Systems of the National Academy of Sciences of Ukraine, and the State University of Information and Communication Technologies, as confirmed by the relevant implementation acts.

The author's contribution consists in formalizing the problem of ensuring trust and integrity in web systems, developing the ITIC model, creating the method for ensuring trust and integrity of critical events and data, the method for detecting web spam and suspicious activity in web systems based on ITIC, and the method of integrated trust and integrity assurance in web systems. This method is based on a combination of the ITIC model, blockchain-verified logging of critical events, and graph neural network-based analysis of suspicious activity. The author also implemented a software prototype and carried out experimental verification of the proposed solutions with analysis of the obtained results.

The dissertation was carried out in accordance with the plans of scientific and research-and-development activities of the State University of Information and Communication Technologies within the research work "Ensuring the Functional Stability of Enterprise Information Systems under the Influence of Destabilizing Factors Using Neural Networks" (state registration number 0226U000249) and the research work "Methods for Building Functionally Stable Secure Information Systems with Centralized Management" (state registration number 0125U002823).

The main research results were presented and discussed at scientific and scientific-practical conferences of different levels.

Keywords: data integrity, web systems, web application architecture, integrated trust circuit, trust assurance subsystem, immutable logging, event audit, blockchain, SQL injections, web spam, neural networks, software, machine learning, software quality.

LIST OF PUBLICATIONS BY THE CANDIDATE

Scientific Papers Containing the Main Scientific Results of the Dissertation

Articles in Specialized Scientific Journals

1. Shakhmatov I.O. Blockchain Technology as a Tool for Counteracting Unauthorized Use of Access to Websites. *Zviazok*, No. 1, 2024. Pp. 20–25. DOI: 10.31673/2412-9070.2024.012025.
2. Shakhmatov I.O., Zamrii I.V. The Potential of Blockchain in Improving Website Security. *Modern Information Security*, No. 1(57), 2024. Pp. 28–38. DOI: 10.31673/2409-7292.2024.010004.
3. Zamrii I.V., Shakhmatov I.O. Improving Web Application Security through Innovative Patterns of Artificial Intelligence Integration. *Current State of Scientific Research and Technologies in Industry*, No. 1(27), 2024. Pp. 67–80. DOI: 10.30837/ITSSI.2024.27.067.
4. Zamrii I.V., Shakhmatov I.O. An Integrated Security System for Protecting Payment Synchronization against MITM Attacks. *Problems in Programming*, No. 2, 2025. Pp. 28–39. DOI: 10.15407/pp2025.02.028.
5. Zamrii I.V., Shakhmatov I.O. Automation of Web Application Security Assessment Using Python. *Zviazok*, No. 4(176), 2025. Pp. 58–66. DOI: 10.31673/2412-9070.2025.045866.
6. Shakhmatov I.O. An Integrated Trust Circuit in Web Applications Based on Graph-Based Risk Assessment and Immutable Decision Logging. *Zviazok*, No. 2(180), 2026. Pp. 72–78. DOI: 10.31673/2412-9070.2026.024909.

Publications in Specialized Scientific Journals Indexed in International Scopus Databases

7. Zhurakovskiy B., Averichev I., Shakhmatov I. Using the Latest Methods of Cluster Analysis to Identify Similar Profiles in Leading Social Networks. *CEUR Workshop Proceedings; 10th International Scientific Conference Information Technology and Implementation, IT&I-WS 2023*, Vol. 3646, 2023. Pp. 116–126. URL: https://ceur-ws.org/Vol-3646/Paper_12.pdf (SCOPUS).

8. Zamrii I.V., Shakhmatov I.O., Yaskevych V.O. BlockchainSQLSecure: Integration of Blockchain Technology to Strengthen Protection against SQL Injections. Bulletin of Taras Shevchenko National University of Kyiv. Physics and Mathematics, No. 1(78), 2024. Pp. 160–168. DOI: 10.17721/1812-5409.2024/1.29 (SCOPUS).

9. Zamrii I., Shakhmatov I., Yudin O., Diana Y., Tyshchenko M., Rudenko Y. Methods for Detecting DDoS Attacks in Web Traffic Using Autoencoders with an Adaptive Three-Level Approach. 2024 IEEE 5th International Conference on Advanced Trends in Information Theory (ATIT), Lviv, Ukraine, 2024. Pp. 1–5. DOI: 10.1109/ATIT64324.2024.11222524 (SCOPUS).

10. Zamrii I., Shakhmatov I. Multi-View Graph Model with Representation Alignment and Adaptive Fusion for Better Spam Detection. Proceedings of the Workshop on Cryptology and Data Security (WCDS 2025), co-located with SMICS 2025, Lviv, Ukraine, October 16–18, 2025. CEUR Workshop Proceedings, 2026, Vol. 4191. Pp. 99–106. URL: <https://ceur-ws.org/Vol-4191/short8.pdf> (SCOPUS).

Scientific Papers Confirming the Approbation of the Dissertation Materials

11. Shakhmatov I.O., Zamrii I.V. Blockchain Technology as a Tool for Counteracting Unauthorized Use of Access to Websites. V International Scientific and Practical Conference of Young Scientists and Students “Software Engineering and Advanced Information Technologies (SoftTech-2023)”, December 19–21, 2023, Kyiv. Pp. 360–364.

12. Zamrii I.V., Shakhmatov I.O. Strengthening Web Identification Security through Blockchain Technology. All-Ukrainian Scientific and Technical Conference “Application of Software in Information and Communication Technologies”, April 24, 2024, Kyiv. Pp. 249–253.

13. Shakhmatov I.O., Zamrii I.V. Use of Blockchain Technology to Improve Protection against SQL Injections. XIII International Scientific and Technical Conference “Information Technology Security: ITSEC-2024”, May 9–11, 2024, Lviv. Pp. 98–101.

14. Shakhmatov I.O., Zamrii I.V. Data Scaling Technologies in Counteracting DDoS Attacks. Scientific and Practical Conference “Artificial Intelligence and Security”, November 19–21, 2024, Kyiv. Pp. 27–30.

15. Bilodid D.V., Shakhmatov I.O. The Effectiveness of CSRF Tokens in Preventing Cross-Site Requests in Front-End Applications. All-Ukrainian Scientific and Technical Conference “Challenges and Solutions in Software Engineering”, November 26, 2024, Kyiv. Pp. 92–96.
16. Shakhmatov I.O., Zamrii I.V. Adaptive Neural Networks in Counteracting Web Spam. XIV International Scientific and Technical Conference “Information Technology Security: ITSEC-2025”, May 22–24, 2025, Ternopil. Pp. 211–213.
17. Zamrii I.V., Shakhmatov I.O. A Multi-Representation GNN Model with Alignment and Adaptive Fusion for Spam Detection. SMICS: Security of Modern Information and Communication Systems, October 16–18, 2025, Lviv. Pp. 320–325.
18. Zamrii I.V., Nesterenko K.S., Zadontsev Yu.V., Hlushkova O.I., Shakhmatov I.O. A Method for Improving the Functional Stability of an Information System through Intrusion Detection and Network Reconfiguration. XIV International Scientific and Practical Conference “Mathematics. Information Technologies. Education”, June 13–15, 2025, Lutsk–Svityaz. Pp. 101–104.
19. Bovkun I.V., Shakhmatov I.O. Defining Requirements for a Web System for Managing Contactless Restaurant Orders. II All-Ukrainian Scientific and Technical Conference “Challenges and Solutions in Software Engineering”, November 26, 2025, Kyiv. Pp. 432–434.

ЗМІСТ

АНОТАЦІЯ.....	2
СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА	7
ANNOTATION.....	10
LIST OF PUBLICATIONS BY THE CANDIDATE	15
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	20
ВСТУП.....	21
РОЗДІЛ 1. АНАЛІЗ СУЧАСНОГО СТАНУ ТА ПРОБЛЕМАТИКИ ЗАБЕЗПЕЧЕННЯ ДОВІРИ І ЦІЛІСНОСТІ У ВЕБСИСТЕМАХ	32
1.1. ОБҐРУНТУВАННЯ ВИБОРУ ТЕМИ ТА АНАЛІЗ ПРОБЛЕМИ ЗАБЕЗПЕЧЕННЯ ДОВІРИ Й ЦІЛІСНОСТІ У ВЕБСИСТЕМАХ	33
1.2. ПРИНЦИПИ ПОБУДОВИ БЕЗПЕЧНИХ ВЕБСИСТЕМ І ПІДХОДИ ДО ОЦІНЮВАННЯ РІВНЯ ЇХ БЕЗПЕКИ	38
1.3. ПІДХОДИ НА ОСНОВІ БЛОКЧЕЙНУ ДЛЯ ФІКСАЦІЇ ПОДІЙ, ПЕРЕВІРКИ ЦІЛІСНОСТІ ТА АУДИТУ ЗМІН.....	47
1.4. МЕТОДИ МАШИННОГО НАВЧАННЯ ДЛЯ ВИЯВЛЕННЯ АТАК, АНОМАЛІЙ І ПІДОЗРІЛОЇ АКТИВНОСТІ У ВЕБСЕРЕДОВИЩІ	54
1.5. ПОСТАНОВКА НАУКОВОГО ЗАВДАННЯ ТА ВИЗНАЧЕННЯ МЕТИ І ЗАВДАНЬ ДОСЛІДЖЕННЯ.....	57
Висновки до розділу 1	60
РОЗДІЛ 2. МОДЕЛЬ ІНТЕГРОВАНОГО КОНТУРУ ДОВІРИ Й ЦІЛІСНОСТІ ТА МЕТОД БЛОКЧЕЙН-ВЕРИФІКОВАНОГО ЖУРНАЛЮВАННЯ КРИТИЧНИХ ПОДІЙ І КОНТРОЛЮ ДОСТУПУ У ВЕБСИСТЕМАХ	61
2.1. МОДЕЛЬ ІНТЕГРОВАНОГО КОНТУРУ ДОВІРИ Й ЦІЛІСНОСТІ У ВЕБСИСТЕМАХ	61
2.2. МЕТОД БЛОКЧЕЙН-ВЕРИФІКОВАНОГО ЖУРНАЛЮВАННЯ КРИТИЧНИХ ПОДІЙ І КОНТРОЛЮ ДОСТУПУ У ВЕБСИСТЕМАХ.....	70
2.3. АДАПТАЦІЯ МЕТОДУ БЛОКЧЕЙН-ВЕРИФІКОВАНОГО ЖУРНАЛЮВАННЯ КРИТИЧНИХ ПОДІЙ І КОНТРОЛЮ ДОСТУПУ ДЛЯ ПЕРЕВІРКИ SQL-ОПЕРАЦІЙ ТА АУДИТНОГО АНАЛІЗУ .	74
Висновки до розділу 2	80

РОЗДІЛ 3. МЕТОД ГРАФОВО-НЕЙРОМЕРЕЖЕВОГО ВИЯВЛЕННЯ ВЕБСПАМУ ТА ПІДОЗРІЛОЇ АКТИВНОСТІ У ВЕБСИСТЕМАХ.....	82
3.1. ФОРМУВАННЯ ДАНИХ І ОЗНАК ДЛЯ АНАЛІЗУ ВЕБКОНТЕНТУ ТА ПОВЕДІНКИ КОРИСТУВАЧІВ	82
3.2. МЕТОД ГРАФОВО-НЕЙРОМЕРЕЖЕВОГО ВИЯВЛЕННЯ ВЕБСПАМУ ТА ПІДОЗРІЛОЇ АКТИВНОСТІ У ВЕБСИСТЕМАХ	89
3.3. АДАПТАЦІЯ МЕТОДУ ГРАФОВО-НЕЙРОМЕРЕЖЕВОГО ВИЯВЛЕННЯ ВЕБСПАМУ ТА ПІДОЗРІЛОЇ АКТИВНОСТІ ДО ЗМІНИ ШАБЛОНІВ ЗАГРОЗ, ПРАВИЛ РЕАГУВАННЯ ТА МЕХАНІЗМІВ ЗАБЕЗПЕЧЕННЯ ЦІЛІСНОСТІ	95
Висновки до розділу 3	104
РОЗДІЛ 4 МЕТОД ІНТЕГРОВАНОГО ЗАБЕЗПЕЧЕННЯ ДОВІРИ Й ЦІЛІСНОСТІ У ВЕБСИСТЕМАХ, ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА	106
4.1. МЕТОД ІНТЕГРОВАНОГО ЗАБЕЗПЕЧЕННЯ ДОВІРИ Й ЦІЛІСНОСТІ У ВЕБСИСТЕМАХ.....	106
4.2. АДАПТАЦІЯ МОДЕЛІ ІКДЦ ТА ІНТЕГРОВАНОГО МЕТОДУ ДЛЯ ПОБУДОВИ ПРОТОТИПУ ЗАХИСТУ ПЛАТІЖНОГО КОНТУРУ У ВЕБСЕРЕДОВИЩІ	111
4.3. АРХІТЕКТУРА ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕГРОВАНОГО КОНТУРУ ДОВІРИ Й ЦІЛІСНОСТІ.....	121
4.4. ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА МОДЕЛІ ІКДЦ ТА ЗАПРОПОНОВАНИХ МЕТОДІВ...	133
Висновки до розділу 4	145
ВИСНОВКИ.....	147
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	151
ДОДАТОК А	169
ДОДАТОК Б	181

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

2FA	двофакторна автентифікація
ABAC	атрибутно-орієнтований контроль доступу
AI	штучний інтелект
API	інтерфейс програмування застосунків
Autoencoder	автоенкодер; нейромережева модель для виявлення аномалій
BERT	двонаправлені представлення енкодера на основі трансформерів
BFT	візантійська відмовостійкість
BiLSTM	двонаправлена довга короткострокова пам'ять
BlockchainS	модель протидії SQL-ін'єкціям на основі блокчейн-журналювання
QLSecure	та фільтра Блума
Bloom filter	фільтр Блума; імовірнісна структура швидкої перевірки належності елемента множині
CAPTCHA	тест перевірки, що користувач не є ботом
CFLAG	кумулятивний показник рівня доступу
cloud-RDB	хмарна реляційна база даних
CRM	система керування взаємовідносинами з клієнтами
DB	база даних
DCS	трілема децентралізації, узгодженості та масштабованості
DataFrame	таблична структура даних для обробки вибірок
DDoS	розподілена атака на відмову в обслуговуванні
Elliptic Data Set	графовий набір транзакційних даних для аналізу нелегітимних операцій
FLAG	показник рівня доступу до функцій на певному рівні
GCN	графова згорткова мережа
GNN	графова нейронна мережа
ІКДЦ	модель інтегрованого контуру довіри й цілісності

ВСТУП

Актуальність дослідження зумовлена тим, що сучасні вебсистеми функціонують як багатокомпонентні архітектури, що забезпечують роботу сервісів електронної комерції, онлайн-оплат, корпоративних порталів і державних послуг. За таких умов визначальними стають не лише вимоги до захищеності, а й до функціональної стійкості, відмовостійкості, надійності, довіри до результатів обробки запитів, перевірюваності дій та збереження цілісності даних. На практиці порушення цілісності, приховане редагування журналів, компрометація облікових записів, SQL-ін'єкції, DDoS-атаки та масовий вебспам призводять не лише до фінансових втрат і репутаційних ризиків, а й до зниження доступності сервісів, погіршення надійності їх функціонування, втрати функціональної стійкості та ускладнення відновлення коректного стану системи після інцидентів. Разом із цим, традиційні засоби захисту, моніторингу та журналювання мають низку обмежень, вони часто не забезпечують архітектурно узгодженого контролю незмінності критичних подій, не гарантують належної доказовості після інциденту, а також недостатньо підтримують функціональну стійкість і надійність вебзастосунків в умовах еволюції загроз та зростання складності їхньої архітектури. Унаслідок цього вебсистема може залишатися функціональною, але втрачати довіру як до даних, так і до механізмів їх обробки та аудиту.

Важливою науковою задачею є побудова моделей і методів організації архітектури вебзастосунків, які одночасно забезпечують контроль цілісності та простежуваності критичних подій, підвищують доказовість журналів, підтримують функціональну стійкість і надійність сервісів, а також реалізують адаптивне виявлення підозрілої активності у режимі, наближеному до реального часу. Це зумовлює потребу інтеграції архітектурних механізмів забезпечення незмінності (зокрема підходів на основі блокчейну) з інтелектуальними методами аналізу даних (машинне та глибоке навчання, включно з графовими моделями для вебспама та автоенкодерами для аномалій), щоб отримати комплексне рішення, у якому

результати детектування загроз узгоджуються з механізмами контролю цілісності й правилами реагування.

Значних результатів у дослідженні застосування блокчейну для підвищення довіри, контролю цілісності та захисту інформації в розподілених системах досягли Tanrıverdi M., Tekerek A., Zhai Z., Shen S., Mao Y., Prakash R., Anoop V.S., Asharaf S., Awadallah R., Samsudin A., [1, 2, 18, 27] а також інші дослідники, які розглядали блокчейн як основу для виявлення атак, побудови інфраструктури ключів, захисту баз даних у хмарному середовищі та підсилення механізмів кібербезпеки. Питання контролю доступу, делегування прав і формування політик безпеки в системах на основі блокчейну та суміжних підходах висвітлено у працях Hu V.C., Namane S., Ben Dhaou I., Awan S.M. [38-42] та інших авторів, що підтверджує актуальність теми забезпечення довіри й цілісності в цифрових сервісах. Узагальнення сучасних проблем і ризиків кібербезпеки блокчейн-технологій, а також систематизацію наукових підходів подано в оглядових роботах, де показано як сильні сторони блокчейну, так і типові обмеження його практичного застосування для реальних інформаційних систем [45, 46]. Окремий напрям становлять дослідження з тестування й аналізу вразливостей вебзастосунків, які демонструють широкий спектр загроз і підходів до їх оцінювання, але водночас підкреслюють складність побудови універсальних механізмів захисту для вебсередовища [13, 29]. Разом із цим, значний розвиток отримали методи машинного навчання для виявлення атак і підозрілої активності, зокрема підходи до детектування SQL-ін'єкцій і шкідливих запитів, аналізу небезпечних URL та побудови інструментів для виявлення ін'єкційних атак [20, 21, 28]. Для задач фільтрації спаму запропоновано різні машинно-навчальні моделі та алгоритмічні рішення, що підтверджує практичну цінність інтелектуальних методів у модерації та захисті вебканалів взаємодії з користувачами [69-75]. Підходи до побудови систем виявлення вторгнень на рівні вебсервера та огляд сучасних AI-методів у кібербезпеці додатково підкреслюють потенціал адаптивних механізмів захисту для вебсистем [70, 51, 53, 57]. Однак у наведених роботах питання забезпечення довіри й цілісності у вебсистемах здебільшого розглядаються окремо за напрямками, або як блокчейн-архітектура без тісного зв'язку з інтелектуальним

виявленню загроз, або як ML-детектування без інтегрованих механізмів незмінного аудиту та доказовості подій, що визначає актуальність досліджень у напрямі поєднання цих підходів в єдиній інженерній моделі.

Серед українських дослідників окрему увагу привертають праці В. Жебки, у яких розглянуто вибір алгоритмів консенсусу для блокчейн-технологій, побудову інтелектуальних систем виявлення вторгнень із застосуванням методів машинного навчання, адаптивний розподіл криптографічних ресурсів у розподілених базах даних, а також підходи до підвищення функціональної стійкості цифрових платформ і захисту критичної інформаційної інфраструктури в умовах кібератак [136-140]. У працях Н. Лащевської досліджено адаптивне оцінювання ризиків кібербезпеки в розподілених інформаційних системах на основі нейронних мереж, моделі застосування блокчейн-технологій у високонавантажених комп'ютерних системах, а також підходи до виявлення шкідливої активності й вторгнень із використанням нейронних мереж [141-144]. Праці А. Аронова присвячені проблемі вибору алгоритмів консенсусу в розподілених системах і застосуванню інформаційних технологій аналітичного опрацювання текстових джерел як чинника підвищення безпеки прийняття рішень [145, 146]. Окремий інтерес становлять також праці Савчука В. В., у яких розглянуто метод захисту вебзастосунків від завантаження і виконання підозрілих файлів [134], а також праці Фролова Д. І. і Дягілевої М. С., присвячені застосуванню технологій машинного навчання в кібербезпеці та сучасним інноваційним підходам у цій сфері [135]. Наведені роботи підтверджують актуальність розвитку вітчизняних досліджень у напрямі поєднання архітектурних, програмних і інтелектуальних засобів забезпечення довіри, цілісності, стійкості та захищеності вебсистем.

Таким чином, у роботі ставиться актуальне та важливе науково-практичне завдання, суть якого полягає в розробці та обґрунтуванні моделей і методів забезпечення довіри, цілісності та надійності вебсистем на рівні архітектури вебзастосунків шляхом поєднання незмінного журналювання критичних подій із застосуванням методів машинного навчання для автоматизованого виявлення

вебспау та аномалій вебтрафіку, що сприяє підвищенню безпеки, керованості, доказовості й стійкості вебсервісів до комбінованих загроз.

Зв'язок роботи з науковими програмами, планами, темами. Дисертаційна робота виконана відповідно до планів наукової і науково-технічної діяльності Державного університету інформаційно-комунікаційних технологій у межах науково-дослідної роботи «Забезпечення функціональної стійкості інформаційних систем підприємства в умовах впливу дестабілізуючих факторів із застосуванням нейронних мереж» (державний реєстраційний номер 0226U000249) та науково-дослідної роботи «Методи побудови функціонально стійких захищених інформаційних систем з централізованим управлінням» (державний реєстраційний номер 0125U002823).

Мета і задача дослідження.

Метою дослідження є підвищення рівня довіри, цілісності та захищеності архітектури вебсистем, за рахунок обґрунтування та побудови моделей і методів, що забезпечують контроль цілісності даних, визначення критичних подій і адаптивне виявлення підозрілої активності у вебсередовищі на основі технологій блокчейну та машинного навчання.

Для досягнення визначеної мети необхідно розв'язати такі наукові завдання:

1. Проаналізувати існуючі моделі, методи й архітектурні підходи до забезпечення безпеки вебсистем, зокрема в частині забезпечення довіри, контролю цілісності, аудиту критичних подій і виявлення динамічних загроз у вебзастосунках.
2. Удосконалити науково-методичні положення щодо формалізації задачі забезпечення довіри й цілісності у вебсистемах з урахуванням вимог до фіксації критичних подій, контролю їх коректності та виявлення підозрілої активності.
3. Розробити модель інтегрованого контуру довіри й цілісності у вебсистемі для формалізованого подання процесів фіксації, контролю, аудиту, верифікації та аналізу критичних подій у межах архітектури вебзастосунку.
4. Розробити метод блокчейн-верифікованого журналювання критичних подій і контролю доступу у вебсистемах для забезпечення цілісності записів,

виявлення несанкціонованих змін, підвищення доказовості аудиту та підтримки прийняття рішень щодо доступу.

5. Розробити метод графово-нейромережевого виявлення вебспаму та підозрілої активності у вебсистемах для підвищення точності розпізнавання небажаних, підозрілих і потенційно небезпечних дій у змінних умовах функціонування вебзастосунків.

6. Розробити метод інтегрованого забезпечення довіри й цілісності у вебсистемах для узгодженого використання результатів фіксації, контролю, верифікації, аудиту та аналітичного оцінювання критичних подій у процесі прийняття рішень.

7. Реалізувати прототип програмного рішення на основі розроблених моделі та методів і провести експериментальні дослідження його функціонування в типових сценаріях використання вебсистем із оцінюванням точності, повноти, F1-міри, частки хибних спрацьовувань, швидкодії та накладних витрат.

Об'єкт дослідження - процеси забезпечення довіри і цілісності вебсистем.

Предмет дослідження - моделі та методи забезпечення довіри, цілісності та надійності вебсистем на рівні архітектури.

Методи дослідження. У дисертаційній роботі використано теорію графів для подання взаємозв'язків між подіями, об'єктами та рішеннями у вебсистемі; положення теорії довіри до інформаційних систем для формалізації моделі довіри; криптографічні методи хешування та цифрового підпису для забезпечення контролю цілісності; підходи блокчейн-технології та незмінного журналювання для фіксації критичних подій; методи машинного навчання, зокрема графові нейронні мережі, для виявлення вебспаму та аномальної активності; методи математичної статистики та теорії ймовірностей для оцінювання якості моделей; методи теорії планування експерименту для організації та інтерпретації результатів експериментальних досліджень.

Наукова новизна одержаних результатів. У результаті теоретичного аналізу, моделювання та експериментальної перевірки одержані такі нові наукові результати:

- *вперше* розроблено модель інтегрованого контуру довіри й цілісності, у вебсистемі, що ґрунтується на коротко-графовому поданні критичних подій і криптографічних принципах їх верифікації та за рахунок поєднання незмінного журналювання критичних подій, формалізованого подання зв'язків між вебформами, SQL-операціями, рішеннями аналітичного модуля та політиками реагування забезпечує єдине інформаційне середовище для контролю цілісності даних, простежуваності подій, аудиторної перевірки та відтворюваності рішень у вебсистемі;
- *вперше* розроблено метод блокчейн-верифікованого журналювання критичних подій і контролю доступу у вебсистемах, що ґрунтується на розробленій моделі ІКДЦ та теорії криптографічно зв'язаного ланцюга подій із хешуванням, цифровим підписом і пороговим правилом прийняття рішення щодо доступу, який дозволяє зменшити ризик прихованої модифікації інформації, підвищити доказовість журналів і контроль цілісності даних під час розслідування інцидентів;
- *вперше* розроблено метод графово-нейромережевого виявлення вебспаму та підозрілої активності у вебсистемах, що ґрунтується на моделі ІКДЦ та багатопредставленому графовому описі подій, поданих через систему ознак технічного, змістовного, часово-поведінкового, контекстного характеру з урахуванням зв'язків між подіями й результатами аналітичного оцінювання, що забезпечує розрізнення легітимних, підозрілих і шкідливих звернень, підвищує точність виявлення вебспаму та зменшує частку хибних спрацювань;
- *вперше* розроблено метод інтегрованого забезпечення довіри й цілісності у вебсистемах, що ґрунтується на моделі ІКДЦ, методі блокчейн-верифікованого журналювання критичних подій і контролю доступу та методі графово-нейромережевого виявлення вебспаму й підозрілої активності, а також на теорії композиції функціональних відображень критичних подій у клас рішень, що забезпечує цілісність системи, простежуваність та точність прийняття рішень.

Практичне значення наукових результатів полягає у наступному:

1. Розроблені модель і методи дозволяють реалізувати в архітектурі вебзастосунку окрему підсистему довіри та контролю цілісності без повної перебудови існуючої прикладної інфраструктури. Це дозволяє інтегрувати незмінне

журналювання критичних подій, контроль цілісності SQL-операцій, доказовий аудит рішень і засоби інтелектуального виявлення загроз у вже функціонуючі вебсервіси.

2. Практичне значення моделі ІКДЦ та розроблених на її основі методів полягає у поєднанні адаптивного виявлення загроз із доказовим журналюванням рішень, що забезпечує підвищення якості детекції, зниження навантаження на ручну модерацію та можливість незалежного ретроспективного аудиту рішень системи безпеки.

3. Програмні засоби, що реалізують метод виявлення вебспаму та підозрілої активності у вебсистемах, впроваджено у практичну діяльність ТОВ «ШЛІФАРБ», а програмні засоби, що реалізують метод забезпечення довіри й цілісності критичних подій і даних на основі блокчейн-орієнтованого контролю цілісності критичних подій і журналювання транзакційних операцій, - у практичну діяльність ТОВ «АРМА МОТОРС КИЇВ», що підтверджується відповідними актами впровадження. Результати дисертаційного дослідження, що стосуються моделі інтегрованого контуру довіри й цілісності у вебсистемі, методу блокчейн-верифікованого журналювання критичних подій і контролю доступу у вебсистемах, методу графово-нейромережевого виявлення вебспаму та підозрілої активності у вебсистемах і методу інтегрованого забезпечення довіри й цілісності у вебсистемах, також впроваджено в Інституті програмних систем Національної академії наук України при формуванні плану перспективних наукових досліджень, що підтверджується актом впровадження. За результатами впровадження у ТОВ «ШЛІФАРБ» узагальнений показник якості автоматичного розпізнавання повідомлень зріс з 78% до 92%, а частка помилкових спрацювань зменшилася з 2,6% до 0,9%. За результатами впровадження у ТОВ «АРМА МОТОРС КИЇВ» узагальнений показник якості виявлення ризикових ситуацій зріс з 74% до 90%, частка помилкових спрацювань зменшилася з 1,9% до 0,8%, а повнота виявлення атак типу МІТМ на канали обміну транзакційними подіями склала 98%. Контрольне оцінювання результатів впровадження в Інституті програмних систем НАН України показало, що застосування методу інтегрованого забезпечення довіри й цілісності у вебсистемах дозволяє підвищити якість оцінювання критичних подій з 85% до 96%, а частку

помилкових спрацювань зменшити з 3% до 0,8%, що підтверджує ефективність запропонованого підходу для задач забезпечення довіри й цілісності у вебсистемах. Окремі результати дисертаційного дослідження також використано в освітньому процесі Державного університету інформаційно-комунікаційних технологій.

4. У результаті експериментальних досліджень встановлено, що застосування методу інтегрованого забезпечення довіри й цілісності у вебсистемах на основі моделі інтегрованого контуру довіри й цілісності дозволяє зменшити частку хибних спрацювань для класу подій SUBMIT на 65,4% порівняно з базовою конфігурацією на основі правил, що підтверджує практичну доцільність використання запропонованих методів у вебсистемах із підвищеними вимогами до точності виявлення підозрілої активності.

5. Розроблений прототип програмного забезпечення реалізує модель інтегрованого контуру довіри й цілісності та розроблені на її основі методи у вигляді окремого програмного компонента, який може бути інтегрований у наявне вебсередовище без повної зміни бізнес-логіки системи. Практична цінність прототипу полягає в тому, що він може застосовуватися у вебзастосунках як додатковий контур контролю довіри й цілісності без істотної перебудови їхньої архітектури та забезпечує збір і нормалізацію критичних подій, оцінювання їх ризику, кероване реагування й незмінне журналювання результатів. Такий підхід забезпечує практичну основу для побудови підсистем довіри, проведення аудиту, перевірки політик безпеки та подальшого розвитку захисного контуру у вебсистемах.

Особистий внесок здобувача. Основні наукові та прикладні результати дисертаційної роботи, що виносяться на захист, отримані автором особисто. У працях, опублікованих одноосібно, здобувачем запропоновано підхід до підвищення безпеки вебсайтів на основі інтеграції блокчейну, коефіцієнта Джині та кривої Лоренца, розроблено алгоритм і проведено його тестування [78]; розроблено інтегрований контур довіри у вебзастосунках на основі графового оцінювання ризику, незмінного журналювання рішень і перевірки цілісності критичних подій [89]. У роботах, опублікованих у співавторстві, здобувачем особисто: проведено порівняльний аналіз блокчейн-технологій і традиційних методів безпеки вебсайтів,

досліджено вплив кількості блоків та якості шифрування на точність рішень і швидкість обробки запитів, розроблено комплексну формулу оцінки надійності системи [79]; визначено критерії оцінювання ризикованості фінансових транзакцій, розроблено UML-схему класів програмної бібліотеки, алгоритм роботи моделі, псевдокод, методи генерації тестових даних і проведено тестування моделі на фінансових даних [80]; досліджено сценарії MITM-атак у багатоканальних платіжних системах, розроблено багаторівневу архітектуру інтегрованої системи безпеки та обґрунтовано поєднання методів штучного інтелекту, цифрових підписів, часових міток і додаткової клієнтської верифікації [91]; розроблено модель автоматизованої оцінки безпеки вебзастосунків, реалізовано порівняння результатів сканування з еталонним набором вразливостей, класифікацію TP, FP, FN, TN, розрахунок метрик Precision, Recall і Accuracy та архітектуру Python-мультисканерної платформи [92]. У публікаціях, що індексуються в міжнародних наукометричних базах, здобувачем розроблено алгоритм роботи прототипу виявлення подібних профілів у соціальних мережах із використанням методів кластерного аналізу та проведено його тестування [75]; розроблено концепцію BlockchainSQLSecure, сформовано архітектуру блокчейн-журналу SQL-запитів, запропоновано механізм валідації SQL-запитів через smart-контракти та підхід до децентралізованого зберігання журналів [77]; розроблено метод виявлення DDoS-атак у вебтрафіку на основі автоенкодерів, запропоновано трирівневу схему класифікації запитів і механізм донавчання моделі з урахуванням реальних даних та експертної оцінки [81]; розроблено мультирепрезентаційну графову модель виявлення вебспаму, сформовано три подання подій, запропоновано механізми узгодження представлень, адаптивного злиття ознак і контрастивного навчання, проведено експериментальне оцінювання моделі [135]. У публікаціях, що індексуються в міжнародних наукометричних базах, здобувачем розроблено алгоритм роботи прототипу виявлення подібних профілів у соціальних мережах із використанням методів кластерного аналізу та проведено його тестування [75]. Розроблено концепцію BlockchainSQLSecure, сформовано архітектуру блокчейн-журналу SQL-запитів, запропоновано механізм валідації SQL-запитів через smart-контракти та підхід до децентралізованого зберігання

журналів [77]. Також розроблено метод виявлення DDoS-атак у вебтрафіку на основі автоенкодерів, запропоновано трирівневу схему класифікації запитів і механізм донавчання моделі з урахуванням реальних даних та експертної оцінки [81]. У частині застосування графових моделей машинного навчання розроблено мультирепрезентаційну графову модель виявлення вебспаму, сформовано три подання подій, запропоновано механізми узгодження представлень, адаптивного злиття ознак і контрастивного навчання, проведено експериментальне оцінювання моделі [135].

Усі результати, представлені в роботі, є продуктом самостійної наукової діяльності здобувача; використані джерела належно задокументовано, а запозичення відсутні або мають коректні посилання відповідно до вимог академічної доброчесності.

Апробація результатів дисертації. Основні положення, теоретичні підходи, методи та практичні результати дисертаційного дослідження доповідалися, обговорювалися та пройшли апробацію на міжнародних і всеукраїнських наукових конференціях, зокрема:

1. *X International Conference “Information Technology and Implementation” (IT&I-2023), 20-21 листопада 2023 року, м.Київ;*
2. *IV Міжнародна науково-практична конференція молодих вчених та студентів «Інженерія програмного забезпечення і передові інформаційні технології (SoftTech-2023)», присвячена 125-й річниці КІІ ім. Ігоря Сікорського, 19-21 грудня 2023 року, м.Київ;*
3. *Всеукраїнська науково-технічна конференція «Застосування програмного забезпечення в інформаційно-комунікаційних технологіях», 24 квітня 2024 року, Київ;*
4. *XIII Міжнародна науково-технічна конференція «Безпека інформаційних технологій: ITSEC-2024», 9-11 травня 2024 року, Львів;*
5. *XIII Міжнародна науково-практична конференція «Математика. Інформаційні технології. Освіта», 31 травня - 2 червня 2024 року, Луцьк-Світязь;*

6. *5th IEEE International Conference on Advanced Trends in Information Theory, 21-23 November, 2024 року, Lviv;*
7. *Науково-практична конференція «Штучний інтелект і безпека» 19- 21 листопада 2024 року, Київ;*
8. *Всеукраїнська науково-технічна конференція «Виклики та рішення в програмній інженерії», 26 листопада 2024 року, Київ;*
9. *XIV Міжнародної науково-практичної конференції «Математика. Інформаційні технології. Освіта», 13-15 червня 2025 року, Луцьк - Світязь;*
10. *XIV Міжнародна науково-технічна конференція «Безпека інформаційних технологій: ITSEC-2025» 22-24 травня 2025 року, Тернопіль;*
11. *SMICS-2025 «Безпека сучасних інформаційно-комунікаційних систем» 16-18 жовтня 2025 року, Львів;*
12. *II Всеукраїнська науково-технічна конференція «Виклики та рішення в програмній інженерії», 26 листопада 2025 року, Київ.*

Публікації. За результатами дисертаційного дослідження опубліковано 19 наукових праць [75-83; 85; 88; 89; 91-94; 135; 149; 150], серед яких 4 публікації, що індексуються у наукометричній базі Scopus [75; 77; 81; 135], 6 статей у фахових наукових виданнях категорії Б [78; 79; 80; 89; 91; 92], а також 9 публікацій у збірниках матеріалів і тез міжнародних та всеукраїнських науково-технічних та науково-практичних конференцій [76; 82; 83; 85; 88; 93; 94; 149; 150].

Структура та обсяг дисертації. Дисертація включає вступ, чотири розділи, висновки, список використаних джерел із 151 найменування. Загальний обсяг роботи становить 187 сторінки, серед яких 135 сторінки основного тексту, 25 рисунків, 7 таблиць.

РОЗДІЛ 1. АНАЛІЗ СУЧАСНОГО СТАНУ ТА ПРОБЛЕМАТИКИ ЗАБЕЗПЕЧЕННЯ ДОВІРИ І ЦІЛІСНОСТІ У ВЕБСИСТЕМАХ

Сучасні вебсистеми (зокрема сервіси онлайн-оплат, кабінети користувачів, форми зворотного зв'язку та швидкого замовлення) працюють із критично важливими даними й бізнес-процесами, питання довіри, цілісності та контрольованості змін є базовими вимогами до їх проєктування й експлуатації. У практичному середовищі вебзастосунки піддаються як класичним атакам на рівні запитів і доступу (наприклад, SQL-ін'єкції та несанкціоновані дії), так і атакам на рівні сервісу та комунікацій (DDoS, автоматизована шкідлива активність), а також зловживанням через канали взаємодії з користувачами (спам/боти у вебформах). Проблематика забезпечення довіри, цілісності та безпеки вебзастосунків системно відображена в оглядових роботах і систематичних оглядах літератури, де підкреслюється різноманітність векторів атак та потреба в комплексних механізмах контролю і виявлення інцидентів [13, 26]. Ключовий виклик полягає в тому, що традиційні захисні механізми (правила фільтрації, сигнатури, локальні журнали подій) не завжди забезпечують достатню прозорість і доказовість, журнали можуть бути змінені, події - приховані, а аналіз інцидентів ускладнюється відсутністю надійної істини. У цьому контексті перспективним напрямом є застосування блокчейну та смарт-контрактів як основи для незмінного журналювання та аудиту дій у вебсистемі, включно з фіксацією подій доступу й критичних запитів. У наукових джерелах блокчейн розглядається як технологічна база для підвищення цілісності даних і побудови захищених сервісів (у т.ч. інфраструктурних компонентів на кшталт інфраструктури відкритих ключів (PKI)), а також для моделей контролю доступу [1], [2], [24], [35], [36], [37], [42], [43]. Водночас практична реалізація таких підходів потребує врахування питань продуктивності, зберігання та обробки запитів у блокчейн-системах, а також оптимізацій на кшталт швидких структур для перевірок (наприклад, Bloom filter), ефективність яких окремо досліджується у [32], [33], [34]. Практична цінність блокчейн-підходів у кібербезпеці пов'язана не лише з незмінністю записів, а й із децентралізацією, яка зменшує ризик централізованих

точок компрометації та ускладнює приховане втручання в історію подій. Водночас під час інтеграції блокчейну у вебсистеми критичним є баланс між продуктивністю, масштабованістю та приватністю, оскільки надмірна деталізація подій у реєстрі може збільшувати накладні витрати, а надмірне приховування - знижувати доказовість аудиту [1], [2].

Другим важливим напрямом, що доповнює механізми фіксації та аудиту, є виявлення атак і аномалій на основі машинного навчання. Для задач детекції SQL-ін'єкцій та ін'єкційних/шкідливих патернів у вебтрафіку застосовуються нейромережеві та ML-підходи, що описані у дослідженнях і систематичних оглядах [20], [21], [25]. Для протидії DDoS-атакам і підозрілій активності в мережевому трафіку розглядаються методи інтелектуальної детекції, а також підходи на основі автоенкодерів для виявлення аномалій [49], [50], [81]. Окрему групу практичних задач становить фільтрація вебспаму у формах взаємодії з користувачами; у сучасних дослідженнях для цього активно застосовуються графові нейронні мережі та багаторепрезентаційні моделі, що підвищують точність класифікації в умовах складних залежностей між ознаками та взаємодіями [95], [96], [97].

У цьому розділі виконано аналіз проблеми забезпечення довіри й цілісності у вебсистемах та узагальнено підходи, які формують основу подальшого дослідження принципи побудови безпечних вебсистем і підходи до оцінювання рівня їх безпеки, блокчейн-орієнтовані рішення для фіксації подій, перевірки цілісності та аудиту змін, методи машинного навчання для виявлення атак, аномалій і підозрілої активності у вебсередовищі.

1.1. Обґрунтування вибору теми та аналіз проблеми забезпечення довіри й цілісності у вебсистемах

Вебсистеми стали базовою інфраструктурою для щоденних процесів у бізнесі, освіті, медицині та державних сервісах, вони обробляють персональні дані, виконують фінансові операції та підтримують критичні бізнес-процеси. У таких умовах питання довіри до результатів роботи вебсистем і збереження цілісності даних

виходить за межі суто технічного аспекту та перетворюється на умову стабільності сервісу й репутації організації. Зі зростанням функціональності вебзастосунків зростає і поверхня атаки, а загрози безпосередньо впливають на конфіденційність, цілісність і доступність інформації, що вимагає системного підходу до безпеки та оцінювання її рівня [13]. Проблема довіри й цілісності у вебсистемах проявляється не лише як ризик витоку даних, а і як ризик непомітної підміни, спотворення або некоректної інтерпретації подій у системі. Користувачі та адміністратори очікують, що кожна дія буде коректно виконана, зафіксована та відтворювана під час перевірки. На практиці ж вебсистеми складаються з багатьох компонентів, зовнішніх інтеграцій і API, а дані проходять кілька рівнів обробки. Це ускладнює контроль, підвищує імовірність помилок і створює точки втручання, де зловмисник може змінювати стан системи так, щоб зміни виглядали легітимними. У підсумку зростає потреба в механізмах, які дозволяють не тільки запобігати інцидентам, а й доводити коректність подій та незмінність критичних даних у ретроспективі [42], [43].

Окремий блок проблем формують атаки, що підривають довіру через деградацію сервісу та спотворення інформаційних потоків. Типовим прикладом є вебспам через форми зворотного зв'язку та інші канали введення даних: масові повідомлення перевантажують інфраструктуру, створюють непотрібну інформацію у даних, погіршують якість взаємодії з клієнтами та можуть містити шкідливі посилання або сценарії. Спам забирає час на модерацію і часто приховує важливі звернення реальних користувачів. При цьому спам еволюціонує та маскується під нормальний контент, змінює структуру текстів і поведінку запитів, прості фільтри за ключовими словами, CAPTCHA чи чорні списки поступово втрачають ефективність. Актуальність цієї проблеми підтверджується і кількісною динамікою автоматизованих атак, за даними 2025 Imperva Bad Bot Report, у 2024 р. шкідливі боти вже становили 37% усього інтернет-трафіку проти 32% у 2023 р., а частка простих високочастотних бот-атак зросла з 40% до 45%, що свідчить про зниження порога входу для масового запуску таких атак [147]. Додатково, згідно з DataDome Global Bot Security Report 2025, у 2025 р. 64% AI-бот-трафіку припадало на форми, 23% - на сторінки входу, а 5% - на сторінки оформлення замовлення, що підтверджує

концентрацію автоматизованого навантаження саме на критичних точках взаємодії вебсистеми [148]. У такій ситуації обґрунтованим є застосування методів машинного навчання для фільтрації спаму, зокрема підходів, що враховують множину ознак і стійкіші представлення контенту та поведінки [66], [107]. Додаткову цінність мають моделі, здатні аналізувати не лише текст, а й зв'язки між подіями й об'єктами вебсистеми; до цього класу належать графові підходи, зокрема рішення на основі графових нейронних мереж, які переносять задачу класифікації у простір графових представлень взаємодій [95], а також прикладні дослідження застосування графової нейронної мережі (GNN) для автоматизації вебспам-фільтрації [84].

Одним із показових формалізованих підходів до задачі спам-фільтрації є подання її як задачі класифікації повідомлень у просторі ознак із подальшим переходом до графового подання даних [95]. У межах такого формалізованого підходу кожне повідомлення описується як елемент простору електронних листів $e \in E$, а множина можливих міток задається як $C = \{c_1, c_2, \dots, c_p\}$, де для задачі спам-фільтрації зазвичай розглядаються два класи - легітимне повідомлення та спам. Навчальна вибірка формалізується як множина розмічених прикладів $T \subset E \times C$, а метою навчання є побудова класифікаційної функції:

$$\sigma: E \rightarrow C, \quad (1.1)$$

яка відображає повідомлення у відповідний клас [95]. Такий підхід задає строгий математичний базис для переходу від традиційної текстової класифікації до графової моделі, у якій спам розглядається не лише через вміст окремого повідомлення, а і через семантичні та структурні зв'язки між об'єктами. Це особливо цінно для вебсистем, де підозріла активність часто проявляється не ізольовано, а у вигляді повторюваних патернів взаємодій між повідомленнями, джерелами, атрибутами запитів і поведінковими ознаками [95].

Не менш критичними є атаки, які безпосередньо порушують цілісність даних або механізми їх обробки, таких як SQL-ін'єкції, перехоплення чи підміна даних, неправомірний доступ до ресурсів, а також DDoS-атаки, що знижують доступність сервісу і створюють умови для вторинних інцидентів. Такі сценарії здатні спричиняти прямі фінансові втрати, витік конфіденційної інформації та довгострокові

репутаційні наслідки; у результаті порушується базовий принцип довіри - користувачі перестають бути впевненими, що система коректно зберігає і обробляє їхні дані. Для SQL-ін'єкцій це проявляється як ризик модифікації або викрадення даних на рівні бази даних (БД), актуальними є як методи виявлення атак на основі моделей (зокрема нейромережових), так і інженерні підходи контролю запитів [20]. Для DDoS-атак загроза полягає у втраті доступності та деградації сервісу, а також у появі “вікон” для супутніх атак, що підсилює потребу у методах детекції підозрілої активності у трафіку, зокрема із застосуванням автоенкодерів та адаптивних схем аналізу [81]. Традиційні підходи до захисту вебзастосунків здебільшого або зосереджені на периметрі, або орієнтовані на відомі шаблони атак. Вони є необхідними, але часто не дають відповіді на запитання “як довести, що дані не були змінені” та “як відстежити і підтвердити коректність дій у системі” в умовах складної архітектури та багаторівневих інтеграцій. В прикладних сценаріях необхідні механізми, які одночасно забезпечують контроль доступу, виявлення аномалій, а також надійний облік подій, щоб інцидент можна було дослідити, а результати перевірити й відтворити [13].

Доцільність поєднання двох напрямів обґрунтовується підсиленням один одного, перший напрям пов'язаний із застосуванням блокчейн-підходів як засобу підвищення довіри через незмінність записів, прозорість фіксації подій та стійкість журналів до підробки. Властивості розподіленого журналювання ускладнюють приховане редагування записів і посилюють аудит та перевірку подій у системі [42], [43]. Практична спрямованість такого підходу підтверджується роботами, де блокчейн застосовується для задач виявлення вебатак у розподіленому середовищі [1], а також для підсилення безпеки реляційних БД у хмарних сценаріях [24], запропоновано архітектурну модель BC over cloud-RDB для підвищення захисту реляційних баз даних у хмарному середовищі, у якій цілісність модифікацій забезпечується завдяки розподілу обробки між кількома хмарними провайдерами, зв'язуванню записів через SHA-256 та механізму клієнтської самоперевірки за узгодженими RDB-signature. Автори розглядають дві реалізації цієї моделі - agile BC-based RDB для сценаріїв із високою пропускнуою здатністю та secure BC-based RDB

для сценаріїв із підвищеними вимогами до захисту чутливих даних; обидві спираються на Byzantine Fault Tolerance, а захищений варіант додатково використовує механізм підтвердження виконаної роботи (Proof-of-Work, PoW). Формально складність PoW задається співвідношенням:

$$D = \frac{T_{max}}{T_c}, \quad (1.2)$$

де T_{max} - максимально допустиме значення цільового порога хешу, що відповідає мінімальній складності, а T_c - поточне значення цільового порога хешу, що дозволяє регулювати обчислювальну складність підтвердження нового запису та, відповідно, компроміс між продуктивністю і стійкістю до несанкціонованої модифікації даних. Для тематики цієї дисертації дана модель є важливою як приклад літературного підходу до побудови блокчейн-орієнтованої підсистеми контролю цілісності на рівні сховища даних, водночас її обмеженням є те, що вона зосереджена переважно на хмарній реляційній БД і не охоплює повною мірою задачі моделювання довіри та адаптивного виявлення загроз у вебзастосунках. Окремо виділяється напрям інтеграції блокчейну для підвищення стійкості вебзастосунків до SQL-ін'єкцій через фіксацію критичних запитів і операцій у незмінному журналі та підтримку процедур перевірки коректності дій [77]. Другий напрям пов'язаний із машинним навчанням як основою адаптивного захисту там, де правила і статичні фільтри швидко застарівають. Для вебспаму це означає здатність системи відокремлювати легітимні звернення від підозрілих за сукупністю ознак (структура повідомлення, повторюваність елементів, час надходження, частота запитів, параметри джерела тощо), зменшуючи потребу у постійному ручному коригуванні фільтрів [66], [107]. Для більшої стійкості до модифікацій спаму перспективними є підходи, що враховують зв'язки у даних і поведінкові взаємодії, зокрема графові моделі та GNN, які розглядають спам як прояв структурних патернів у мережі подій [95], [84].

Актуальність теми визначається потребою вебсистем у механізмах, які забезпечують контрольованість критичних дій, збереження цілісності даних і доказовість подій, а також здатність адаптивно виявляти мінливі загрози (вебспам, аномальний трафік, SQL-ін'єкції, DDoS-атак). Теоретичне підґрунтя такого поєднання полягає в тому, що блокчейн-підходи підсилюють доказовість та

незмінність обліку подій [42], [43], [24], [77], тоді як машинне навчання забезпечує адаптивність виявлення атак і підозрілої активності в умовах еволюції загроз [66], [107], [95], [81]. Поєднання блокчейну з інтелектуальними методами аналізу даних формує перевірюваний контур фіксації подій, блокчейн відповідає за незмінність і верифікованість записів, тоді як методи штучного інтелекту забезпечують виявлення складних і мінливих патернів загроз у потоках запитів та повідомлень. Таке поєднання розглядається як перспективна основа для нових класів безпекових рішень, здатних одночасно підтримувати доказовість інцидентів та адаптивність детекції [3], [14], [46].

1.2. Принципи побудови безпечних вебсистем і підходи до оцінювання рівня їх безпеки

Побудова безпечних вебсистем ґрунтується на поєднанні базових властивостей інформаційної безпеки - конфіденційності, цілісності та доступності - із вимогами до довіри, тобто перевірюваності дій, підзвітності та відтворюваності подій у разі аудиту або розслідування інцидентів [13]. У вебзастосунках це означає, що безпека має закладатися на рівні архітектури, життєвого циклу розробки та експлуатаційних процесів, а не реалізовуватися як набір точкових правок під відомі загрози [13], [66]. Ключовим принципом є *security-by-design* - проектування з урахуванням загроз і меж довіри між компонентами. Вебсистема має бути побудована так, щоб мінімізувати поверхню атаки, ізолювати критичні підсистеми та забезпечувати багаторівневий захист (*defense-in-depth*) на рівні мережі, застосунку, бази даних і керування доступом [13]. У практичній реалізації це зводиться до вимог найменших привілеїв і розподілу ролей, коректної автентифікації та авторизації, захисту сесій, контролю введення й нормалізації даних на межах довіри (форми, API, інтеграції), а також криптографічного захисту даних під час передавання і зберігання [13]. Такий підхід знижує ймовірність типових інцидентів, пов'язаних із неправомірним доступом і витоком даних, та обмежує наслідки у разі компрометації окремого компонента [13]. У сучасних архітектурах безпеки додатково посилюється принцип "нульової довіри"

(Zero Trust), коли жоден компонент або суб'єкт не вважається довіреним за замовчуванням, а доступ визначається політиками з урахуванням атрибутів і контексту (ABAC). У роботах, орієнтованих на IoT- та розподілені середовища, показано, що блокчейн може виступати технічною основою для незмінного журналу рішень доступу й узгодження політик, а розподілене сховище - підтримувати доступність даних при часткових відмовах [40]. У вебсистемах ці ідеї доцільно трактувати як підсилення керованості доступу до критичних API та подій, що формують контур довіри.

Для опису проблеми забезпечення довіри й цілісності у вебсистемах доцільно розглядати три взаємопов'язані рівні: автентифікацію та контроль доступу, оцінювання ймовірності порушення базових властивостей безпеки, а також адаптивний криптографічний захист інфраструктурних компонентів. На рівні автентифікації базова модель входу користувача до окремого сервісу задається формулою [139]:

$$L_i = F_{S_i}(ID_i, AU_i), \quad (1.3)$$

де L_i - результат входу користувача до i -го сервісу, F_{S_i} - функція перевірки автентифікаційних даних у сервісі S_i , S_i - окремий сервіс системи, ID_i - ідентифікатор користувача в сервісі S_i , AU_i - автентифікаційний секрет, наприклад пароль, токен або інший засіб підтвердження особи. Для захисту автентифікаційного секрету використовується хешування, що описується формулою [139]:

$$L_i = F_{S_i}(ID_i, \mathcal{H}(AU_i)), \quad (1.4)$$

де $\mathcal{H}(AU_i)$ - хешоване представлення секрету, $\mathcal{H}(\)$ - хеш-функція, AU_i - автентифікаційний секрет користувача. У вебсистемах така формула описує принцип зберігання не відкритого пароля, а його криптографічного представлення. Для об'єднаної інформаційної системи автентифікація може виконуватися через єдиний захисний шлюз, що подано формулою [139]:

$$L|_{VS} = F(ID, \mathcal{H}(AU)), \quad S = [S_1, S_2, \dots, S_n], \quad n \in \mathbb{Z}, \quad (1.5)$$

де $L|_{VS}$ - результат автентифікації, що поширюється на всю множину сервісів, F - загальна функція перевірки автентифікаційних даних, ID - ідентифікатор користувача, AU - автентифікаційний секрет, S - множина сервісів, S_1, S_2, \dots, S_n -

окремі сервіси, n - кількість сервісів. Така модель відповідає централізованій автентифікації. Для реалізації принципу Zero Trust модель доповнюється ролями доступу, як показано у формулі [139]:

$$L|_{\forall S} = F(ID, \mathcal{H}(AU)), S = [(S_1, R_1), (S_2, R_2), \dots, (S_n, R_n)], n \in \mathbb{Z}, \quad (1.6)$$

де R_i - роль користувача в сервісі S_i , (S_i, R_i) - пара “сервіс - роль”, S - множина сервісів із визначеними ролями. У вебзастосунках це відповідає рольовому розмежуванню доступу, принципу мінімально необхідних привілеїв і перевірці прав користувача не лише під час входу, а й під час виконання кожної критичної дії. У разі використання кількох ідентифікаторів користувача модель набуває вигляду, поданого у формулі [139]:

$$L|_{\forall S} = F(ID, \mathcal{H}(AU)), ID = [ID_1, ID_2, \dots, ID_m], n, m \in \mathbb{Z}, \quad (1.7)$$

де ID - множина ідентифікаторів користувача, $(ID_1, ID_2, \dots, ID_m)$ - окремі ідентифікатори, наприклад логін, email, номер телефону або зовнішній ідентифікатор, m - кількість ідентифікаторів. Якщо використовується кілька автентифікаційних секретів, формула узагальнюється так, як показано у формулі [139]:

$$L|_{\forall S} = F(ID, \mathcal{H}(AU)), AU = [AU_1, AU_2, \dots, AU_k], n, m, k \in \mathbb{Z}, \quad (1.8)$$

де AU - множина автентифікаційних секретів, AU_1, AU_2, \dots, AU_k - окремі секрети, наприклад пароль, одноразовий код, біометричний фактор, апаратний ключ або токен, k - кількість секретів. Така формалізація відповідає багатофакторній автентифікації та гнучкій ідентифікації користувача у вебсистемах. Оцінювання рівня безпеки може виконуватися через імовірність порушення функціональних властивостей безпеки. У джерелі [139] наведено формалізацію, подану у формулі:

$$P = 1 - \sum_{i=1}^4 (1 - q_i), \quad (1.9)$$

де P - узагальнена ймовірність порушення функціональних властивостей безпеки, q_1 - ймовірність порушення конфіденційності, q_2 - ймовірність порушення цілісності даних, q_3 - ймовірність порушення автентифікаційної безпеки, q_4 - ймовірність порушення невідмовності або неспростовності дій, i - індекс властивості безпеки. Для вебзастосунків ці складові можна співвіднести з ризиками несанкціонованого

читання даних, підміни або видалення інформації, компрометації автентифікації та неможливості довести факт виконання користувачем певної дії.

З урахуванням класичного ймовірнісного підходу для оцінювання ймовірності настання хоча б одного порушення доцільною є також добуткова форма, наведена у формулі:

$$P_{web} = 1 - \prod_{i=1}^r (1 - q_i), \quad (1.10)$$

де P_{web} - інтегральна ймовірність порушення безпеки вебзастосунку, q_i - ймовірність реалізації i -ї загрози або порушення окремої властивості безпеки, r - кількість врахованих загроз, \prod - добуток ймовірностей ненастання окремих порушень, $(1 - q_i)$ - ймовірність того, що відповідне порушення не відбулося. У межах вебзастосунку до таких q_i можуть належати ризики порушення конфіденційності, цілісності, автентифікації, контролю доступу, захисту сесій, безпеки API, а також реалізації XSS, SQL Injection, CSRF, IDOR, SSRF та інших атак. Таку формулу доцільно застосовувати як спрощену інтегральну оцінку за умови незалежності або умовної незалежності подій.

Інфраструктурний рівень безпеки може бути описаний через децентралізацію та гібридизацію каналів передавання даних. Комунікаційний канал у гібридній мережі подається формулою [139]:

$$e_j = (i_f, i_t, i_\varepsilon, i_m), \quad i_f = \overline{1, n_f}, \quad i_t = \overline{1, n_t}, \quad i_\varepsilon = \overline{1, n_\varepsilon}, \quad i_m = \overline{1, n_m}, \quad j = \overline{1, J}, \quad (1.11)$$

де e_j - j -й комунікаційний канал, j - ідентифікатор каналу, i_f - ідентифікатор частотного розділення, i_t - ідентифікатор часового розділення, i_ε - ідентифікатор фізичної природи сигналу, i_m - ідентифікатор середовища передавання, $n_f, n_t, n_\varepsilon, n_m$ - кількість каналів відповідних типів, J - загальна кількість каналів. Для вебсистем ця формула має інфраструктурне значення: вона може використовуватися для опису резервування каналів, стійкості до відмов і зменшення залежності від одного каналу передавання даних. Залежність параметрів сигналу від фізичної природи сигналу та середовища передавання формалізується за допомогою співвідношень [139]:

$$X[e_j = (i_f, i_t, i_\varepsilon, i_m)] \neq X[e_j = (i_f, i_t, i_{\varepsilon'}, i_m)],$$

$$X[e_j = (i_f, i_t, i_\varepsilon, i_m)] \neq X[e_j = (i_f, i_t, i_\varepsilon, i_{m'})], \quad (1.12)$$

де X - параметр сигналу, наприклад загасання, дальність або максимальна швидкість передавання, $\varepsilon, \varepsilon'$ - сигнали різної фізичної природи, m, m' - різні середовища передавання. Зміст цих формул полягає в тому, що зміна фізичної природи сигналу або середовища передавання змінює характеристики каналу. Для вебсистем це можна трактувати як принцип резервування й інфраструктурної стійкості, але не як модель прикладної безпеки вебзастосунку.

Криптографічний рівень захисту розподіленої вебсистеми може бути описаний через адаптивний розподіл криптографічних ресурсів. Вхідними параметрами такої моделі є кількість вузлів N , навантаження кожного вузла L_i , рівень загрози для вузла S_i , максимальна кількість доступних ключів K_{max} , загальна кількість ключів K_{total} , а також порогове значення загрози θ [138]. Для приведення навантаження та рівня загрози до єдиного діапазону використовуються формули нормалізації [138]:

$$L_i^{norm} = \frac{L_i}{\max(L)}, \quad S_i^{norm} = \frac{S_i}{\max(S)}, \quad (1.13)$$

де L_i^{norm} - нормалізоване навантаження i -го вузла, L_i - фактичне навантаження вузла, $\max(L)$ - максимальне навантаження серед усіх вузлів, S_i^{norm} - нормалізований рівень загрози, S_i - фактичний рівень загрози, $\max(S)$ - максимальне значення рівня загрози в системі. Пріоритетність вузла визначається ваговим коефіцієнтом за формулою:

$$W_i = \alpha L_i^{norm} + \beta S_i^{norm}, \quad (1.14)$$

де W_i - ваговий коефіцієнт i -го вузла, α - коефіцієнт впливу навантаження, β - коефіцієнт впливу рівня загрози, L_i^{norm} - нормалізоване навантаження, S_i^{norm} - нормалізований рівень загрози. У вебсистемі W_i можна інтерпретувати як показник критичності окремого компонента: API-шлюзу, сервера авторизації, бази даних, платіжного модуля або журналу критичних подій. Якщо рівень загрози перевищує поріг, тобто $S_i > \theta$, виконується аварійне оновлення ключів і рівномірний розподіл криптографічних ресурсів за формулою:

$$K_i = \frac{K_{total}}{N}, \quad (1.15)$$

де K_i - кількість ключів, призначених i -му вузлу, K_{total} - загальна кількість ключів, які потрібно розподілити, N - кількість вузлів. У нормальному режимі, коли $S_i \leq \theta$, ключі розподіляються пропорційно до вагових коефіцієнтів за формулою:

$$K_i = \frac{K_{max} \cdot W_i}{\sum_{j=1}^N W_j}, \quad (1.16)$$

де K_{max} - максимальна кількість доступних криптографічних ключів, W_i - вага i -го вузла, W_j - вага j -го вузла, $\sum_{j=1}^N W_j$ - сума ваг усіх вузлів. Для контролю дисбалансу навантаження між вузлами використовується функція варіації, подана у формулі:

$$V(t) = \sum_{i=1}^N \left(L_i(t) - L_{avg}(t) \right)^2, \quad L_{avg}(t) = \frac{1}{N} \sum_{i=1}^N L_i(t), \quad (1.17)$$

де $V(t)$ - показник нерівномірності навантаження в момент часу t , $L_i(t)$ - навантаження i -го вузла, $L_{avg}(t)$ - середнє навантаження системи, N - кількість вузлів. Перерозподіл криптографічного навантаження між вузлами задається рівнянням міграції:

$$M_{ij} = \gamma(L_i - L_j), \forall i, j \in N, L_i > L_j, \quad (1.18)$$

де M_{ij} - обсяг міграції ключів або криптографічного навантаження від вузла i до вузла j , γ - коефіцієнт адаптації, L_i - навантаження вузла-джерела, L_j - навантаження вузла-приймача. Ефективність криптографічного захисту оцінюється через час доступу до зашифрованих даних за формулою:

$$T_{access} = T_{proc} + T_{comm}, \quad (1.19)$$

де T_{access} - загальний час доступу до зашифрованих даних, T_{proc} - час обробки запиту на сервері, T_{comm} - затримка, пов'язана з передаванням криптографічних ключів або службової криптографічної інформації. Для виявлення перевантаженого вузла використовується умова [138]:

$$L_i > \theta \cdot \frac{\sum_{j=1}^N L_j}{N}, \quad (1.20)$$

де L_i - навантаження i -го вузла, θ - коефіцієнт дисбалансу навантаження, $\sum_{j=1}^N L_j$ - сумарне навантаження всіх вузлів, $\frac{\sum_{j=1}^N L_j}{N}$ - середнє навантаження системи. Якщо

умова (1.20) виконується, частина запитів або криптографічних операцій переноситься на інший вузол відповідно до формули [138]:

$$L_j = L_j + \alpha(L_i - L_{avg}), L_i = L_i - \alpha(L_i - L_{avg}), \quad (1.21)$$

де L_j - навантаження вузла, який приймає частину операцій, L_i - навантаження перевантаженого вузла, α - коефіцієнт розподілу, L_{avg} - середнє навантаження, Оновлення ключів може також виконуватися за умови зниження продуктивності вузла, що подано у формулі [138]:

$$K_i \rightarrow K_j, \quad \text{if } C_i < \beta \cdot C_{avg}, \quad (1.22)$$

де $K_i \rightarrow K_j$ - передавання або перерозподіл криптографічних ключів від вузла i до вузла j , C_i - продуктивність i -го вузла, C_{avg} - середня продуктивність вузлів, β - пороговий коефіцієнт для оновлення або перерозподілу ключів. У вебсистемах це доцільно застосовувати для компонентів, що виконують криптографічно значущі операції: автентифікацію, авторизацію, обробку токенів, перевірку цифрових підписів, журналювання критичних подій і доступ до зашифрованих даних.

Окрему групу принципів формує забезпечення цілісності даних і подій. Для вебсистем важливо не лише запобігти модифікації даних, а й гарантувати можливість доведення коректності виконаних операцій. На рівні застосунку це підтримується транзакційністю, валідацією бізнес-правил і контрольованими змінами стану, а на рівні спостережуваності - повним і несуперечливим журналюванням критичних дій. Перспективним підсиленням журналювання є використання незмінних механізмів аудиту, зокрема підходів на основі блокчейну, де властивості незмінності записів і прозорості фіксації подій ускладнюють приховане редагування логів після інциденту та підвищують доказовість історії операцій [42], [43]. У прикладних сценаріях це може поєднуватися з перевітками на рівні смарт-контрактів або процедур контролю запитів, що розглядається як додатковий шар захисту для критичних операцій доступу та взаємодії з БД, зокрема у контексті протидії SQL-ін'єкціям [77], [24].

Оскільки значна частина загроз для вебсистем має динамічний характер, принципи безпечної побудови доповнюються вимогою до адаптивності. Це стосується як деградаційних атак на доступність (наприклад, DDoS), так і атак, що спотворюють інформаційні потоки (вебспам, автоматизовані зловмисні запити), де

статичні правила швидко втрачають ефективність [81], [66]. У таких умовах доцільним є використання аналітичних механізмів і методів машинного навчання для виявлення аномалій, підозрілої поведінки та нових шаблонів атак, що дозволяє зменшити залежність від ручного налаштування захисту та підвищити оперативність реагування [66], [107]. Прикладом є інтелектуальні підходи до детекції спаму й вторгнень у вебсередовищі, де демонструється потенціал високої точності виявлення за умови коректної постановки задачі та валідації на даних експлуатації [78], [81]. Оцінювання рівня безпеки вебсистем має будуватися як безперервний процес, що охоплює як етапи розробки, так і етапи впровадження та експлуатації. На рівні інженерних практик основою є моделювання загроз і ризиків, яке дозволяє визначати активи, можливі вектори атак, критичні точки доступу та пріоритети захисту, узгоджуючи технічні рішення з вимогами до конфіденційності, цілісності й доступності [13]. На рівні реалізації застосовуються статичний аналіз і огляд коду для виявлення помилок проектування та небезпечних конструкцій, автоматизоване сканування на відомі вразливості, а також динамічні перевірки, що імітують поведінку зловмисника в контрольованих умовах (тестування на проникнення) [13]. Для класу атак на цілісність даних, зокрема SQL-ін'єкцій, важливим є поєднання перевірок на рівні коду, конфігурації БД і поведінкових тестів, які демонструють стійкість системи до модифікації запитів і обходу валідацій [77], [20].

Поряд із підходами, орієнтованими на незмінність журналів і контроль цілісності даних, у літературі представлені моделі виявлення SQL-ін'єкцій на основі нейромережевого аналізу [20]. У таких підходах детекція атаки розглядається як задача класифікації вхідних параметрів і запитів за набором ознак, а для нелінійного перетворення вхідних даних застосовуються типові функції активації нейронної мережі:

$$f(x) = \frac{1-e^{-x}}{1+e^{-x}}, \quad (1.23)$$

де x - зважена сума вхідних ознак. Ця функція використовується для відображення вхідних даних у нелінійний простір ознак. За потреби для виділення характерних шаблонів у параметрах запиту застосовуються інші нелінійні перетворення [20]. Такий підхід є важливим, як приклад моделі, що забезпечує автоматизоване

виявлення аномальних або потенційно шкідливих SQL-конструкцій на основі навчання за даними. Водночас його обмеження полягає в тому, що він зосереджений передусім на класифікації атак і не охоплює повною мірою задачі доказової перевірки подій, незмінного журналювання та моделювання довіри в архітектурі вебзастосунку (Табл. 1.1).

Таблиця 1.1

Характеристика нейромережевого підходу до виявлення SQL-ін'єкцій

Елемент підходу	Характеристика	Науково-прикладне значення
Тип моделі	Нейромережевий підхід до виявлення SQL-ін'єкцій	Забезпечує автоматизовану детекцію підозрілих SQL-конструкцій
Математична основа	Функція активації	Дає змогу реалізувати нелінійну класифікацію шкідливих шаблонів
Рівень застосування	Аналіз параметрів запитів і SQL-операторів	Орієнтована на підсистему виявлення атак у вебзастосунках
Обмеження	Не забезпечує незмінного журналювання та аудиту подій	Вказує на доцільність поєднання з засобами контролю цілісності та перевірюваності дій

Окремий аспект оцінювання стосується контролю цілісності журналів і можливості аудиту, система вважається більш надійною, коли критичні події мають захищений, перевірюваний слід, а механізми логування стійкі до підробки й “редагування заднім числом” [42], [43]. У межах комплексного оцінювання також враховується експлуатаційна ефективність захисту, показники точності виявлення (частка помилкових спрацювань і пропусків), стійкість до зміни атак, затримки обробки запитів, вплив на продуктивність та організаційні витрати на супровід [66], [107]. Під час проєктування блокчейн-компонента для вебсистем важливо враховувати компроміси між децентралізацією, узгодженістю та масштабованістю (DCS/трілема), які безпосередньо впливають на вибір типу мережі, механізму консенсусу та спосіб зберігання журналів. Це пояснює практичну доцільність permissioned-підходів і схем контрольних значень, коли повні журнали лишаються у контрольованому сховищі, а блокчейн використовується як незалежний доказ незмінності [36], [2]. Такий підхід узгоджується з вимогою практичної впроваджуваності, коли безпека має підвищувати довіру до сервісу без надмірних

накладних витрат і ускладнення адміністрування [13], [66]. Принципи безпечної побудови вебсистем і підходи до оцінювання їх безпеки доцільно розглядати як єдину систему, архітектурні та процесні рішення формують стійкість до загроз, а комплекс перевірок і метрик забезпечує вимірюваність рівня захисту, виявлення слабких місць і обґрунтування подальших удосконалень [13], [66].

1.3. Підходи на основі блокчейну для фіксації подій, перевірки цілісності та аудиту змін

Забезпечення довіри до результатів роботи вебсистеми тісно пов'язане з можливістю підтвердити, що критичні події та дані не були змінені непомітно, а історія операцій зберігається у формі, придатній для перевірки під час аудиту або розслідування інциденту [13]. У практичних умовах класичне журналювання часто є вразливим до постфактум-редагування, а саме, логи можуть бути змінені адміністратором, скомпрометованим обліковим записом або шкідливим кодом, а відновлення достовірної послідовності подій стає складним [42].

З іншого боку підходи на основі блокчейну розглядаються як засіб підсилення довіри через властивості незмінності записів, прозорості фіксації подій та стійкості журналів до маніпуляцій [42], [43]. У контексті вебсистем блокчейн доцільно трактувати не як заміну традиційних механізмів безпеки, а як додатковий шар контролю цілісності та доказовості. Практичний зміст такого підходу полягає в тому, що критичні дії (автентифікація/авторизація, зміни прав доступу, транзакції, модифікації записів у БД, виконання адміністративних операцій, виклики API, зміни конфігурацій) мали захищений «слід» у вигляді записів, які неможливо непомітно видалити або підмінити без залишення ознак втручання [42]. У такому підході блокчейн виконує функцію незалежного реєстру, що зменшує довіру до одного вузла або одного адміністратора й підвищує можливість перевірки, коректність історії подій встановлюється через відтворення та верифікацію ланцюга записів [43].

Технічно підходи до фіксації подій на основі блокчейну зазвичай спираються на поєднанні двох моделей. Перша модель [42] полягає у формуванні незмінного

журналу шляхом криптографічного зв'язування записів, де кожна подія має нормалізоване представлення (тип події, ідентифікатор об'єкта, суб'єкт дії, часові атрибути, контекст запиту, результат), від якого обчислюється хеш, далі цей хеш зв'язується з попереднім хешем журналу, утворюючи ланцюг, де підміна одного запису порушує цілісність усієї наступної послідовності. Друга модель [43] передбачає збереження контрольних значень у блокчейні, де замість запису всіх деталей подій у ланцюг зберігають агрегований доказ (наприклад, корінь дерева хешів для пакета подій за інтервал часу), а повний зміст журналу залишається поза блокчейном у внутрішньому сховищі. Це дозволяє зменшити накладні витрати й водночас отримати криптографічно перевірюваний доказ того, що журнал за певний період не редагувався після фіксації контрольного значення [42], [43].

Підходи на основі блокчейну доцільно розглядати як механізм фіксації критичних подій, перевірки цілісності записів, аудиту змін і контролю доступу до захищених ресурсів. Базовим елементом такого підходу є хеш-функція, яка перетворює довільне повідомлення або запис події у значення фіксованої довжини, що подано у формулі (1.24) [37]

$$h: X \rightarrow Y, \quad h(M) = H, \quad (1.24)$$

де h - хеш-функція, X - множина всіх можливих повідомлень, Y - множина бінарних векторів фіксованої довжини, M - вхідне повідомлення або запис події, H - хеш-значення, або криптографічний відбиток повідомлення. У вебсистемі M може відповідати запису журналу, критичній дії користувача, SQL-операції, зміні прав доступу, зміні даних або транзакційній події. Хеш-функція у джерелі [37] подається як відображення $h: X \rightarrow Y$, а результат хешування повідомлення - як $h(M) = H$. Хешування може будуватися на основі одноетапної функції стискання, що наведено у формулі [37]:

$$y = f(x_1, x_2), \quad (1.25)$$

де f - функція стискання, x_1 - блок повідомлення, x_2 - попередній внутрішній стан або проміжне хеш-значення, y - нове проміжне хеш-значення, t - довжина блоку повідомлення, n - довжина результату стискання. Якщо повідомлення поділяється на блоки M_1, M_2, \dots, M_N , то послідовне обчислення хешу описується формулою [37]:

$$H_0 = v, H_i = f(M_i, H_{i-1}), \quad i = 1, \dots, N, \quad h(M) = H_N, \quad (1.26)$$

де H_0 - початкове значення, v - ініціалізаційний вектор або початкова константа, M_i - i -й блок повідомлення, H_{i-1} - попереднє проміжне хеш-значення, H_i - поточне проміжне хеш-значення, N - кількість блоків повідомлення, H_N - фінальний хеш усього повідомлення. Така схема дає змогу перевіряти цілісність великих журналів, наборів транзакцій або послідовностей подій у вебсистемі. У джерелі [37] також зазначено, що хешування широко використовується для перевірки цілісності даних, а в блокчейні хеш гарантує цілісність блоку, оскільки вхідні дані для хешування містять хеш попереднього блоку. Для фіксації змін у вебресурсі блок можна подати як структурований запис, що наведено у формулі [37]:

$$B_i = (index_i, t_i, D_i, H_{i-1}, H_i), \quad (1.27)$$

де B_i - i -й блок, $index_i$ - номер блоку, t_i - часова мітка створення запису, D_i - дані блоку, H_{i-1} - хеш попереднього блоку, H_i - хеш поточного блоку. У джерелі [37] блок містить номер, часову мітку, хеш попереднього блоку та хеш поточного блоку, а даними блоку є транзакційна інформація: відправник, отримувач і сума. Для аудиту подій вебсистеми цю структуру доцільно адаптувати у вигляді формули [37]:

$$E_i = (user_i, action_i, object_i, t_i, result_i), \quad (1.28)$$

де E_i - критична подія вебсистеми, $user_i$ - користувач або сервіс, який виконав дію, $action_i$ - тип дії, $object_i$ - об'єкт дії, t_i - час події, $result_i$ - результат виконання дії. На основі такої події хеш поточного запису може бути сформований за формулою:

$$H_i = SHA256(H_{i-1} \parallel t_i \parallel E_i), \quad (1.29)$$

де $SHA256()$ - криптографічна хеш-функція, \parallel - операція конкатенації. У такій формалізації кожен запис залежить від попереднього, що дозволяє виявити будь-яке порушення цілісності. Це безпосередньо узгоджується з вимогами до побудови підсистем довіри, де важливо не лише зберігати події, а й забезпечувати доказовість їх незмінності. Перевірка коректності журналу формалізується умовою, наведеною у формулі:

$$Valid(BC) = \forall i: (H_i = Hash(B_i)) \wedge (PrevHash_i = H_{i-1}), \quad (1.30)$$

де $Valid(BC)$ - результат перевірки коректності блокчейн-ланцюга, BC - блокчейн-журнал, B_i - поточний блок, H_i - збережений хеш поточного блоку, $Hash(B_i)$ -

повторно обчислений хеш, $PrevHash_i$ - збережене посилання на попередній хеш. Таким чином, блокчейн забезпечує формальну перевірку цілісності журналу подій, що є ключовим для розслідування інцидентів і аналізу безпеки вебсистем. Окрім журналювання, блокчейн-підходи застосовуються для контролю доступу через механізм токенів і делегування прав. Формально множини доступу задаються у формулі:

$$S = s_i, \quad O = o_i, \quad P = p_i, \quad S_{legal} \subseteq S, \quad (1.31)$$

де S - множина суб'єктів доступу, O - множина об'єктів доступу, P - множина дозволів, S_{legal} - множина легальних суб'єктів. Проблема виникає у випадку неконтрольованого делегування прав, що подано у формулі:

$$s_{from} \in S_{legal}, \quad s_{to} \notin S_{legal} \Rightarrow UnauthorizedAccess, \quad (1.32)$$

де s_{from} - суб'єкт, який передає дозвіл, s_{to} - суб'єкт, який отримує дозвіл. Це означає, що передача прав без перевірки політики доступу може призводити до порушення довіри в системі. Для обмеження такого ризику використовується модель токен-обмеженого делегування, що наведена у формулі:

$$PT = f(s_i) \rightarrow \{o_j, p, N, C\}, \quad (1.33)$$

де PT - токен дозволу, $f(s_i)$ - функція відображення суб'єкта на токен, o_j - об'єкт доступу, p - множина дозволів, N - характеристики токена, C - обмеження доступу, сформовані на основі політики. Делегування дозволу відбувається лише за умови, наведеною у формулі:

$$Delegation(s_{from}, s_{to}, pt) = true \Leftrightarrow Decision(s_{to}, C_{pt}) = true, \quad (1.34)$$

де $Delegation$ - результат делегування, pt - токен дозволу, $Decision(s_{to}, C_{pt})$ - результат перевірки відповідності суб'єкта обмеженням токена. Такий підхід дозволяє інтегрувати політику доступу безпосередньо у токен, що зменшує ризик несанкціонованого використання прав. Узагальнення підходів журналювання та контролю доступу дає змогу сформувати аудитний запис, поданий у формулі:

$$A_i = (E_i, PT_i, Decision_i, t_i, H_{i-1}, H_i), \quad H_i = SHA256(H_{i-1} \parallel E_i \parallel PT_i \parallel Decision_i \parallel t_i), \quad (1.35)$$

де A_i - аудитний запис, E_i - подія вебсистеми, PT_i - токен дозволу або його ідентифікатор, $Decision_i$ - результат рішення політики доступу, t_i - часова мітка, H_{i-1}

- хеш попереднього аудитного запису, H_i - хеш поточного запису. У цьому випадку фіксується не лише подія, а й підстава її виконання, що підвищує обґрунтованість прийнятих рішень. Перевірка такого журналу визначається умовою, наведеною у формулі:

$$ValidAudit(A) = \forall i: (H_i = Hash(A_i)) \wedge (PrevHash_i = H_{i-1}) \wedge (Owner(PT_i) \in S_{legal}), \quad (1.36)$$

де $ValidAudit(A)$ - результат перевірки аудитного журналу, A_i - окремий аудитний запис, $Hash(A_i)$ - повторно обчислений хеш запису, $PrevHash_i$ - збережене посилання на попередній хеш, $Owner(PT_i)$ - власник токена дозволу, S_{legal} - множина легальних суб'єктів відповідно до політики доступу.

Окремим підходом є застосування смарт-контрактів як формалізованого механізму контролю правил фіксації та перевірки. Смарт-контракт може виступати «політикою аудиту», яка задає, які події підлягають реєстрації, атрибути мають бути включені, обмеження є припустимими, і як виконується перевірка цілісності журналу (Рис.1.1) [43]. У задачах захисту БД і протидії SQL-ін'єкціям блокчейн-підхід розглядається також як додатковий рівень контролю, коли критичні запити або параметризовані шаблони запитів фіксуються та перевіряються через смарт-контракти, що підвищує контрольованість дій і ускладнює приховану модифікацію операцій доступу [77]. Така логіка не замінює класичні практики безпечної роботи з БД, але посилює можливість аудиту й доведення коректності виконаних операцій у випадку інциденту [77], [13].

Для практичного використання у вебсистемах важливим є вибір архітектури блокчейн-інтеграції. У сценаріях корпоративних сервісів, де потрібні керованість, контроль доступу й прогнозовані накладні витрати, доцільними є permissioned/consortium-підходи, коли учасники мережі відомі, а політики зберігання та доступу визначаються організаційно [43]. Для відкритих середовищ акцент може зміщуватися до публічного «якоріння» контрольних значень як способу посилити незалежність доказу. У будь-якому випадку інтеграція повинна враховувати, що блокчейн-реєстр є шаром забезпечення незмінності, але не автоматично гарантує коректність первинних даних, якщо подія сформована неправильно або зафіксована неповно, блокчейн лише «консервує» цю помилку [13]. Блок фіксації подій має бути

узгоджений із принципами безпечної архітектури, нормалізації даних і контролю доступу, а також із процедурою визначення переліку критичних подій, які справді впливають на довіру до системи [13].

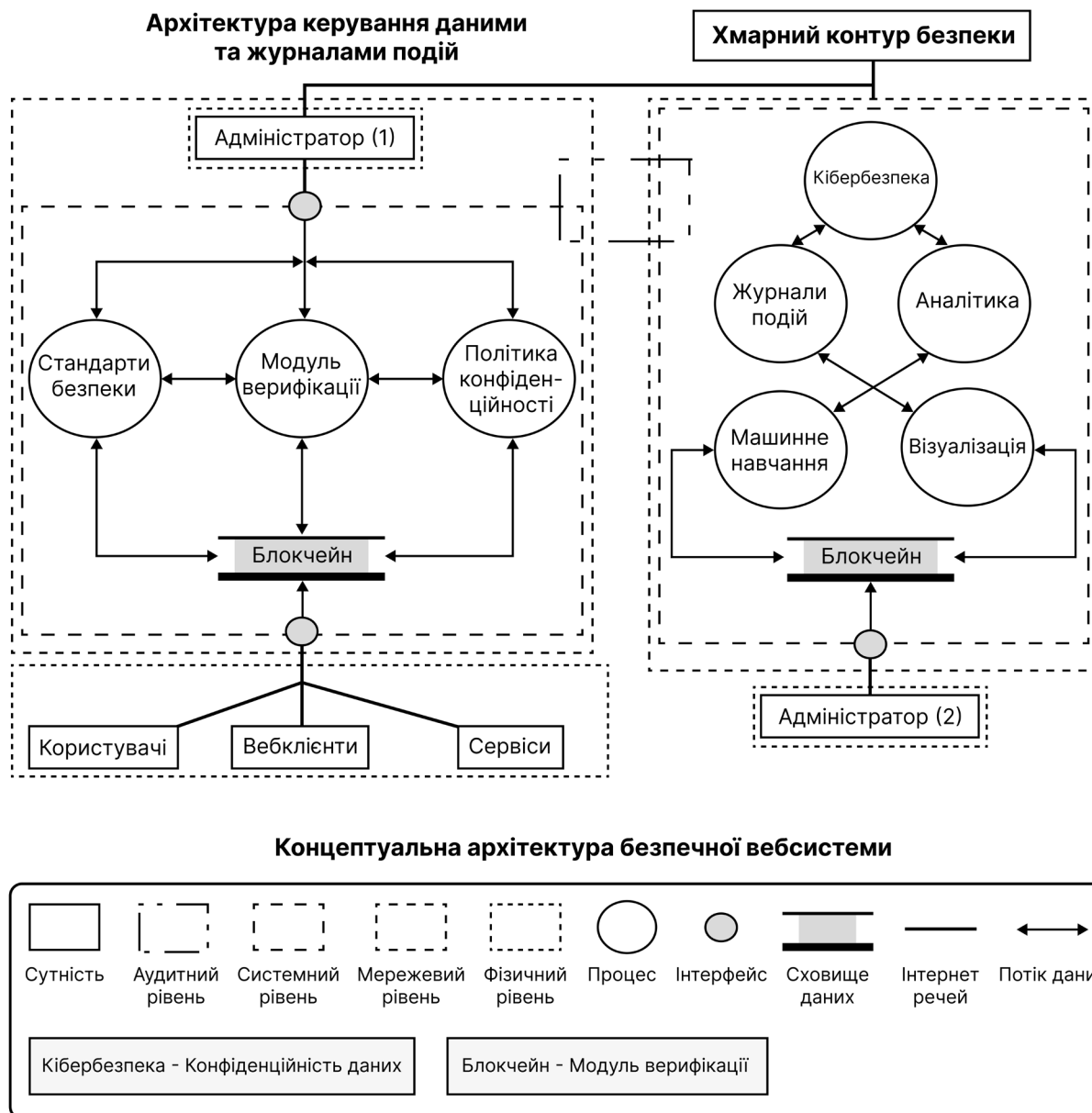


Рис.1.1 Концептуальна архітектура безпечної вебсистеми з контуром керування журналами подій та аналітичним контуром безпеки

Ключовою перевагою блокчейн-підходів для аудиту змін є доказовість зафіксованих подій і змін, аудит у такій моделі означає, що сторонній або внутрішній контролер може відтворити послідовність подій за журналом, перевірити хеш-зв'язність, співставити контрольні значення з блокчейн-записами і встановити факт відсутності постфактум-редагування у межах зафіксованих інтервалів [42], [43]. Це

особливо важливо для інцидентів, де зловмисник намагається залишатися непоміченим, змінюючи дані або приховуючи сліди доступу, а також для сценаріїв, де організація повинна довести коректність виконаних операцій (фінансові дії, доступ до персональних даних, критичні адміністративні зміни) [13], [42].

Інтеграція систем детекції вторгнень у блокчейн-інфраструктуру може підвищувати ефективність реагування за рахунок узгодженого обміну інформацією про інциденти та забезпечення цілісності повідомлень про атаки між компонентами [22]. Для вебсистем це є аргументом на користь побудови єдиного контуру моніторингу, де події та ознаки аналізу фіксуються у формі, придатній для перевірки та подальшої кореляції.

Разом із тим впровадження блокчейн-механізмів потребує врахування обмежень. По-перше, зростають накладні витрати на запис подій і підтримку реєстру, на практиці застосовують вибірккову фіксацію лише критичних подій або пакетування з періодичним «якорінням» [43]. По-друге, виникають вимоги до конфіденційності, журнали можуть містити чутливі атрибути, тому частіше фіксують хеші, метадані або псевдонімізовані ідентифікатори, а деталі зберігають у контрольованому сховищі з чіткими політиками доступу [13], [43]. По-третє, критичним стає керування ключами та довіреними компонентами, адже компрометація вузла, що формує події, здатна знизити практичну цінність доказів навіть за наявності незмінного реєстру [13]. Блокчейн-підходи у вебсистемах доцільно розглядати як інструмент підвищення довіри через незмінність і доказовість історії критичних подій, вони забезпечують захищений контур аудиту змін, ускладнюють приховану фальсифікацію журналів і підтримують відтворюваність подій у разі розслідування [42], [43]. У поєднанні з класичними механізмами контролю доступу та з аналітичними методами виявлення підозрілої активності це формує основу комплексного підходу, де контроль цілісності та доказовість подій доповнюють адаптивні механізми реагування на загрози [13], [77]. Окрім продуктивності та конфіденційності, у працях з аналізу впровадження блокчейну в кібербезпеці окремо виділяють комплекс класів викликів, серед яких, технічні, управлінські, регуляторні, правові, соціально-економічні та екологічні аспекти, що впливають на архітектурні рішення, операційну модель і вартість

супроводу [44]. У контексті вебсистем це означає необхідність формувати вимоги не лише до механізмів фіксації подій, а й до політик зберігання, моделі доступу, процесів ротації ключів та процедур аудиту, які забезпечують практичну відтворюваність доказової бази.

1.4. Методи машинного навчання для виявлення атак, аномалій і підозрілої активності у вебсередовищі

Вебсередовище характеризується високою динамікою загроз, де зловмисники постійно змінюють структуру повідомлень, патерни запитів, інтенсивність трафіку та способи обходу традиційних бар'єрів на кшталт статичних правил, CAPTCHA або чорних списків. Через це підходи, що спираються лише на заздалегідь визначені сигнатури та ручне налаштування фільтрів, поступово втрачають ефективність у задачах виявлення вебспаму, аномального трафіку та прихованих атак на доступність і цілісність сервісу [13]. У таких умовах машинне навчання розглядається як інструмент переходу від реактивного захисту до більш адаптивного аналізу поведінки, де рішення ухвалюються на основі закономірностей у даних, а не лише за рахунок фіксованих правил.

Методи машинного навчання у веббезпеці застосовують для трьох взаємопов'язаних класів задач. Перший клас - класифікація небажаного контенту й запитів, де типовим прикладом є фільтрація вебспаму у формах зворотного зв'язку та інших каналах введення. У цьому випадку модель аналізує текстові та поведінкові ознаки (лексика, повторювані шаблони, структурні елементи, джерело запиту, частота надсилання, часові характеристики) і відокремлює легітимні звернення від підозрілих, знижуючи навантаження на ручну модерацію та підтримуючи адаптацію до нових варіантів атак [13]. Другий клас - виявлення вторгнень і підозрілої активності в інфраструктурі (web/server/DB-рівні), де застосовують як класичні методи класифікації, так і моделі глибинного навчання. Зокрема, у дослідженнях детекції DDoS-атак у вебтрафіку запропоновано автоенкодерний підхід з адаптивною трирівневою організацією аналізу, що підтверджує придатність моделей без учителя

та напівкерованого навчання для виявлення складних атак у динамічному середовищі [81]. Третій клас - пошук аномалій у трафіку й журналах подій, що є особливо важливим для атак на доступність та для сценаріїв, де зломисник намагається маскуватися під легітимну поведінку [13]. У прикладних дослідженнях детекції ін'єкційних атак і шкідливих URL спостерігається зсув у бік моделей глибокого навчання, зокрема, у роботі [21] розглянуто ANN, BiLSTM, BERT, Random Forest, XGBoost, AdaBoost і KNN, причому BERT показує найкращі результати для SQLi та NoSQLi, а BiLSTM - для виявлення шкідливих URL-адрес. Для вебсистем це важливо як підтвердження того, що виявлення небажаної активності має спиратися на моделі, здатні адаптуватися до варіативності атак і водночас підтримувати керований рівень хибних спрацьовувань.

Показовим прикладом багаторівневої організації фільтрації небажаних повідомлень є поєднання серверного та клієнтського рівнів перевірки [66]. Хоча така архітектура сформувалася насамперед у системах електронної пошти, її логіка є релевантною і для задач вебспау, де первинна фільтрація звернень може виконуватися на межі інфраструктури, а додатковий аналіз - на рівні прикладного сервісу або кінцевого клієнта (Рис. 1.2).

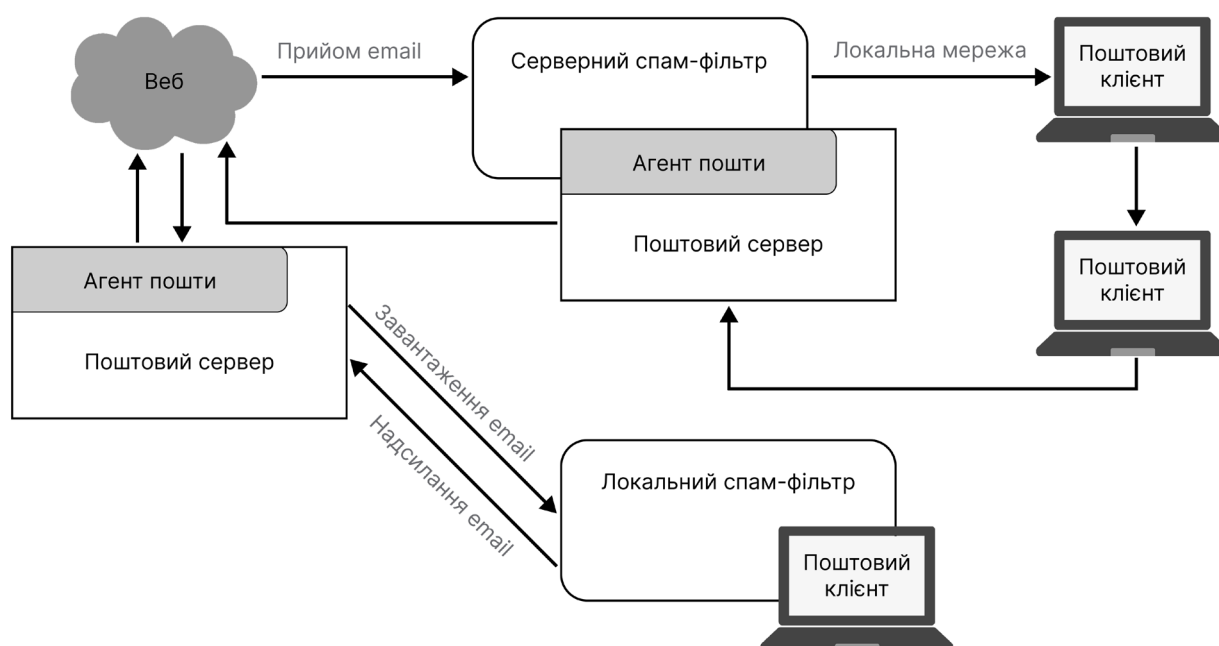


Рис.1.2 Архітектура клієнтської та серверної фільтрації спау в електронній пошті

Вибір моделі залежить від того, які дані доступні та які вимоги висуваються до пояснюваності й швидкодії. Для задач, де є мітки класів і накопичені приклади атак/спаму, ефективними є методи навчання з учителем, байєсівські класифікатори, дерева рішень, ансамблеві методи, зокрема Random Forest, нейронні мережі та їх комбінації [78], [81]. Для ситуацій, коли повних міток немає або атаки є новими, вагому роль відіграють підходи без учителя та напівкеровані методи, орієнтовані на моделювання «нормальної» поведінки й фіксацію відхилень. Такі методи працюють на потоці подій та сигналізувати про аномальні стани, які потребують перевірки [13]. Практична цінність цього класу методів полягає у здатності виявляти невідомі або комбіновані сценарії атак без необхідності оперативно формалізувати їх у вигляді нових правил. Окремої уваги заслуговує використання моделей, що враховують не лише зміст події, а й її контекст та зв'язки з іншими об'єктами вебсистеми. Для вебспаму це принципово, оскільки зловмисники змінюють тексти, але часто зберігають характерні взаємозв'язки, спільні джерела запитів, повторювані шаблони поведінки, «серійність» надсилання, схожі траєкторії переходів, типові комбінації параметрів форм і часові патерни [13]. В такому ключі перспективними є графові підходи, де дані представляються як граф взаємодій (користувач/сесія/форма/повідомлення/IP/endpoint), а моделі навчаються виявляти підозрілі підграфи та нетипові структури зв'язків. Така постановка задачі узгоджується з практичною потребою виявляти не тільки одиничні «погані» повідомлення, а й кампанії атак, що проявляються через сукупність взаємопов'язаних подій у вебсередовищі [13].

Ефективність машинного навчання у веббезпеці значною мірою визначається організацією повного контуру, де відбувається збір даних, підготовка ознак, навчання/валідація, впровадження та безперервне оновлення. Оскільки вебатаки еволюціонують, ключовим є явище дрейфу концепції, через розподіл ознак «спаму» або «підозрілої активності», що з часом змінюється, і модель потребує регулярного до навчання або адаптації на нових прикладах [13]. Додатковими практичними викликами є дисбаланс класів, вартість помилкових спрацьовувань, а також вимоги до інтерпретованості рішень у прикладних сценаріях, де необхідно пояснювати

причину блокування звернення чи маркування активності як підозрілої [78], [81]. Машинне навчання у вебсередовищі забезпечує адаптивний рівень захисту, який доповнює традиційні механізми безпеки та дозволяє виявляти спам, аномалії й підозрілу активність у ситуаціях, де статичні правила швидко застарівають [13]. Особливу практичну цінність мають моделі, що поєднують аналіз контенту з аналізом взаємодій та залежностей між подіями, а також підтримують оновлення на основі нових даних експлуатації, зберігаючи прийнятний баланс між чутливістю до атак і стабільністю роботи сервісу [78], [81].

1.5. Постановка наукового завдання та визначення мети і завдань дослідження

Аналіз сучасних підходів до забезпечення безпеки вебсистем показує, що їх основним обмеженням є фрагментарність, одні засоби орієнтовані переважно на виявлення окремих класів атак, інші - на контроль доступу або журналювання подій, однак вони, як правило, не формують єдиної архітектурно узгодженої підсистеми, у межах якої поєднуються забезпечення довіри до критичних подій, контроль цілісності даних, можливість аудиторської перевірки рішень і адаптивне виявлення підозрілої активності [13], [78], [81]. У багатокomпонентних вебзастосунках, що інтегруються з API, зовнішніми сервісами, платіжними модулями та каналами введення даних, порушення безпеки проявляються не лише як явний витік інформації, а і як непомітна підміна, викривлення або некоректна інтерпретація подій, що ускладнює підтвердження коректності виконаних операцій, розслідування інцидентів і відтворення послідовності змін у системі. Додатковою проблемою є динамічний характер загроз у вебсередовищі. Статичні правила захисту і сигнатурні механізми не забезпечують достатньої стійкості до нових варіантів вебспаму, аномальної активності, спроб маніпуляції транзакційними подіями чи прихованого втручання в критичні запити. Водночас вимоги до сучасних вебзастосунків передбачають не лише високий рівень захисту, а й збереження прийнятної продуктивності, масштабованості, впроваджуваності та керованості хибними спрацьовуваннями, оскільки надмірна

кількість помилкових реакцій безпосередньо знижує якість сервісу та ускладнює експлуатацію системи [78], [81]. Виникає наукове протиріччя між потребою у побудові архітектурно узгодженої підсистеми довіри та цілісності для вебзастосунків, яка забезпечує доказовість, простежуваність і незмінність критичних подій, та потребою в адаптивному виявленні підозрілої активності, що не зводиться до наперед заданих сигнатур. Це зумовлює необхідність побудови нових моделей і методів, здатних об'єднати контур доказовості та контур інтелектуального моніторингу в межах єдиної архітектури вебсистеми.

Наукове завдання полягає у побудові моделі довіри, методів забезпечення цілісності критичних подій і даних та інтегрованого контуру довіри для вебсистем, що ґрунтуються на поєднанні положень теорії довіри до інформаційних систем, криптографічних методів контролю цілісності, принципів незмінного журналювання подій, графового подання зв'язків між подіями та методів машинного навчання для виявлення підозрілої активності, і забезпечують архітектурно узгоджене підвищення безпеки вебзастосунків за рахунок доказовості рішень, простежуваності змін, зменшення ризику прихованих модифікацій та підвищення якості виявлення загроз.

Мета дослідження - підвищення рівня довіри, цілісності та захищеності архітектури вебсистем, за рахунок обґрунтування та побудови моделей і методів, що забезпечують контроль цілісності даних, визначення критичних подій і адаптивне виявлення підозрілої активності у вебсередовищі на основі технологій блокчейну та машинного навчання.

Для досягнення поставленої мети необхідно розв'язати такі наукові завдання:

1. Проаналізувати існуючі моделі, методи й архітектурні підходи до забезпечення безпеки вебсистем, зокрема в частині забезпечення довіри, контролю цілісності, аудиту критичних подій і виявлення динамічних загроз у вебзастосунках.
2. Удосконалити науково-методичні положення щодо формалізації задачі забезпечення довіри й цілісності у вебсистемах з урахуванням вимог до фіксації критичних подій, контролю їх коректності та виявлення підозрілої активності.

3. Розробити модель інтегрованого контуру довіри й цілісності у вебсистемі для формалізованого подання процесів фіксації, контролю, аудиту, верифікації та аналізу критичних подій у межах архітектури вебзастосунку.

4. Розробити метод блокчейн-верифікованого журналювання критичних подій і контролю доступу у вебсистемах для забезпечення цілісності записів, виявлення несанкціонованих змін, підвищення доказовості аудиту та підтримки прийняття рішень щодо доступу.

5. Розробити метод графово-нейромережевого виявлення вебспаму та підозрілої активності у вебсистемах для підвищення точності розпізнавання небажаних, підозрілих і потенційно небезпечних дій у змінних умовах функціонування вебзастосунків.

6. Розробити метод інтегрованого забезпечення довіри й цілісності у вебсистемах для узгодженого використання результатів фіксації, контролю, верифікації, аудиту та аналітичного оцінювання критичних подій у процесі прийняття рішень.

7. Реалізувати прототип програмного рішення на основі розроблених моделі та методів і провести експериментальні дослідження його функціонування в типових сценаріях використання вебсистем із оцінюванням точності, повноти, F1-міри, частки хибних спрацьовувань, швидкодії та накладних витрат.

Об'єкт дослідження - процеси забезпечення довіри і цілісності вебсистем.

Предмет дослідження - моделі та методи забезпечення довіри, цілісності та надійності вебсистем на рівні архітектури.

Для розв'язання поставлених завдань у роботі застосовано теорію графів для подання взаємозв'язків між подіями, об'єктами та рішеннями у вебсистемі, положення теорії довіри до інформаційних систем для формалізації моделі довіри, криптографічні методи хешування та цифрового підпису для забезпечення контролю цілісності, підходи блокчейн-технології та незмінного журналювання для фіксації критичних подій, методи машинного навчання, зокрема графові нейронні мережі, для виявлення вебспаму та підозрілої активності, методи математичної статистики та теорії ймовірностей для оцінювання якості моделей, методи теорії планування

експерименту для організації та інтерпретації результатів експериментальних досліджень.

Висновки до розділу 1

На основі проведено аналізу проблеми забезпечення довіри й цілісності у сучасних вебсистемах, які обробляють критично важливі дані та підтримують бізнес-процеси встановлено, що традиційні механізми захисту та журналювання є необхідними, однак не забезпечують повною мірою незмінність історії критичних подій і можливість доведення коректності виконаних дій у багатокomпонентних вебархітектурах.

Аналіз підходів на основі блокчейну для фіксації подій, перевірки цілісності записів та аудиту змін показав, що криптографічне зв'язування записів, збереження контрольних значень і використання смарт-контрактів можуть застосовуватися як додатковий шар забезпечення незмінності та підзвітності критичних дій. Водночас їх практичне використання потребує врахування обмежень продуктивності, конфіденційності, керування ключами та оптимізації процедур перевірки.

Аналіз методів машинного навчання для виявлення атак, аномалій, вебспаму та підозрілої активності показав їхню доцільність у задачах, де статичні правила є недостатньо гнучкими. Особливу увагу приділено підходам, що враховують не лише окремі ознаки подій, а й зв'язки між користувачами, об'єктами, діями та технічними параметрами. Це підтверджує доцільність використання графових і гібридних моделей, оскільки вони підвищують точність, адаптивність і повноту виявлення підозрілої активності у вебсистемах.

Таким чином, результати розділу обґрунтовують необхідність розроблення інтегрованого підходу, який поєднує механізми криптографічної фіксації подій, контроль цілісності, аудит змін і інтелектуальне виявлення підозрілої активності для підвищення довіри до роботи вебсистем.

РОЗДІЛ 2. МОДЕЛЬ ІНТЕГРОВАНОВОГО КОНТУРУ ДОВІРИ Й ЦІЛІСНОСТІ ТА МЕТОД БЛОКЧЕЙН-ВЕРИФІКОВАНОГО ЖУРНАЛЮВАННЯ КРИТИЧНИХ ПОДІЙ І КОНТРОЛЮ ДОСТУПУ У ВЕБСИСТЕМАХ

2.1. Модель інтегрованого контуру довіри й цілісності у вебсистемах

Забезпечення довіри й цілісності у вебсистемах потребує не лише фіксації окремих критичних подій, а й їх узгодженого подання в межах єдиного середовища аналізу, прийняття рішень і аудитної перевірки. У межах цього дослідження модель інтегрованого контуру довіри й цілісності розглядається як формальна основа, що поєднує події вебформ, транзакційні операції, їх ознаковий опис, графовий контекст, результати оцінювання ризику та механізми незмінного журналювання. Така постановка дозволяє перейти від ізольованого розгляду подій до цілісного подання, у якому рішення системи безпеки є не лише результатом обчислення, а й відтворюваним та перевірюваним артефактом. Модель інтегрованого контуру довіри й цілісності задається кортежем:

$$M_{\text{ІКДЦ}} = \langle E, V, R, \Phi, \Gamma, P, \Lambda \rangle, \quad (2.1)$$

де E позначає множину критичних подій вебсистеми, V - множину сутностей, пов'язаних із подіями, R - множину відношень між цими сутностями, Φ - відображення формування ознак, Γ - відображення прийняття рішення щодо події, P - множину політик реагування, а Λ - незмінний журнал подій і рішень. З формальної точки зору множина E відповідає потоку подій безпеки, V та R визначають структуру графового подання, Φ відповідає модулю побудови векторів ознак, Γ - модулю оцінювання ризику і класифікації, P - набору правил реагування, а Λ - ланцюгу блоків або іншому механізму незмінного журналювання. Кожна критична подія подається у вигляді кортежу:

$$e_t = (id_t, type_t, ts_t, actor_t, session_t, resource_t, ctx_t, payload_t), \quad (2.2)$$

де e_t є подією з порядковим номером t , id_t - її унікальним ідентифікатором, $type_t$ - типом події, ts_t - часовою міткою, $actor_t$ - суб'єктом ініціації, $session_t$ - ідентифікатором сесії, $resource_t$ - цільовим ресурсом або функціональним об'єктом, ctx_t - контекстом виконання, а $payload_t$ - корисним навантаженням події. У прикладному аспекті id_t є або цілим числом або рядковим UUID, ts_t задається у форматі Unix time, а $type_t$ належить скінченній множині типів (SUBMIT, TX), де SUBMIT відповідає події надсилання вебформи, а TX - транзакційній події. Контекст події як набір технічних і поведінкових атрибутів у загальнішому вигляді інтерпретується як структурований кортеж:

$$ctx_t = (ip_t, ua_t, device_t, geo_t, ref_t, net_t), \quad (2.3)$$

де ip_t задає IP-адресу або її префікс, ua_t - User-Agent або його відбиток, $device_t$ - тип пристрою, geo_t - географічну зону, ref_t - джерело переходу, а net_t - додаткові мережеві параметри. Контекст зберігається у нормалізованому структурованому форматі, в JSON-представленні з фіксованими полями. Для транзакційної події $type_t = TX$, її корисне навантаження описується окремим профілем:

$$X_i = (T_i, P_i, C_i, V_i, R_i), \quad (2.4)$$

де X_i є узагальненим профілем i -тої транзакції, T_i містить її ідентифікаційні атрибути, P_i - платіжні реквізити, C_i - дані електронного чека, V_i - верифікаційні дані, а R_i - відповідь зовнішнього сервісу. Компонент T_i включає унікальний ідентифікатор транзакції та канал оплати, P_i - суму платежу, час ініціації і спосіб оплати, C_i - номер чека, V_i - номер телефону і одноразовий код підтвердження, а R_i - статус відповіді зовнішнього сервісу та текстове повідомлення або код помилки. Таке подання дозволяє розглядати вебформи й транзакції як два різновиди одного класу критичних подій, для яких надалі застосовується спільний контур обробки. Для забезпечення однозначності подальших перевірок кожна подія спочатку приводиться до канонічного подання:

$$c_t = \text{canon}(e_t), \quad (2.5)$$

де $\text{canon}(\cdot)$ є функцією канонізації, що впорядковує атрибути, приводить часові значення до єдиного формату, уніфікує структуру вкладених полів і формує стабільний серіалізований рядок. Тобто, одна й та сама подія за однакового набору

атрибутів має давати ідентичне канонічне представлення незалежно від середовища обробки. На основі канонічного подання обчислюється криптографічний хеш:

$$h_t = H(c_t), \quad (2.6)$$

де h_t є цифровим відбитком події, а $H(\cdot)$ - криптографічною хеш-функцією. У реалізації використовується SHA-256, h_t містить довжину 256 біт. Будь-яка зміна в події після канонізації призводить до зміни h_t , що є базовим механізмом контролю цілісності. Після хешування формується цифровий підпис:

$$\text{sig}_t = \text{Sign}_{sk}(h_t), \quad (2.7)$$

де sig_t є підписом хешу події, а sk - закритим ключем довіреного сервісного компонента. Функція $\text{Sign}_{sk}(\cdot)$ реалізується як алгоритм генерації цифрового підпису, де використовується стандартний алгоритм асиметричного підпису ECDSA, який забезпечує автентичність джерела події та неможливість підробки підпису без знання закритого ключа [151]. Для перевірки використовується відповідний відкритий ключ pk та функція:

$$\text{Verify}_{pk}(h_t, \text{sig}_t) \in 0,1, \quad (2.8)$$

де значення 1 означає, що підпис є коректним, а значення 0 - що цілісність або походження запису порушено. У такий спосіб подія отримує не лише формальний опис, а й криптографічно перевірювану прив'язку до джерела її реєстрації. Оскільки ризик події визначається не тільки її власними атрибутами, а й оточенням, модель вводить графове подання контексту:

$$G = (V_G, E_G), \quad (2.9)$$

де V_G є множиною вузлів, а $E_G \subseteq V_G \times V_G$ - множиною зв'язків між ними. Вузли відповідають подіям, користувачам, сесіям, пристроям, ресурсам, формам та транзакціям. Ребра графа відображають причинно-часові, технічні, поведінкові чи семантичні зв'язки. Відповідно дані з різних джерел повинні бути перетворені в єдину графову структуру, у якій можна відновити не лише сам факт події, а й її контекст. Для кожної події формується вектор ознак:

$$x_t = \Phi(e_t, G_t) \in \mathbb{R}^d, \quad (2.10)$$

де x_t є числовим ознаковим поданням події, $\Phi(\cdot)$ - функцією побудови ознак, G_t - локальним графовим контекстом цієї події, $d \in \mathbb{N}$ - кількістю ознак, \mathbb{R}^d означає,

що вектор x_t складається з d дійсних чисел. Значення d визначається схемою ознак, що реалізується в системі. Для вебформ до ознак входять текстові, структурні, часові й мережеві параметри, а для транзакцій - сума, час, канал, географічний фактор і тип платіжного інструмента. Для транзакційних подій у межах моделі задається інтегральна оцінка ризику:

$$\tilde{R}_{TX} = \alpha_P R_P(P) + \alpha_C R_C(C) + \alpha_T R_T(T) + \alpha_K R_K(K), \quad (2.11)$$

$$\alpha_P + \alpha_C + \alpha_T + \alpha_K = 1, \alpha_P, \alpha_C, \alpha_T, \alpha_K \geq 0, \quad (2.11a)$$

$$R_{TX} = \min \left(1, \max (0, \tilde{R}_{TX}) \right), \quad (2.11b)$$

де \tilde{R}_{TX} - ненормалізована сумарна оцінка ризику транзакції, R_{TX} є загальною оцінкою ризику транзакції, $R_P(P)$ - ризиком, пов'язаним із сумою платежу, $R_C(C)$ - географічним ризиком, $R_T(T)$ - часовим ризиком, а $R_K(K)$ - ризиком, пов'язаним із типом платіжного інструмента. Змінна (P) означає суму платежу, C - географічну характеристику або зону, T - час виконання транзакції, а K - тип платіжного інструмента, значення R_{TX} нормалізовано до інтервалу $[0,1]$. Компонента ризику суми задається логарифмічною функцією:

$$R_P(P) = \min (1, a \ln (P + 1)), \quad (2.12)$$

де $P \in \mathbb{R}_{>0}$ є сумою транзакції, а $a > 0$ - коефіцієнтом чутливості, який визначає, наскільки швидко зростає ризик при збільшенні суми. Значення a має бути зафіксоване в конфігурації моделі. Часова компонента ризику задається функцією Гаусса:

$$R_T(T) = \min \left(1, b \cdot \exp \left(-\frac{(T-\mu)^2}{2\sigma_T^2} \right) \right), \quad (2.13)$$

де $T \in \mathbb{R}_{\geq 0}$ є часовою координатою транзакції, $b > 0$ - коефіцієнтом ваги часового фактора, μ - центром пікового ризику, а $\sigma_T > 0$ - параметром ширини розподілу. Для реалізації це означає, що система явно зберігає значення a , b , μ та σ_T , а також уніфікувати часову шкалу, використовуючи секунди від початку доби. Географічна компонента $R_C(C)$ та компонента типу інструмента $R_K(K)$ задається через словник, де кожному допустимому значенню C та K відповідає певний коефіцієнт ризику. На основі вектора ознак x_t обчислюється оцінка ризику події:

$$p_i = f_\theta(x_i) = g(w^\top x_i + b_0), \quad p_i \in [0,1], \quad (2.14)$$

де $\mathbb{R}^d \ni x_i$ - вектор ознак події, $\mathbb{R}^d \ni w$ - вектор параметрів моделі, $\mathbb{R} \ni b_0$ - зсув, а $g(\cdot)$ - функція нормалізації, що переводить результат у діапазон $([0,1])$, f_θ є функцією оцінювання ризику з параметрами θ , а p_i - числовою оцінкою небезпеки події. Параметри моделі задаються як $\theta = (w, b_0)$. Значення p_i інтерпретується як скаляр у межах від 0 до 1, де 0 відповідає мінімальному ризику, а 1 - максимальному.

$$g(z) = \frac{1}{1+e^{-z}}, \quad (2.15)$$

де $\mathbb{R} \ni z$ проміжне ненормалізоване значення, а $g(z) \in [0,1]$ - результат логістичної нормалізації, що дозволяє інтерпретувати p_i як нормалізовану оцінку ризику події. Параметри моделі $\theta = (w, b_0)$, визначаються на основі множини попередньо зафіксованих подій вебсистеми, для яких відомий їхній стан з погляду ризику. Для цього формується навчальна вибірка:

$$D_{tr} = \{(x_i, y_i)\}_{i=1}^n, \quad x_i \in \mathbb{R}^d, \quad y_i \in \{0,1\}, \quad (2.16)$$

де D_{tr} - навчальна множина подій, x_i - вектор ознак i -ї події, сформований відповідно до (2.10), y_i - цільова мітка події, якщо $y_i = 0$ то відповідає легітимній події, а $y_i = 1$ - небезпечній події, n - кількість подій у навчальній множині, d - кількість ознак у векторі події.

Для кожної події навчальної вибірки проміжна лінійна оцінка визначається як:

$$z_i = w^\top x_i + b_0, \quad (2.17)$$

де $z_i \in \mathbb{R}$ - проміжне ненормалізоване значення ризику для i -ї події, $w = (w_1, w_2, \dots, w_d) \in \mathbb{R}^d$ - вектор вагових коефіцієнтів моделі, $b_0 \in \mathbb{R}$ - зсув моделі, w^\top - транспонування вектора w , тобто добуток $w^\top x_i$ є скалярним добутком:

$$w^\top x_i = \sum_{j=1}^d w_j x_{ij}, \quad (2.18)$$

де w_j - ваговий коефіцієнт j -ї ознаки, x_{ij} - значення j -ї ознаки для i -ї події. Отже, значення z_i є зваженою сумою ознак події з урахуванням зсуву b_0 .

Нормалізована оцінка ризику для події навчальної вибірки визначається як:

$$p_i = g(z_i) = \frac{1}{1 + e^{-z_i}}, \quad (2.19)$$

де $p_i \in [0,1]$ - нормалізована оцінка ризику i -ї події, $g(\cdot)$ - логістична функція нормалізації, визначена у (2.15). Значення p_i інтерпретується як оцінка належності події до ризикового класу.

Для визначення параметрів w та b_0 використовується функція втрат, яка оцінює розбіжність між фактичною міткою події y_i та розрахованою моделлю оцінкою ризику p_i :

$$L(w, b_0) = -\frac{1}{n} \sum_{i=1}^n [y_i \ln(p_i) + (1 - y_i) \ln(1 - p_i)], \quad (2.20)$$

де $L(w, b_0)$ - значення функції втрат для параметрів w та b_0 , y_i - фактична мітка i -ї події, p_i - оцінка ризику цієї події, обчислена за (2.19), $\ln(\cdot)$ - натуральний логарифм. Параметри моделі визначаються як розв'язок задачі мінімізації функції втрат:

$$(w^*, b_0^*) = \arg \min_{w, b_0} L(w, b_0), \quad (2.21)$$

де w^* - оптимальний вектор вагових коефіцієнтів, b_0^* - оптимальне значення зсуву; $\arg \min$ позначає такі значення параметрів, за яких функція втрат набуває мінімального значення. Для ітераційного знаходження параметрів використовується правило градієнтного оновлення:

$$w^{(r+1)} = w^{(r)} - \eta \frac{\partial L}{\partial w}, \quad (2.22)$$

$$b_0^{(r+1)} = b_0^{(r)} - \eta \frac{\partial L}{\partial b_0}, \quad (2.23)$$

де r - номер ітерації навчання, $w^{(r)}$ та $b_0^{(r)}$ - значення параметрів на r -й ітерації; $w^{(r+1)}$ та $b_0^{(r+1)}$ - оновлені значення параметрів, $\eta > 0$ - коефіцієнт швидкості навчання, $\frac{\partial L}{\partial w}$ та $\frac{\partial L}{\partial b_0}$ - часткові похідні функції втрат за відповідними параметрами, задаються як:

$$\frac{\partial L}{\partial w_j} = \frac{1}{n} \sum_{i=1}^n (p_i - y_i) x_{ij}, j = \overline{1, d}, \quad (2.24)$$

$$\frac{\partial L}{\partial b_0} = \frac{1}{n} \sum_{i=1}^n (p_i - y_i), \quad (2.25)$$

де $\frac{\partial L}{\partial w_j}$ - часткова похідна функції втрат за вагою j -ї ознаки, $p_i - y_i$ - похибка моделі для i -ї події; x_{ij} - значення j -ї ознаки цієї події.

Кожен ваговий коефіцієнт w_j визначає внесок відповідної ознаки x_{ij} у формування оцінки ризику події. Якщо певна ознака статистично пов'язана з ризиковими подіями, у процесі мінімізації функції втрат її ваговий коефіцієнт зростає у додатному напрямі. Якщо ознака не має суттєвого впливу на класифікацію, її ваговий коефіцієнт наближається до нуля.

Кінцеве рішення щодо події визначається пороговим правилом:

$$c_i = \begin{cases} \text{SAFE}, & p_i < \tau_1 \\ \text{SUSPECT}, & \tau_1 \leq p_i < \tau_2, \\ \text{THREAT}, & p_i \geq \tau_2 \end{cases} \quad (2.26)$$

де c_i є класом рішення, а $\tau_1, \tau_2 \in [0, 1]$ - керованими порогами класифікації, причому має виконуватися умова $\tau_1 < \tau_2$. У прикладному аспекті це означає, що подія з низьким значенням p_i переходить у штатний режим обробки, подія з проміжним ризиком передається на додаткову перевірку або в карантин, а подія з високим ризиком ініціює блокування, посилену верифікацію чи інший сценарій реагування.

Порогові значення τ_1 та τ_2 визначаються на основі розподілу оцінок ризику у валідаційній вибірці. Для цього формується множина оцінок ризику $P_{val} = p_1, p_2, \dots, p_n$, де P_{val} - множина оцінок ризику для подій валідаційної вибірки, p_i - оцінка ризику i -ї події (2.14), x_i - її вектор ознак, а n - кількість подій у валідаційній вибірці. Середній рівень ризику у валідаційній вибірці визначається як:

$$\mu_p = \frac{1}{n} \sum_{i=1}^n p_i, \quad (2.27)$$

де μ_p показує середній рівень ризику для подій, на яких налаштовуються пороги класифікації. Для врахування розкиду значень ризику відносно середнього обчислюється стандартне відхилення:

$$s_p = \sqrt{\frac{1}{n} \sum_{i=1}^n (p_i - \mu_p)^2}, \quad (2.28)$$

де s_p характеризує варіативність оцінок ризику, що більше значення s_p , то сильніше оцінки ризику відрізняються між собою. Нижній поріг класифікації визначається як:

$$\tau_1 = \max(0, \mu_p - s_p), \quad (2.29)$$

де τ_1 відокремлює події з низьким рівнем ризику від подій, що потребують додаткової уваги. Функція $\max(0, \cdot)$ гарантує, що значення порогу не виходить за нижню межу допустимого діапазону $[0,1]$. Верхній поріг класифікації визначається як:

$$\tau_2 = \min(1, \mu_p + s_p), \quad (2.30)$$

де τ_2 відокремлює події проміжного ризику від подій високого ризику. Функція $\min(1, \cdot)$ гарантує, що значення порогу не перевищує верхню межу допустимого діапазону $[0,1]$. Відповідно, пороги τ_1 та τ_2 адаптуються до фактичного розподілу оцінок ризику в конкретній вебсистемі та задовольняють умову $0 \leq \tau_1 < \tau_2 \leq 1$. Після прийняття рішення формується аудитний запис:

$$a_i = (h_i, sig_i, model_{id_i}, model_{hash_i}, policy_{hash_i}, p_i, c_i, ts_i), \quad (2.31)$$

де a_i містить криптографічний хеш події h_i , її підпис sig_i , ідентифікатор версії моделі $model_{id_i}$, контрольну суму параметрів моделі $model_{hash_i}$, контрольну суму політики реагування $policy_{hash_i}$, оцінку ризику p_i , клас рішення c_i та часову мітку ts_i .

$$model_{hash_i} = H(\text{canon}(model_{id_i}, \theta^*)), \quad (2.31a)$$

$$policy_{hash_i} = H(\text{canon}(P)), \quad (2.31b)$$

де $model_{id_i}$ - ідентифікатор версії моделі, $\theta^* = (w^*, b_0^*)$ - оптимальні параметри моделі, P - множина політик реагування, визначена у (2.1), $\text{canon}(\cdot)$ - функція канонізації, визначена у (2.5), а $H(\cdot)$ - хеш-функція, визначена у (2.6).

Для практичного застосування будь-яке рішення системи повинно бути однозначно прив'язане не лише до події, а й до конкретної версії моделі та правил реагування, які використовувалися під час оцінювання. Аудитні записи групуються у блоки незмінного журналу:

$$B_k = (k, ts_k, prev_k, M_k, root_k), \quad (2.32)$$

де $B_k \in k$ - тим блоком журналу, $N \ni k$ - його індексом, ts_k - часом формування блоку, $prev_k$ - хешем попереднього блоку, M_k - множиною записів, що входять до поточного блоку, а $root_k$ - кореневий хеш ієрархічного дерева хешів, побудованого

для множини аудитних записів M_k . Множина аудитних записів поточного блоку визначається як:

$$M_k = \{a_{k,1}, a_{k,2}, \dots, a_{k,m_k}\}, \quad (2.33)$$

де $a_{k,j}$ - j -й аудитний запис у k -му блоці, сформований відповідно до (2.31), $j = \overline{1, m_k}$, а m_k - кількість записів у поточному блоці. Кількість записів m_k не фіксується наперед і визначається часовим інтервалом формування блоку. Для побудови ієрархічного дерева хешів кожний аудитний запис спочатку хешується:

$$l_{k,j} = H(a_{k,j}), \quad (2.34)$$

де $l_{k,j}$ - хеш j -го аудитного запису в k -му блоці. Кореневий хеш дерева визначається як:

$$root_k = \text{RootHash}(l_{k,1}, l_{k,2}, \dots, l_{k,m_k}) \quad (2.35)$$

де $\text{RootHash}(\cdot)$ - функція побудови кореневого хешу ієрархічного дерева хешів на основі хешів аудитних записів блоку. $\text{RootHash}(\cdot)$ реалізується як ітеративне попарне хешування елементів до отримання одного кореневого значення; у разі непарної кількості елементів останній елемент дублюється. Хеш поточного блоку визначається формулою:

$$bh_k = H(k|ts_k|prev_k|root_k), \quad (2.36)$$

де bh_k є хешем блоку, а символ $|$ позначає операцію конкатенації полів. Будь-яка зміна хоча б одного елемента в блоці або в послідовності блоків призводить до зміни bh_k , що забезпечує виявлюваність підміни даних. У підсумку журнал $\Lambda = B_1, B_2, \dots, B_K$ набуває властивості незмінності, а система отримує можливість перевірити не лише окремі події, а й цілісність усієї історії рішень.

У формалізованому поданні модель ІКДЦ задає такий узагальнений конвеєр обробки, спочатку система отримує критичну подію e_t , після чого формує її канонічне подання c_t , обчислює хеш h_t і цифровий підпис sig_t . Далі на основі події та її контексту формується вектор ознак x_t , обчислюється оцінка ризику p_t і визначається клас рішення s_t . Після цього формується аудитний запис a_t , який додається до поточного блоку B_k , а зв'язок блоків перевіряється через значення $prev_k$ і bh_k .

Таким чином, формалізовано модель інтегрованого контуру довіри й цілісності у вебсистемі, яка поєднує кортежне подання критичних подій, графове представлення їх контексту, оцінювання ризику, прийняття рішень та криптографічно захищене журналювання, це дозволило розробити модель ІКДЦ, що ґрунтується на кортежно-графовому поданні критичних подій і криптографічних принципах їх верифікації.

2.2. Метод блокчейн-верифікованого журналювання критичних подій і контролю доступу у вебсистемах

Метод блокчейн-верифікованого журналювання критичних подій і контролю доступу у вебсистемах розроблено на основі моделі ІКДЦ, основою методу є криптографічно зв'язаний ланцюг подій, у якому кожна критична дія подається як формалізована подія моделі ІКДЦ, проходить канонізацію, хешування, підписування, включення до аудитного запису та подальше групування у блок незмінного журналу відповідно до (2.2), (2.5)-(2.8), (2.31)-(2.36). Узагальнено метод подається як відображення:

$$M_{BVJCA}(e_t) = (d_t, a_t, B_k, v_t), \quad (2.37)$$

де M_{BVJCA} - метод блокчейн-верифікованого журналювання критичних подій і контролю доступу, e_t - критична подія вебсистеми, подана відповідно до моделі ІКДЦ, d_t - рішення щодо доступу, a_t - аудитний запис, сформований відповідно до (2.31), B_k - блок незмінного журналу, сформований відповідно до (2.32)-(2.36), v_t - результат верифікації запису та його включення до журналу. У межах методу подія доступу розглядається як спеціалізований випадок критичної події e_t , у якій корисне навантаження містить запитувану дію, розраховані показники доступу та результат прийняття рішення:

$$payload_t^{access} = (A_t, \{FLAG_t(L)\}_{L=1}^N, CFLAG_t, \tau_t, d_t), \quad (2.38)$$

де $payload_t^{access}$ - корисне навантаження події доступу, A_t - запитувана дія, $FLAG_t(L)$ - показник доступу користувача на рівні L , $CFLAG_t$ - кумулятивний показник доступу,

τ_t - порогове значення доступу; d_t - результат рішення щодо надання або обмеження доступу.

$$A_t = (op_t, res_t), \quad (2.39)$$

де A_t - запитувана дія в момент часу t , op_t - тип операції (перегляд, створення, зміна, видалення або виконання адміністративної функції), а res_t - об'єкт вебсистеми, до якого запитується доступ.

Для кожного рівня доступу L визначається частка функцій, які доступні користувачу порівняно із загальною кількістю функцій цього рівня:

$$FLAG_t(L) = \frac{|F_{u_t,L}^{acc}|}{|F_L^{all}|}, \quad (2.40)$$

де $F_{u_t,L}^{acc}$ - множина функцій рівня L , доступних користувачу u_t , F_L^{all} - повна множина функцій, визначених для рівня L , $|\cdot|$ - кількість елементів множини. Значення $FLAG_t(L)$ належить інтервалу $[0,1]$, де 0 означає відсутність доступу до функцій рівня, а 1 - повний доступ.

Для отримання узагальненої оцінки доступу обчислюється середнє значення показників $FLAG_t(L)$ за всіма рівнями доступу:

$$CFLAG_t = \frac{1}{N} \sum_{L=1}^N FLAG_t(L), \quad (2.41)$$

де $CFLAG_t$ - кумулятивний показник доступу для поточного запиту, а N - кількість рівнів доступу, які враховуються під час прийняття рішення.

Порогове значення доступу визначається через частку функцій, необхідних для виконання запитуваної дії A_t :

$$\tau_t = \frac{1}{N} \sum_{L=1}^N \frac{|F_{A_t,L}^{req}|}{|F_L^{all}|}, \quad (2.42)$$

де $\tau_t \in [0,1]$ - порогове значення доступу для поточного запиту, N - кількість рівнів доступу. Для кожного рівня доступу L повна множина функцій задається як:

$$F_L^{all} = f_{L,1}, f_{L,2}, \dots, f_{L,m_L}, \quad (2.43)$$

де $f_{L,j}$ - j -та функція цього рівня, а $m_L = |F_L^{all}|$ - загальна кількість функцій рівня L . Для визначення того, чи потрібна конкретна функція для виконання запитуваної дії A_t , використовується індикатор:

$$\rho(A_t, f_{L,j}) = \begin{cases} 1, & \text{якщо функція } f_{L,j} \text{ необхідна для виконання } A_t \\ 1, & \text{якщо функція } f_{L,j} \text{ не потрібна для виконання } A_t \end{cases}, \quad (2.44)$$

де $\rho(A_t, f_{L,j})$ визначається політикою доступу P , тобто набором правил, які встановлюють, які функції потрібні для виконання певної дії. Кількість функцій рівня L , необхідних для виконання дії A_t , визначається як:

$$|F_{A_t, L}^{req}| = \sum_{j=1}^{m_L} \rho(A_t, f_{L,j}). \quad (2.45)$$

Такий підхід дозволяє визначати поріг не довільно, а відповідно до складності та вимог запитуваної дії. Рішення щодо доступу приймається за пороговим правилом:

$$d_t = \begin{cases} ALLOW, & CFLAG_t \geq \tau_t \\ DENY, & CFLAG_t < \tau_t \end{cases}, \quad (2.46)$$

де d_t - результат прийняття рішення. Якщо кумулятивний показник доступу не менший за порогове значення, доступ надається, якщо менший - доступ обмежується або відхиляється.

Після прийняття рішення подія доступу включається до загального контуру ІКДЦ, тобто формується її канонічне подання відповідно до (2.5), обчислюється хеш відповідно до (2.6), формується цифровий підпис відповідно до (2.7), перевірка підпису виконується відповідно до (2.8), а результат додається до аудитного запису та блоку журналу згідно з (2.31)-(2.36). Результат верифікації запису визначається як:

$$v_t = \begin{cases} 1, & \text{якщо виконано умови перевірки підпису, Merkle-кореня та зв'язку блоків,} \\ 0, & \text{якщо хоча б одна з умов перевірки порушена,} \end{cases}, \quad (2.47)$$

де $v_t = 1$ означає, що запис події, його підпис і включення до блоку журналу є коректними, а $v_t = 0$ свідчить про порушення цілісності, автентичності або зв'язності журналу. Умови перевірки реалізуються на основі формул (2.8), (2.35) і (2.36).

Отримані показники застосовуються в процедурі прийняття рішення щодо надання/відмови в доступі (Рис. 2.1), а також повинні бути включені до складу атрибутів події, що фіксується у журналі, оскільки саме вони пояснюють підстави

прийнятого рішення. Послідовність етапів методу блокчейн-верифікованого журналювання критичних подій і контролю доступу у вебсистемах, проходить від надходження критичної події, аналізу профілю користувача та обчислення показників $FLAG_t(L)$, $CFLAG_t$ і τ_t , до прийняття рішення щодо доступу, формування події доступу, її криптографічної обробки, включення до незмінного журналу та подальшої верифікації (Рис. 2.1).

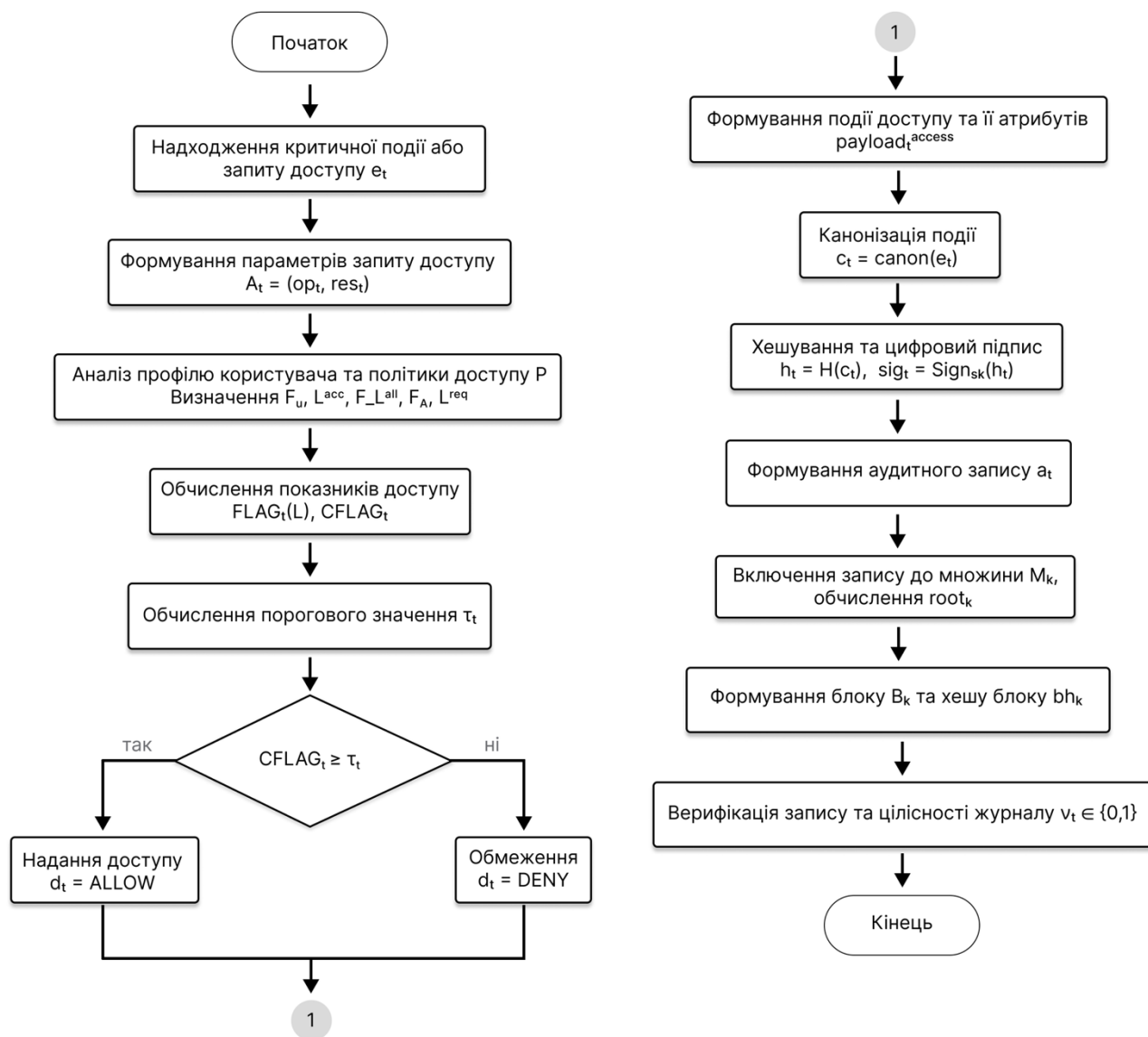


Рис. 2.1. Блок-схема методу блокчейн-верифікованого журналювання критичних подій і контролю доступу у вебсистемах

Контроль доступу в межах методу блокчейн-верифікованого журналювання критичних подій і контролю доступу у вебсистемах реалізується як послідовність взаємопов'язаних етапів, що охоплюють аналіз профілю користувача, обчислення

показників доступу, прийняття рішення щодо надання або обмеження доступу, фіксацію результату як критичної події та його подальше незмінне журналювання (Рис. 2.3).

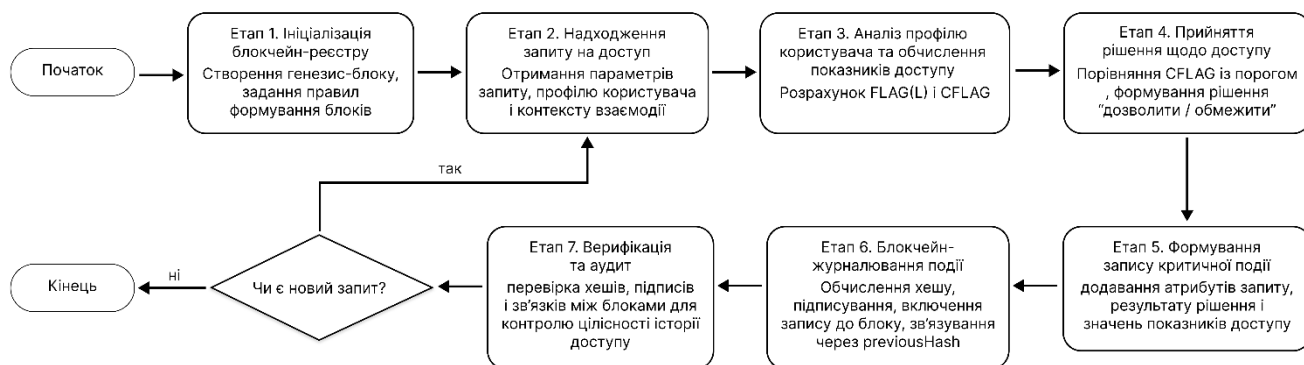


Рис. 2.3. Етапи алгоритму контролю доступу в межах методу блокчейн-верифікованого журналювання критичних подій і контролю доступу у вебсистемах

Таким чином, вперше розроблено метод блокчейн-верифікованого журналювання критичних подій і контролю доступу у вебсистемах, що ґрунтується на моделі ІКДЦ та теорії криптографічно зв'язаного ланцюга подій, який поєднує канонічне подання критичної події, хешування, цифровий підпис, блокчейн-верифіковане журналювання та порогове правило прийняття рішення щодо доступу. Це дозволяє фіксувати не лише факт виконання дії, а й підстави прийнятого рішення, зокрема значення $FLAG$, $CFLAG$, порогове значення τ_t та результат d_t .

2.3. Адаптація методу блокчейн-верифікованого журналювання критичних подій і контролю доступу для перевірки SQL-операцій та аудитного аналізу

Адаптація методу блокчейн-верифікованого журналювання критичних подій і контролю доступу у вебсистемах у межах моделі ІКДЦ орієнтована на формалізований опис можливостей його застосування для фіксації, перевірки та подальшого аудиту подій, що безпосередньо впливають на цілісність даних, коректність функціонування сервісів і безпеку взаємодії користувачів з вебресурсами.

До таких подій належать SQL-операції, рішення щодо надання або обмеження доступу, зміни параметрів безпеки та інші дії, для яких важливими є простежуваність, незмінність запису та можливість ретроспективної верифікації. У межах реалізації кожна критична подія проходить послідовність етапів перевірки, після чого фіксується в блокчейн-журналі разом із контекстом виконання та результатом обробки.

Одним із прикладних напрямів реалізації методу є контроль SQL-операцій як критичних подій вебсистеми. SQL-ін'єкції становлять небезпеку через можливість прихованої модифікації структури запиту, несанкціонованого доступу до даних та порушення цілісності інформації. Тому в межах реалізації методу кожен SQL-запит розглядається не лише як засіб звернення до бази даних, а як подія, що підлягає перевірці, журналюванню та подальшому аудиту. Алгоритм обробки SQL-запиту включає первинну перевірку синтаксичної коректності, швидку попередню фільтрацію, оцінювання схожості із відомими шаблонами, валідацію за правилами безпеки та фіксацію результату перевірки в незмінному журналі (Рис. 2.4).

Для швидкої попередньої фільтрації SQL-запитів у реалізації використовується фільтр Блума як компактна бітова структура, що дозволяє перевірити можливу належність запиту до множини раніше зафіксованих SQL-шаблонів без виконання повного порівняння з кожним шаблоном. Фільтр Блума у межах попередньої перевірки SQL-запитів можна подати як пару $BF = (B, \mathcal{H})$, де $B \in \{0,1\}^m$ - бітовий масив довжини m , а $\mathcal{H} = h_1, h_2, \dots, h_k$ - множина хеш-функцій, що відображають SQL-шаблон або SQL-запит у позиції бітового масиву. Додавання шаблону полягає у встановленні в одиницю всіх бітів, визначених відповідними хеш-функціями, а перевірка належності виконується шляхом логічного об'єднання значень цих бітів. Такий підхід забезпечує швидке відсікання запитів, які точно не належать до множини відомих шаблонів, але допускає хибнопозитивні спрацьовування, тому позитивний результат перевірки потребує подальшого уточнення за допомогою метрики схожості.

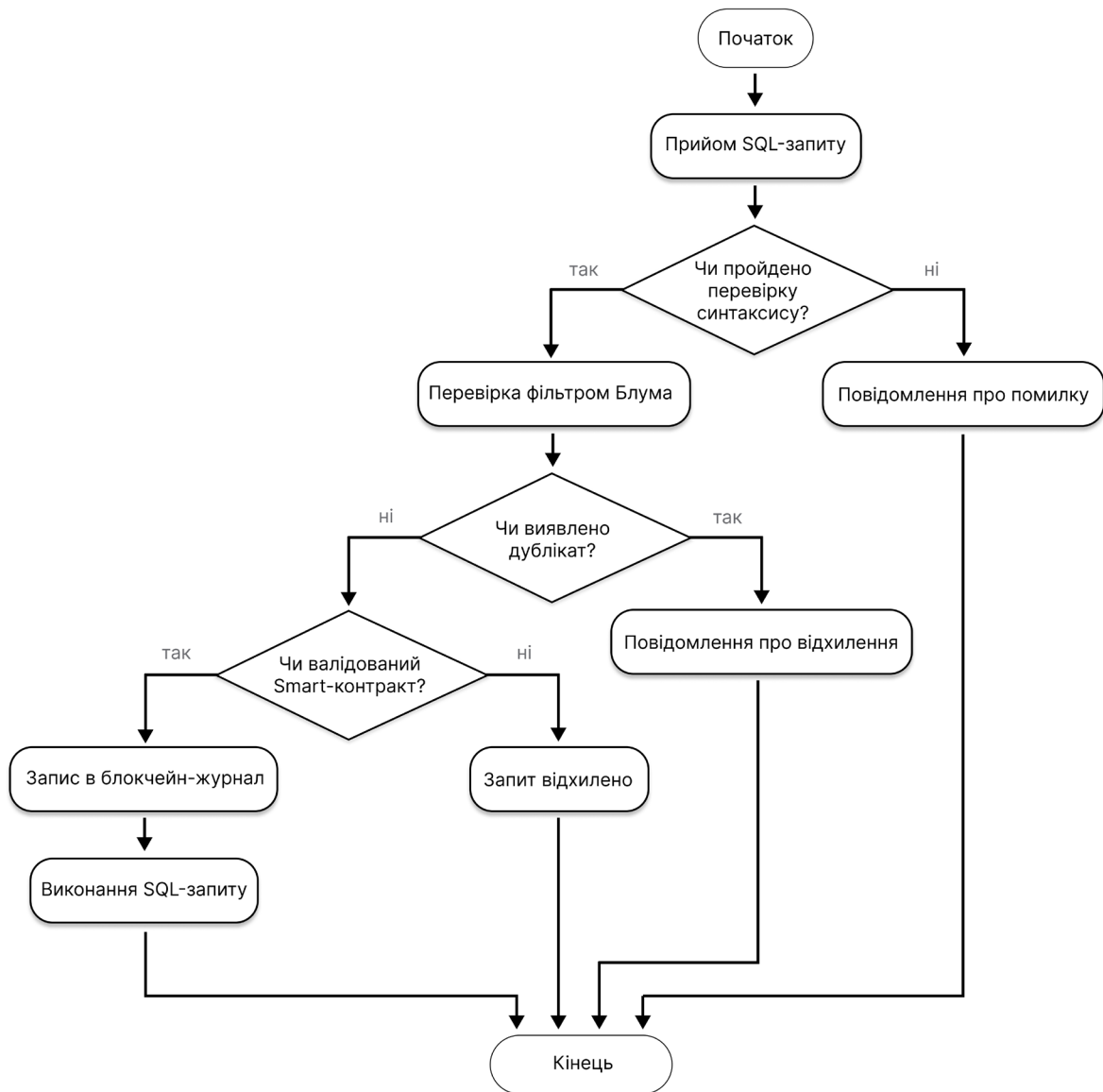


Рис. 2.4. Блок-схема реалізації контролю SQL-операцій у межах методу блокчейн-верифікованого журналювання критичних подій і контролю доступу у вебсистемах

Нехай $S = \{q_1, q_2, \dots, q_n\}$ - множина відомих SQL-шаблонів, $B = (b_1, b_2, \dots, b_m)$ - бітовий масив довжини m , де на початку $b_r = 0$ для всіх $r = \overline{1, m}$, а h_1, h_2, \dots, h_k - набір хеш-функцій, які відображають SQL-запит у позицію бітового масиву:

$$h_j(q) = 1 + (H(q \parallel j) \bmod m), \quad j = 1, 2, \dots, k, \quad (2.48)$$

де $H(\cdot)$ - криптографічна або стабільна хеш-функція, \parallel - операція конкатенації, j - номер хеш-функції, а m - довжина бітового масиву, $k \in \mathbb{N}$ - кількість хеш-функцій,

що використовуються у фільтрі Блума. Додавання шаблону $q_s \in S$ до фільтра виконується встановленням у 1 усіх позицій, визначених хеш-функціями:

$$b_{h_j(q_s)} := 1. \quad (2.49)$$

Після заповнення фільтра перевірка нового SQL-запиту q виконується за умовою:

$$BF(q) = \bigwedge_{j=1}^k b_{h_j(q)}, \quad (2.50)$$

де $BF(q)$ - результат перевірки запиту у фільтрі Блума, $b_{h_j(q)}$ - значення біта з індексом, отриманим за j -тою хеш-функцією, а \bigwedge означає логічну операцію “і” для всіх перевірюваних позицій. Якщо $BF(q) = 0$, то запит точно не належить до множини відомих шаблонів. Якщо $BF(q) = 1$, то запит може належати до цієї множини і передається на поглиблену перевірку, оскільки фільтр Блума допускає хибнопозитивні спрацьовування.

Після попередньої фільтрації виконується поглиблена перевірка схожості SQL-запиту з раніше зафіксованими зразками. Для цього використовується метрика Жаккара, яка дозволяє порівнювати запити на рівні множин ознак або токенів. У моделі вектори X та Y подаються як бінарні вектори ознак SQL-запитів, де кожен елемент вказує на наявність або відсутність певного токена, ключового слова чи конструкції. Оцінювання схожості здійснюється за формулою:

$$Jaccard(X, Y) = \frac{\sum_{i=1}^n X_i Y_i}{\sum_{i=1}^n X_i^2 + \sum_{i=1}^n Y_i^2 - \sum_{i=1}^n X_i Y_i}, \quad (2.51)$$

де X_i і Y_i - елементи векторів ознак двох SQL-запитів, $\sum_{i=1}^n X_i Y_i$ - число спільних ознак, а знаменник відображає загальний обсяг ознак обох запитів без подвійного врахування спільної частини. Значення, близьке до 1, означає високу схожість запитів, тоді як значення, близьке до 0, вказує на низьку схожість. Поєднання фільтра Блума з метрикою Жаккара дозволяє збалансувати швидкодію попередньої перевірки та точність виявлення модифікованих ін'єкційних шаблонів.

Після завершення перевірок SQL-запит або блокується, або допускається до виконання. Результат прийнятого рішення разом із контекстом запиту, службовими

атрибутами перевірки та ознаками схожості фіксується в блокчейн-журналі. У такий спосіб журнал містить не лише факт виконання чи відхилення SQL-операції, а й підстави прийнятого рішення. Це підвищує доказовість аудиту та дозволяє відтворити послідовність дій під час аналізу інцидентів або перевірки коректності застосування політик безпеки. Схема реалізації цього процесу (Рис. 2.4) відображає логіку багаторівневої обробки SQL-операцій у межах запропонованого методу.

Хеш поточного блоку в підсистемі аудитного контролю обчислюється відповідно до формули (2.36) моделі ІКДЦ, при цьому множина транзакцій доступу розглядається як множина аудитних записів поточного блоку M_k , визначена у (2.33), а їх включення до блоку перевіряється через хеші записів і корінь ієрархічного дерева хешів відповідно до (2.34)-(2.35). Криптографічна функція $H(\cdot)$ визначена у (2.6) і в реалізації відповідає алгоритму хешування SHA-256. Отже, зміна будь-якої транзакції доступу, аудитного запису або зв'язку з попереднім блоком призводить до зміни хешу блоку bh_k , що робить приховану модифікацію історії доступу виявлюваною.

Для розширення можливостей аудитного моніторингу в реалізацію методу включено аналітичний модуль, що оцінює нерівномірність розподілу доступу до ресурсів між користувачами. Такий аналіз не замінює базовий алгоритм контролю доступу, а виступає допоміжним засобом виявлення нетипових сценаріїв активності. Для кількісного оцінювання нерівномірності використовується коефіцієнт Джині:

$$G = \frac{\sum_{i=1}^{N_u} \sum_{j=1}^{N_u} |x_i - x_j|}{2N_u^2 \bar{r}}, \quad (2.52)$$

де N_u - кількість користувачів, x_i - кількість звернень або подій доступу i -го користувача за вибраний період, \bar{r} - середнє значення кількості звернень:

$$\bar{r} = \frac{1}{N_u} \sum_{i=1}^{N_u} x_i. \quad (2.53)$$

Зростання значення G свідчить про посилення концентрації доступу в обмеженої групи користувачів, що в контексті безпеки буде індикатором зловживання привілеями, компрометації облікового запису або помилок у розмежуванні прав.

Алгоритм роботи підсистеми контролю доступу та аудитного аналізу, поданий на Рис. 2.6, відображає послідовність обробки події доступу від моменту формування користувацького запиту до прийняття рішення щодо коригування політик безпеки. Після надходження запиту система формує транзакцію доступу, у якій фіксуються параметри звернення до вебресурсу, зокрема ідентифікатор користувача, ресурс доступу, час виконання операції, тип запиту та супровідні контекстні атрибути. Сформована транзакція додається до поточного блоку, після чого виконується перевірка її синтаксичної коректності. У разі успішного проходження перевірки блок включається до ланцюга, а для нового блоку обчислюється хеш, що забезпечує криптографічний зв'язок із попередніми записами та виявлюваність подальших змін.

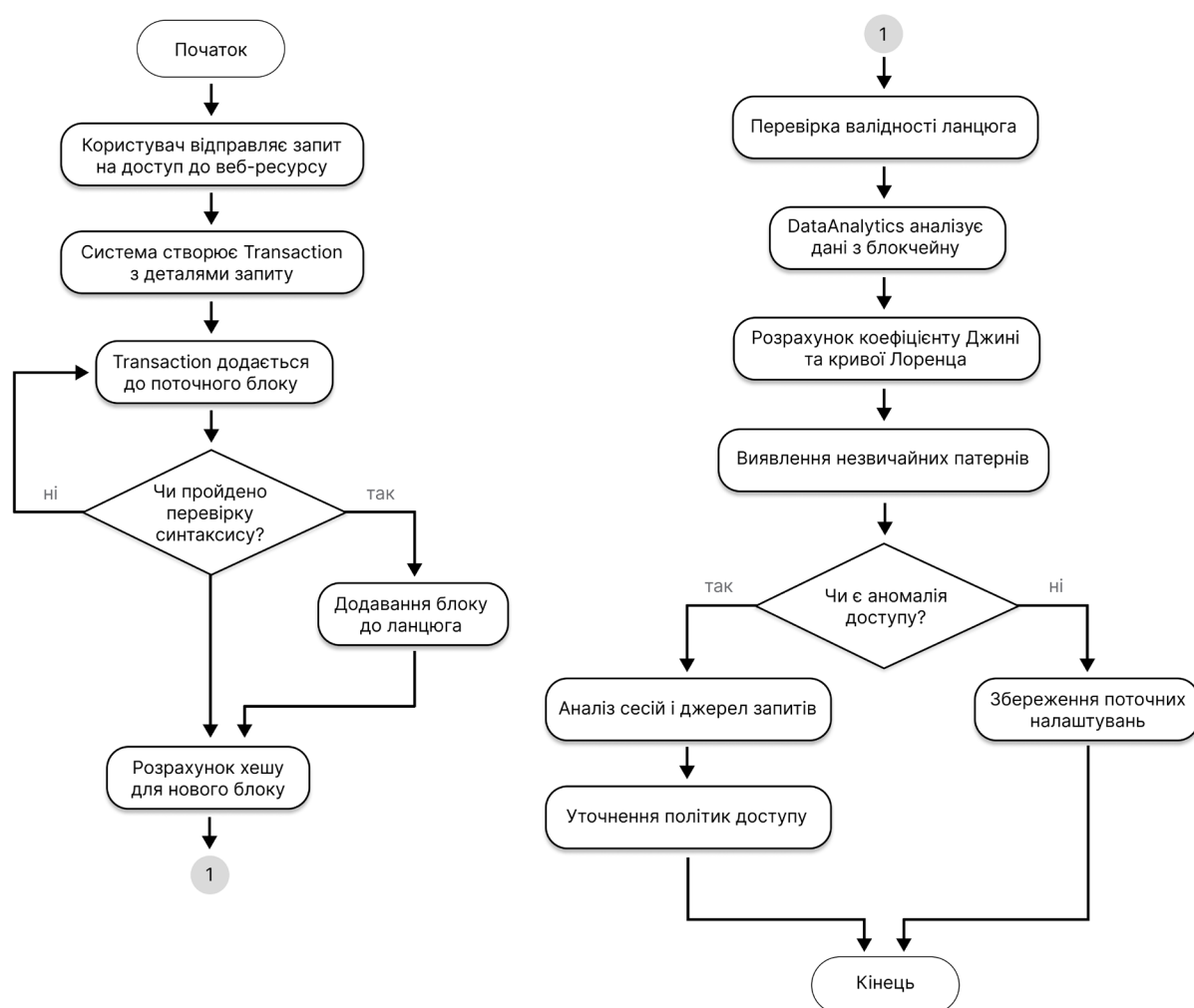


Рис. 2.6. Блок-схема аудитного моніторингу подій доступу в межах методу блокчейн-верифікованого журналювання критичних подій і контролю доступу у вебсистемах

Після формування блоку виконується перевірка валідності ланцюга, що передбачає контроль цілісності хеш-зв'язків і коректності включених записів. Накопичені дані передаються до аналітичного модуля «DataAnalytics», у якому здійснюється розрахунок коефіцієнта Джині та кривої Лоренца для оцінювання нерівномірності розподілу доступів між користувачами. На основі цих показників визначаються нетипові патерни активності, пов'язані з надмірною концентрацією звернень, аномальною частотою доступу або нестандартною поведінкою окремих облікових записів. У разі відсутності ознак аномалії поточні налаштування політик безпеки зберігаються без змін. Якщо аномалію виявлено, здійснюється додатковий аналіз сесій і джерел запитів, за результатами якого уточнюються політики доступу. Таким чином, наведена схема формалізує замкнений цикл контролю: реєстрацію події доступу, її незмінне журналювання, перевірку цілісності ланцюга, аналітичне оцінювання накопичених даних і прийняття рішення щодо подальшого налаштування правил доступу.

Отже, адаптація методу блокчейн-верифікованого журналювання критичних подій і контролю доступу до сценаріїв перевірки SQL-операцій, фіксації рішень доступу та аудитного аналізу активності користувачів показала, що поєднання попередньої фільтрації SQL-запитів, оцінювання їх схожості з відомими шаблонами, незмінного журналювання, хешування, цифрового підпису та перевірки зв'язності блоків дозволяє не лише реєструвати критичні події, а й формувати доказову основу для подальшої перевірки.

Висновки до розділу 2

За результатами проведеного дослідження встановлено, що забезпечення довіри й цілісності у вебсистемах потребує узгодженого подання критичних подій, їх контексту, результатів оцінювання, рішень системи безпеки та аудитних записів у межах єдиного формального контуру.

У результаті дослідження вперше розроблено модель інтегрованого контуру довіри й цілісності у вебсистемі, що ґрунтується на кортежно-графовому поданні

критичних подій і криптографічних принципах їх верифікації. Запропонована модель за рахунок поєднання незмінного журналювання критичних подій, формалізованого подання зв'язків між вебформами, SQL-операціями, рішеннями аналітичного модуля та політиками реагування забезпечує єдине інформаційне середовище для контролю цілісності даних, простежуваності подій, аудитної перевірки та відтворюваності рішень у вебсистемі.

Вперше розроблено метод блокчейн-верифікованого журналювання критичних подій і контролю доступу у вебсистемах, що ґрунтується на розробленій моделі ІКДЦ та теорії криптографічно зв'язаного ланцюга подій із хешуванням, цифровим підписом і пороговим правилом прийняття рішення щодо доступу. Запропонований метод дозволяє фіксувати критичні події, рішення щодо доступу та підстави їх прийняття в незмінному журналі, що зменшує ризик прихованої модифікації інформації, підвищує доказовість журналів і посилює контроль цілісності даних під час розслідування інцидентів.

Проведено адаптацію методу блокчейн-верифікованого журналювання критичних подій і контролю доступу до сценаріїв перевірки SQL-операцій, подій доступу та аудитного аналізу активності користувачів. Для цього в межах запропонованого підходу використано попередню фільтрацію SQL-запитів за допомогою фільтра Блума, оцінювання схожості запитів за метрикою Жаккара, а також аналіз нерівномірності розподілу доступу із застосуванням кривої Лоренца та коефіцієнта Джині як допоміжних аналітичних засобів. Це розширює можливості методу щодо виявлення нетипових сценаріїв активності, підтримки аудитних перевірок і коригування політик безпеки.

Таким чином, у другому розділі сформовано модельно-методичний базис для забезпечення довіри й цілісності у вебсистемах шляхом формалізованого подання критичних подій, їх контексту, результатів оцінювання, рішень щодо доступу та аудитних записів. Запропоновані модель ІКДЦ і метод блокчейн-верифікованого журналювання визначають структуру, логіку та криптографічні умови фіксації подій, перевірки цілісності даних і підтвердження обґрунтованості прийнятих рішень.

РОЗДІЛ 3. МЕТОД ГРАФОВО-НЕЙРОМЕРЕЖЕВОГО ВИЯВЛЕННЯ ВЕБСПАМУ ТА ПІДОЗРІЛОЇ АКТИВНОСТІ У ВЕБСИСТЕМАХ

3.1. Формування даних і ознак для аналізу вебконтенту та поведінки користувачів

Інтенсивне зростання електронної комерції, дистанційних сервісів та онлайн-фінансових операцій супроводжується масштабуванням загроз, серед яких окреме місце займають вебспам та шахрайські дії, що маскуються під легітимну активність користувачів. У межах задачі підвищення безпеки вебзастосунків ключовим стає формування репрезентативного набору даних і ознак, які дозволяють алгоритмам машинного навчання розрізняти нормальну та ризикову поведінку, а також виявляти нетипові транзакційні сценарії.

У межах моделі ІКДЦ та методу графово-нейромережевого виявлення вебспаму та підозрілої активності у вебсистемах подію (транзакцію) розглянемо як набір параметрів, що відображають як фінансові характеристики, так і поведінковий контекст. До базових атрибутів віднесемо географічне положення користувача, суму платежу, час здійснення операції та тип платіжної картки. Такий набір є достатнім для побудови первинної моделі ризику і водночас є типовим для більшості платіжних сценаріїв, що полегшує інтеграцію в існуючі системи. Зазначені атрибути інтерпретуються як ознаки, що потенційно корелюють із шахрайськими практиками.

Підготовка даних включає послідовність операцій, спрямованих на підвищення якості навчальної вибірки та зменшення впливу шумів. На етапі очищення виконуються перевірки коректності значень, вилучення або виправлення аномальних/порожніх записів, уніфікація форматів представлення часу та категоріальних атрибутів. Окремо враховується необхідність стандартизації одиниць виміру та форматів, оскільки неоднорідність журналів та інтеграційних даних є типовою проблемою для вебсистем. Нормалізація ознак використовується для приведення параметрів до співставної шкали, що критично для коректного навчання

ряду моделей. Частина ознак в результаті нормалізації визначається як значення з інтервалу $(0;1)$, що забезпечує стабільність подальших обчислень і дозволяє інтерпретувати інтегральний показник як рівень ризику в уніфікованій шкалі. Підготовлені дані організовуються у структурі DataFrame, після чого виконуються процедури кодування та масштабування, необхідні для подачі до моделі графово-нейромережевого виявлення вебспау та підозрілої активності у вебсистемах. Для задачі оцінювання ризику транзакцій використано багатофакторний підхід, у якому кожний параметр події відповідає окремому компоненту ризику. Тоді інтегральний показник ризику транзакції, який використовується як критерій прийняття рішення щодо безпечності операції, визначається за формулою

$$R = \alpha_P R_P(P) + \alpha_C R_C(C) + \alpha_T R_T(T) + \alpha_K R_K(K), \quad (3.1)$$

де $\alpha_P, \alpha_C, \alpha_T, \alpha_K$ - вагові коефіцієнти компонентів ризику, причому $\alpha_P + \alpha_C + \alpha_T + \alpha_K = 1, \alpha_i \in [0;1]$. У методі графово-нейромережевого виявлення вебспау та підозрілої активності у вебсистемах компонент $R_P(P)$ відображає ризик залежно від суми платежу, компонент $R_C(C)$ - ризик, пов'язаний із географічним положенням користувача, компонент $R_T(T)$ - ризик залежно від часу проведення операції, а компонент $R_K(K)$ - ризик, пов'язаний із типом платіжної картки. Таке подання дозволяє врахувати вплив різнорідних факторів у межах єдиної шкали оцінювання.

Ризик $R_P(P)$, пов'язаний із сумою платежу, визначається за формулою (2.12) вибирається так, щоб значення належало інтервалу $(0;1)$, за максимального очікуваного значення суми P_{max} коефіцієнт задається як:

$$a = \frac{1}{\ln(P_{max}+1)}. \quad (3.2)$$

Таке логарифмічне перетворення забезпечує нелінійне зростання ризику зі збільшенням суми платежу та зменшує вплив одиничних надвеликих значень.

Географічний компонент ризику визначається як таблично-розрахункове відображення країни або регіону транзакції у нормалізоване значення ризику:

$$R_C(C) = w^{geo}(C), \quad (3.3)$$

де C - країна або регіон виконання транзакції, $w^{geo}(C) \in [0,1]$ - нормалізований географічний коефіцієнт ризику для відповідного регіону. На відміну від вагового коефіцієнта α_C , який визначає внесок географічної компоненти у загальний ризик у

формулі (3.1), значення $w^{geo}(C)$ характеризує ризиковість саме конкретної географічної зони. Для визначення $w^{geo}(C)$ використовується статистика історичних транзакцій або навчальної вибірки. Нехай n_C - загальна кількість транзакцій, зафіксованих для регіону C , а z_C - кількість транзакцій цього регіону, які були позначені як ризикові або шахрайські. Тоді емпірична частка ризикових транзакцій для регіону визначається як:

$$\hat{p}_C = \frac{z_C}{n_C}. \quad (3.4)$$

Оскільки для окремих регіонів кількість транзакцій може бути малою, для уникнення нестійких оцінок доцільно застосовувати згладжену оцінку:

$$\hat{p}_C^* = \frac{z_C + \beta \bar{p}}{n_C + \beta}, \quad (3.5)$$

де \bar{p} - середня частка ризикових транзакцій у всій вибірці, а $\beta > 0$ - параметр згладжування, який зменшує вплив малих вибірок. Отримане значення \hat{p}_C^* нормалізується до інтервалу $[0,1]$ і використовується як $w^{geo}(C)$. Вищі значення $w^{geo}(C)$ відповідають регіонам, у яких історично спостерігається вища частка підозрілих або шахрайських транзакцій. У практичному застосуванні значення $w^{geo}(C)$ можуть зберігатися у вигляді довідкової таблиці географічних ризиків, яка формується на основі історичних журналів транзакцій або навчальної вибірки. Така таблиця для кожного регіону C містить кількість транзакцій n_C , кількість ризикових або шахрайських транзакцій z_C , згладжену частку ризику \hat{p}_C^* та нормалізований коефіцієнт $w^{geo}(C)$. У межах дослідження ця таблиця розглядається не як фіксований зовнішній довідник, а як результат обробки даних конкретної вебсистеми.

Часовий компонент ризику описується функцією на основі гауссівського розподілу:

$$R_T(T) = b_T \cdot e^{-\frac{(T - \mu_T)^2}{2\sigma_T^2}}, \quad (3.6)$$

де T - час транзакції, поданий у єдиній часовій шкалі, $b_T \in [0,1]$ - коефіцієнт максимальної інтенсивності часового ризику, μ_T - часовий центр підвищеного ризику, $\sigma_T > 0$ - параметр ширини часового інтервалу підвищеного ризику. Чим ближче значення T до μ_T , тим вищим є значення $R_T(T)$. Параметри μ_T та σ_T визначаються на

основі історичних транзакцій, позначених як ризикові або шахрайські. Нехай T_i - час i -ї транзакції, а $y_i \in 0,1$ - її мітка, де $y_i = 1$ відповідає ризиковій транзакції. Тоді центр часового ризику визначається як середній час ризикових транзакцій:

$$\mu_T = \frac{\sum_{i=1}^n y_i T_i}{\sum_{i=1}^n y_i}. \quad (3.7)$$

Параметр ширини часового інтервалу підвищеного ризику визначається як стандартне відхилення часу ризикових транзакцій відносно μ_T :

$$\sigma_T = \sqrt{\frac{\sum_{i=1}^n y_i (T_i - \mu_T)^2}{\sum_{i=1}^n y_i}}. \quad (3.8)$$

Коефіцієнт b_T визначає максимальне значення часового компонента ризику, оскільки при $T = \mu_T$ експоненційна частина дорівнює 1, а отже $R_T(\mu_T) = b_T$. Для нормалізованої шкали часовий компонент змінюється в межах $[0,1]$, а внесок часу в загальний ризик регулюється ваговим коефіцієнтом α_T у формулі (3.1).

Ризик, пов'язаний із типом платіжного інструмента, визначається як таблично-розрахункове відображення типу картки або платіжного засобу у нормалізоване значення ризику:

$$R_K(K) = w^{pay}(K), \quad (3.9)$$

де K - тип платіжного інструмента, наприклад дебетова картка, кредитна картка, віртуальна картка, передплачена картка або інший платіжний засіб, а $w^{pay}(K) \in [0,1]$ - нормалізований коефіцієнт ризику для відповідного типу платіжного інструмента. На відміну від вагового коефіцієнта α_K у формулі (3.1), який визначає внесок компонента $R_K(K)$ у загальний ризик, значення $w^{pay}(K)$ характеризує ризиковість саме конкретного типу платіжного інструмента.

Для визначення $w^{pay}(K)$ використовується статистика історичних транзакцій або навчальної вибірки. Нехай n_K - загальна кількість транзакцій, виконаних із використанням платіжного інструмента типу K , а z_K - кількість таких транзакцій, які були позначені як ризикові або шахрайські. Тоді емпірична частка ризикових транзакцій для типу платіжного інструмента визначається як:

$$\hat{p}_K = \frac{z_K}{n_K}. \quad (3.10)$$

Для зменшення впливу малих вибірок використовується згладжена оцінка:

$$\hat{p}_K^* = \frac{z_K + \beta \bar{p}}{n_K + \beta}, \quad (3.11)$$

де \bar{p} - середня частка ризикових транзакцій у всій вибірці, а $\beta > 0$ - параметр згладжування. Отримане значення \hat{p}_K^* нормалізується до інтервалу $[0,1]$ і використовується як $w^{pay}(K)$. Вищі значення $w^{pay}(K)$ відповідають типам платіжних інструментів, для яких в історичних даних спостерігається вища частка підозрілих або шахрайських транзакцій. У практичному застосуванні значення $w^{pay}(K)$ зберігатися у вигляді довідкової таблиці ризиків платіжних інструментів, яка формується на основі історичних журналів транзакцій або навчальної вибірки. Така таблиця для кожного типу K містить кількість транзакцій n_K , кількість ризикових або шахрайських транзакцій z_K , згладжену оцінку частки ризикових транзакцій \hat{p}_K^* та нормалізований коефіцієнт $w^{pay}(K)$. У межах дослідження ця таблиця розглядається не як фіксований зовнішній довідник, а як результат обробки даних конкретної вебсистеми.

Загальний ризик транзакції, обчислений як зважена сума компонентів у формулі (3.1), порівнюється з пороговим значенням τ_{TX} , де $\tau_{TX} \in [0,1]$. Це порогове значення визначає межу, після якої транзакція потребує додаткової перевірки. На відміну від порогів τ_1 та τ_2 , введених у моделі ІКДЦ для багатокласової класифікації подій відповідно до (2.26)-(2.30), у цьому випадку використовується бінарне правило для попереднього відокремлення безпечних і потенційно підозрілих транзакцій.

Для визначення τ_{TX} використовується розподіл значень інтегрального ризику на валідаційній вибірці:

$$R_{val} = \{R_1, R_2, \dots, R_n\}, \quad (3.12)$$

де R_{val} - множина значень інтегрального ризику транзакцій у валідаційній вибірці, R_i - значення ризику i -ї транзакції, обчислене за формулою (3.1), а n - кількість транзакцій у валідаційній вибірці. Середнє значення інтегрального ризику визначається як:

$$\mu_R = \frac{1}{n} \sum_{i=1}^n R_i, \quad (3.13)$$

де μ_R характеризує типовий рівень ризику транзакцій у валідаційній вибірці. Для врахування розкиду значень ризику обчислюється стандартне відхилення:

$$s_R = \sqrt{\frac{1}{n} \sum_{i=1}^n (R_i - \mu_R)^2}, \quad (3.14)$$

де s_R показує, наскільки значення ризику окремих транзакцій відхиляються від середнього рівня. Порогове значення ризику транзакції задається як:

$$\tau_{TX} = \min(1, \max(0, \mu_R + s_R)), \quad (3.15)$$

де функція $\min(1, \cdot)$ гарантує, що поріг не перевищує верхню межу нормалізованої шкали ризику $[0,1]$. Такий підхід дозволяє визначати поріг не довільно, а на основі фактичного розподілу ризикових оцінок у валідаційній вибірці. Рішення щодо транзакції задається пороговим правилом:

$$d_i^{TX} = \begin{cases} \text{SAFE}, & R_i < \tau_{TX}, \\ \text{SUSPECT}, & R_i \geq \tau_{TX}. \end{cases} \quad (3.16)$$

де d_i^{TX} - результат попередньої класифікації i -ї транзакції. Якщо $R_i < \tau_{TX}$, транзакція вважається безпечною; якщо $R_i \geq \tau_{TX}$, транзакція класифікується як потенційно підозріла та потребує додаткової перевірки.

Послідовність обробки транзакцій реалізована у вигляді блок-схеми (Рис. 3.1), де первинним етапом виступає підготовка та нормалізація даних, після чого кожна транзакція проходить через процедуру оцінювання. Результати аналізу зберігаються у форматі JSON разом з усіма параметрами транзакції та обчисленим ризиковим показником.

Агреговані історичні дані використовуються як основа для аналізу тенденцій, адаптації та подальшого вдосконалення методу графово-нейромережевого виявлення вебспау та підозрілої активності у вебсистемах. У контексті протидії вебспау це має принципове значення, оскільки спам- та шахрайські стратегії є адаптивними: зміщуються часові патерни, змінюються географічні маршрути, модифікуються поведінкові сценарії. Тому зазначений метод у дослідженні розглядається як такий, що підлягає регулярному оновленню на нових даних, а накопичений JSON-архів - як джерело для перевірки гіпотез, повторних експериментів та розширення набору ознак.



Рис. 3.1 Узагальнений алгоритм класифікації подій вебсистеми у методі графово-нейромережевого виявлення вебспаму

Запропонована схема формування та аналізу даних забезпечує гнучкість і масштабованість підходу, оскільки дає змогу окремо оновлювати процедури очищення, нормалізації, кодування ознак, розрахунку ризикових компонентів і збереження результатів аналізу. Такий модульний підхід полегшує подальшу

адаптацію методу графово-нейромережевого виявлення вебспаму та підозрілої активності до нових типів даних, змін у поведінкових сценаріях користувачів і появи нових шахрайських практик.

3.2. Метод графово-нейромережевого виявлення вебспаму та підозрілої активності у вебсистемах

Метод графово-нейромережевого виявлення вебспаму та підозрілої активності у вебсистемах ґрунтується на моделі ІКДЦ, у якій кожне подання через вебформу або інша критична подія розглядається не ізольовано, а як елемент множини пов'язаних подій. Теоретичною основою методу є багатопредставлене графове подання подій, де одна й та сама множина звернень аналізується через декілька типів зв'язків: контентну подібність, спільність технічного джерела, часову близькість і поведінкові ознаки. Такий підхід дозволяє враховувати не лише локальний зміст повідомлення, а й контекст його появи в загальному потоці подій вебсистеми. Узагальнено метод подається як відображення:

$$\mathcal{M}_{\mathcal{GN}}(E, \mathcal{G}, X; \Theta) = (p_v, c_v), \quad (3.17)$$

де $\mathcal{M}_{\mathcal{GN}}$ - метод графово-нейромережевого виявлення вебспаму та підозрілої активності, E - множина подій вебсистеми, \mathcal{G} - множина графових представлень подій, X - матриця ознак вузлів, Θ - множина параметрів графово-нейромережевої моделі, p_v - вектор імовірностей належності події v до класів; c_v - кінцевий клас події. Множина графових представлень задається як:

$$\mathcal{G} = G^{(1)}, G^{(2)}, \dots, G^{(R)}, \quad (3.18)$$

де R - кількість типів графових зв'язків між подіями. Кожне окреме представлення має вигляд:

$$G^{(r)} = (V, E^{(r)}), r = 1, \dots, R, \quad R \in \mathbb{N} \quad (3.19)$$

де V - спільна множина вузлів, що відповідають поданням через вебформи або іншим подіям вебсистеми, а $E^{(r)}$ - множина ребер у r -му графовому представленні. Одне представлення відображає подібність текстового вмісту повідомлень, друге - спільність IP-адреси або пристрою, третє - часову близькість подань, четверте -

подібність поведінкових сценаріїв. Ребро між двома подіями у r -му представленні формується за правилом:

$$(v, u) \in E^{(r)}, \text{ якщо } s^{(r)}(v, u) \geq \tau_r, \quad (3.20)$$

де $s^{(r)}(v, u) \in [0, 1]$ - значення подібності між подіями (v) та (u) за (r)-м типом зв'язку, а $\tau_r \in [0, 1]$ - поріг утворення ребра для відповідного представлення. Значення $s^{(r)}(v, u)$ визначається за текстовою подібністю, збігом технічних атрибутів, часовою близькістю або поведінковою схожістю. Поріг τ_r задається в конфігурації методу або визначається на валідаційній вибірці. Для кожного вузла $v \in V$ формується початковий ознаковий вектор:

$$x_v = \Phi(e_v, G_v) \in \mathbb{R}^d, \quad (3.21)$$

де x_v - вектор ознак події e_v , $\Phi(\cdot)$ - функція формування ознак, визначена в моделі ІКДЦ, G_v - локальний графовий контекст події, а d - кількість ознак. До складу x_v можуть входити текстові, структурні, часові, мережеві та поведінкові параметри подання. Початковий стан вузла в кожному графовому представленні задається його ознаковим вектором:

$$h_v^{(r,0)} = x_v. \quad (3.22)$$

На l -му шарі графової нейронної мережі для вузла v спочатку виконується агрегація представлень сусідніх вузлів:

$$m_v^{(r,l)} = \text{AGG} \left(h_u^{(r,l)} : u \in N_r(v) \right), \quad (3.23)$$

де $N_r(v)$ - множина сусідів вузла v у графовому представленні $G^{(r)}$, $h_u^{(r,l)}$ - приховане представлення сусіднього вузла u на шарі l , а $m_v^{(r,l)}$ - агрегований вектор сусіднього контексту. Для однозначності обчислення операцію агрегації можна задати як середнє значення представлень сусідніх вузлів:

$$m_v^{(r,l)} = \frac{1}{|N_r(v)|} \sum_{u \in N_r(v)} h_u^{(r,l)}. \quad (3.24)$$

Якщо $|N_r(v)| = 0$, тобто вузол не має сусідів у відповідному представленні, агрегаційний вектор приймається нульовим або замінюється власним представленням вузла. Це усуває невизначеність у випадку ізольованих подій.

Далі обчислюється проміжне представлення вузла:

$$u_v^{(r,l)} = W_{self}^{(r,l)} h_v^{(r,l)} + W_{nb}^{(r,l)} m_v^{(r,l)} + b^{(r,l)}, \quad (3.25)$$

де $W_{self}^{(r,l)}$ - матриця параметрів для власного представлення вузла, $W_{nb}^{(r,l)}$ - матриця параметрів для агрегованої інформації від сусідів, $b^{(r,l)}$ - вектор зсуву, а $u_v^{(r,l)}$ - проміжний вектор перед застосуванням нелінійної функції.

Новий стан вузла визначається як:

$$h_v^{(r,l+1)} = \varphi \left(u_v^{(r,l)} \right), \quad (3.26)$$

де $\varphi(\cdot)$ - нелінійна функція активації. Для практичного використання її можна задати як ReLU-функцію:

$$\varphi(z)_s = \max(0, z_s), \quad s = 1, \dots, d_h, \quad (3.27)$$

де z_s - s -й елемент вектора z , а d_h - розмірність прихованого представлення вузла. Така функція відсікає від'ємні значення та зберігає додатні, що дозволяє моделі формувати нелінійне представлення ознак і сусіднього контексту.

Після проходження L шарів графової нейронної мережі для кожного представлення отримуються кінцеві стани $h_v^{(1,L)}, h_v^{(2,L)}, \dots, h_v^{(R,L)}$. Для формування єдиного багатопредставленого опису події ці вектори об'єднуються:

$$z_v = \text{CONCAT} \left(h_v^{(1,L)}, h_v^{(2,L)}, \dots, h_v^{(R,L)} \right), \quad (3.28)$$

де z_v - підсумкове графове представлення події v , а $\text{CONCAT}(\cdot)$ - операція об'єднання векторів. Якщо кожне представлення $h_v^{(r,L)}$ має розмірність d_h , то z_v має розмірність $R \cdot d_h$. Класифікаційний шар формує вектор оцінок:

$$s_v = W_o z_v + b_o, \quad (3.29)$$

де s_v - вектор оцінок для класів, W_o - матриця параметрів класифікаційного шару, а b_o - вектор зсуву. Ймовірність належності події до класу c визначається функцією:

$$p_v^c = \frac{e^{s_v^c}}{\sum_{c' \in \mathcal{C}} e^{s_v^{c'}}}, \quad \mathcal{C} = \text{LEGIT}, \text{SUSPECT}, \text{SPAM}, \quad (3.30)$$

де p_v^c - імовірність належності події v до класу c , s_v^c - оцінка класифікаційного шару для цього класу, а \mathcal{C} - множина класів. Клас *LEGIT* відповідає легітимному зверненню, *SUSPECT* - події, що потребує додаткової перевірки адміністратором, а *SPAM* - шкідливому або спам-зверненню. Кінцевий клас події визначається за найбільшою імовірністю:

$$c_v = \arg \max_{c \in \mathcal{C}} p_v^c. \quad (3.31)$$

Параметри $W_{self}^{(r,l)}$, $W_{nb}^{(r,l)}$, $b^{(r,l)}$, W_o та b_o входять до множини параметрів моделі Θ визначаються під час навчання графово-нейромережевого класифікатора на розміченій вибірці подій. Для цього використовується крос-ентропійна функція втрат:

$$\mathcal{L}(\Theta) = -\frac{1}{|V_{train}|} \sum_{v \in V_{train}} \sum_{c \in \mathcal{C}} y_{v,c} \ln(p_v^c), \quad (3.32)$$

де V_{train} - множина навчальних вузлів, $y_{v,c} \in 0,1$ - істинна мітка вузла v для класу c , а p_v^c - імовірність, обчислена моделлю. Мінімізація $\mathcal{L}(\Theta)$ дозволяє налаштувати параметри моделі так, щоб підвищити правильність розрізнення легітимних, підозрілих і шкідливих звернень.

Ітераційний механізм поширення інформації по графу забезпечує накопичення контексту, а саме стан вузла поступово відображає не тільки його індивідуальні характеристики, а й закономірності в підграфі, з яким він пов'язаний. Це підвищує стійкість детекції в умовах, коли спамери модифікують тексти повідомлень, але зберігають повторювані поведінкові або технічні шаблони: часові серії, спільні джерела, повтори структур повідомлень чи групову синхронність подань.

У межах методу рішення *SUSPECT* виступає механізмом керованої перевірки: система не блокує потенційно легітимні повідомлення автоматично, а передає їх на розгляд адміністратору. Адміністратор може виправляти помилкові позначки для окремих подань, а ці уточнені приклади надалі використовуються для донавчання або періодичного перенавчання графово-нейромережевого класифікатора. Такий механізм спрямовано на зменшення частки хибнопозитивних спрацювань, коли

легітимне повідомлення помилково віднесено до спаму, і хибнонегативних спрацювань, коли спам не виявлено.

Застосування методу у вебсистемі передбачає його використання як окремого аналітичного контуру, до якого передаються дані вебформ для класифікації. Після отримання даних виконується попередня обробка, нормалізація та векторизація ознак, після чого сформовані представлення подаються до графово-нейромережевої моделі. Результат класифікації визначає подальшу обробку подання: легітимні повідомлення передаються у стандартний бізнес-процес, підозрілі маркуються для перевірки адміністратором, а спам-звернення блокуються або ізолюються (Рис. 3.2).

Для забезпечення прозорості та можливості подальшого аналізу всі вхідні дані, незалежно від результату класифікації, доцільно реєструвати у сховищі подій. Накопичені дані виконують дві функції: по-перше, підтримують аудит рішень системи й відстеження еволюції спам-патернів; по-друге, формують емпіричну базу для донавчання графово-нейромережевого класифікатора та перевірки його узагальнювальної здатності на нових вибірках. Ефективність запропонованого механізму доцільно контролювати через показники, що відображають як якість детекції, так і експлуатаційні характеристики, серед яких швидкість обробки подань, точність класифікації та частка хибних спрацювань. Регулярний моніторинг цих параметрів дозволяє своєчасно виявляти деградацію якості (наприклад, при зміні типових сценаріїв атак) і ініціювати корекцію навчання на актуальних даних. У підсумку, застосування GNN у задачі фільтрації вебспаму у вебформах забезпечує аналіз подань з урахуванням їхніх зв'язків та контексту, підвищує стійкість до модифікацій спам-тактик, а також створює основу для керованої адаптації методу графово-нейромережевого виявлення вебспаму та підозрілої активності у вебсистемах через адміністративний зворотний зв'язок.

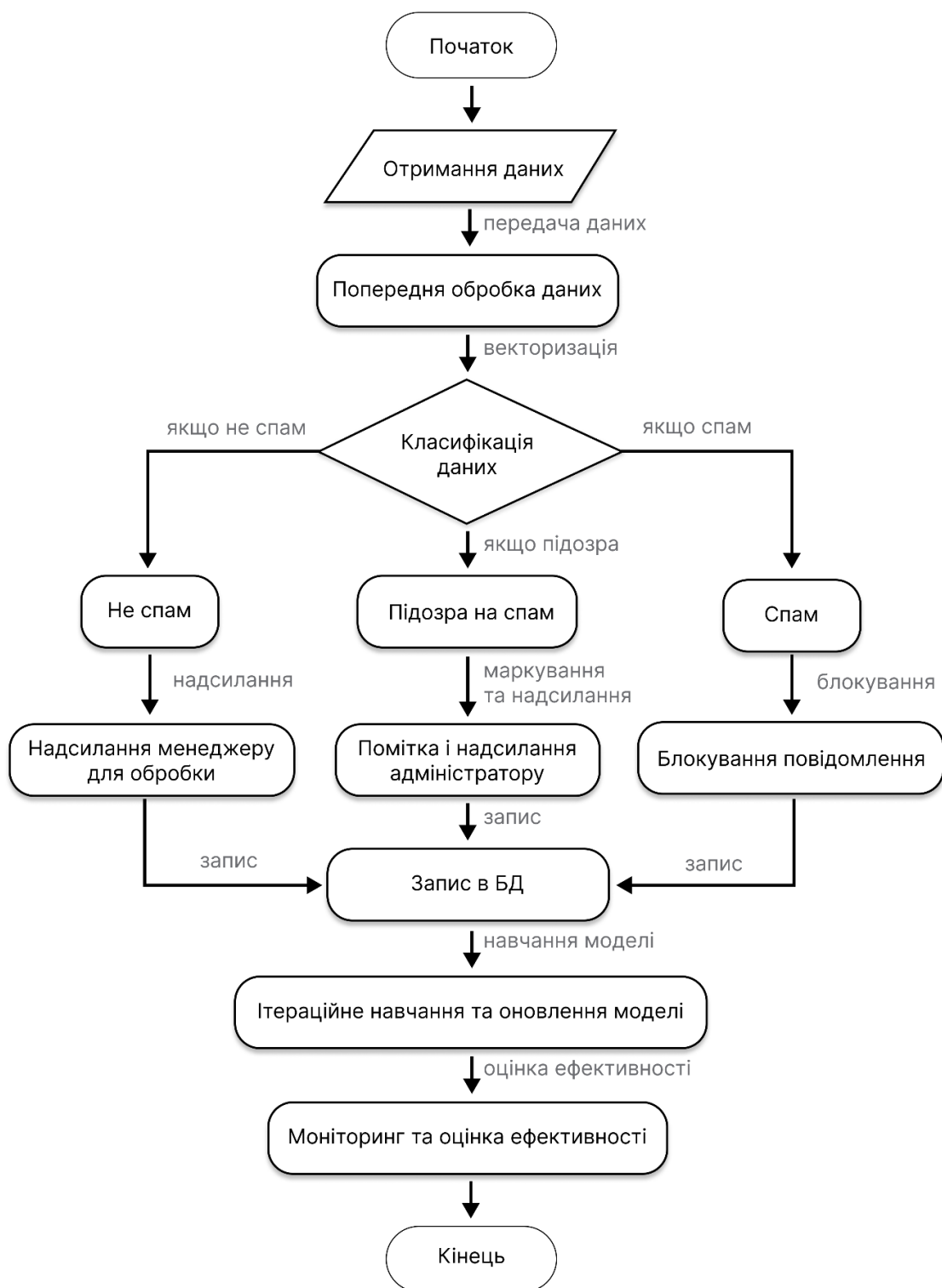


Рис. 3.2 Визначення класифікації даних у методі графово-нейромережевого виявлення вебспаму

Таким чином, вперше розроблено метод графово-нейромережевого виявлення вебспаму та підозрілої активності у вебсистемах, що ґрунтується на моделі ІКДЦ та багатопредставленому графовому описі подій, поданих через систему ознак

технічного, змістового, часово-поведінкового та контекстного характеру з урахуванням зв'язків між подіями й результатами аналітичного оцінювання. Це забезпечує розрізнення легітимних, підозрілих і шкідливих звернень, підвищення точності виявлення вебспаму та зменшення частки хибних спрацювань.

3.3. Адаптація методу графово-нейромережевого виявлення вебспаму та підозрілої активності до зміни шаблонів загроз, правил реагування та механізмів забезпечення цілісності

Забезпечення стійкого захисту вебформ від координованого вебспаму потребує поєднання двох взаємодоповнювальних контурів. Перший контур відповідає за інтелектуальну оцінку ризику відправки (ймовірність спаму) на основі контентних, поведінкових і структурних ознак. Другий контур гарантує цілісність і відтворюваність прийнятого рішення, а також регламентує операційні дії системи у відповідь на виявлену загрозу. Результати машинного навчання узгоджуються з механізмами забезпечення цілісності та з формалізованими правилами реагування. Вебспам у сучасних вебзастосунках характеризується відходом від примітивних текстових шаблонів до кампаній, що комбінують масові подачі форм, повторне використання структур полів, синхронні часові сплески, варіативність URL та технічне маскуванню. За таких умов підходи, що покладаються на один канал сигналів демонструють обмежену робастність до дрейфу розподілів і обфускацій. Практичне розгортання, крім точності, вимагає керованої асиметрії помилок (контроль хибних спрацювань), прозорості рішень для аудиту та підтримки приватності, зокрема шляхом анонімізації чутливих ідентифікаторів [135].

Схема методу графово-нейромережевого виявлення вебспаму та підозрілої активності у вебсистемах подається як конвеєр багатопредставленого графового моделювання для виявлення вебспаму у вебформах (Рис. 3.4).

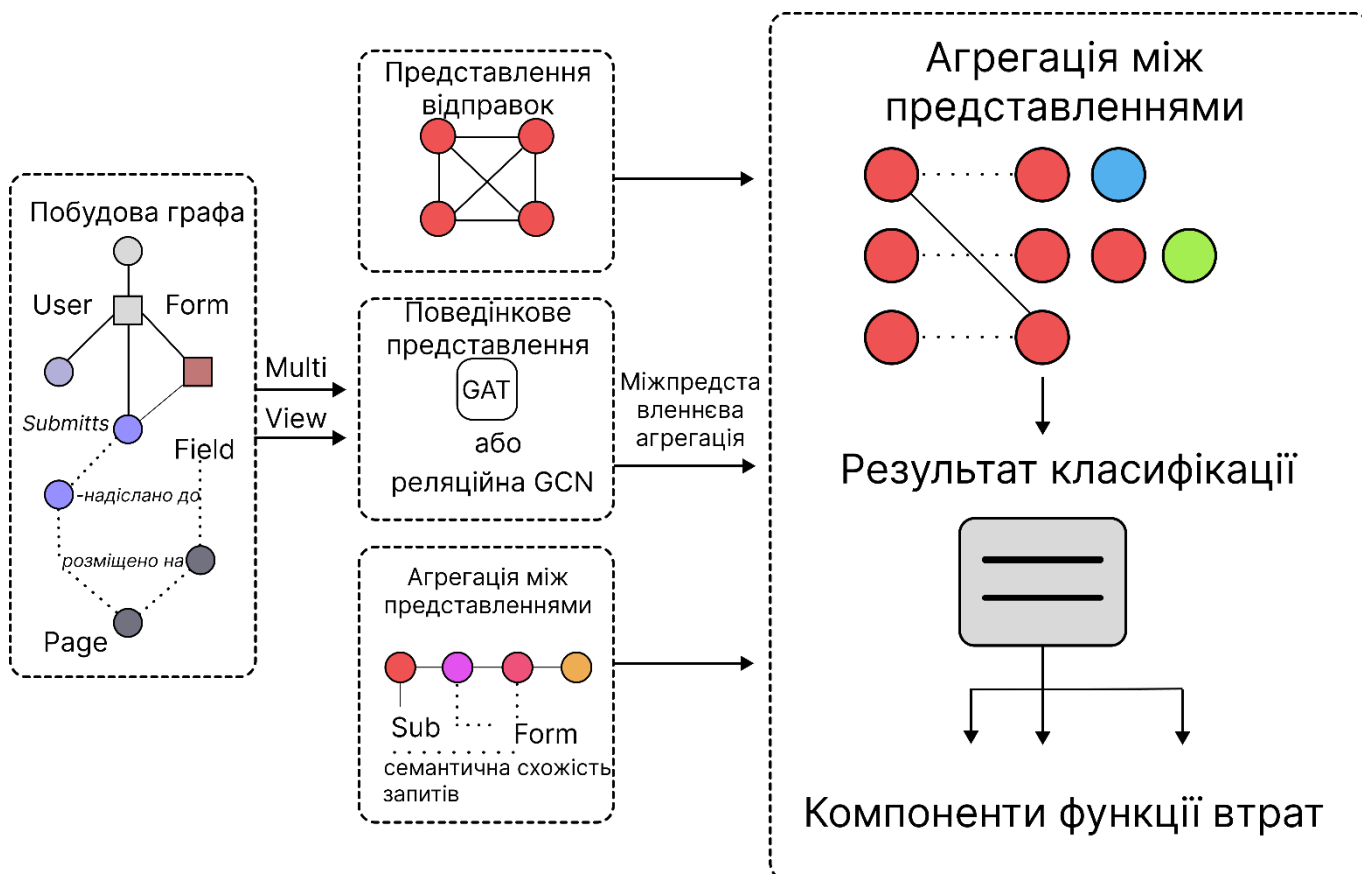


Рис. 3.4 Формалізація графово-неймережевого методу фільтрації вебспаму (з контрастивним навчанням)

На першому етапі з подій взаємодії та метаданих формується граф, у якому вузли репрезентують сутності, пов'язані з поданнями форм (користувач, форма/поля, сторінки, відправка), а ребра відображають факти взаємодій, повторюваність джерел та інші типи зв'язків. Далі з цієї структури конструюються кілька взаємодоповнювальних подань: подання на рівні відправок, поведінково-технічне (гетерогенне) подання зв'язків та семантичне подання, яке узагальнює близькість текстових полів і URL. Кожне подання обробляється окремим графовим енкودером, що виконує локальну агрегацію контексту та формує векторні представлення для вузлів, які відповідають відправкам. Наступний етап передбачає міжпредставленнєве узгодження та адаптивне злиття отриманих векторів в інтегральне представлення, придатне для прийняття рішення; це підвищує робастність методу у випадках шуму, обфускації або часткової недоступності окремих каналів сигналів. Завершальний блок формує прогноз (ймовірність/клас належності до спаму), а навчальний процес

поєднує компоненту на розмічених прикладах та контрастивне узгодження між поданнями.

Об'єктом аналізу вважається окрема відправка вебформи (submission), для якої доступні контентні атрибути (вміст полів, URL), поведінкові та технічні характеристики (пристрій, IP-префікс, User-Agent, маршрут сторінками), часові параметри, а також зв'язки з іншими подіями. Для підвищення інформативності та стійкості до маскуванню дані організовуються як багатопредставленнєва структура, де різні «погляди» відображають різні механізми узгодженості сигналів.

Багатопредставленнєва графова структура задається як множина графів, кожен з яких відповідає певному типу сигналів і зв'язків (рівень відправок, поведінково-гетерогенний рівень, семантичний рівень). Загальний опис подань задається формулою:

$$\mathcal{G}_{spat} = \{G^{(r)}\}_{r \in s, u, t}, \quad G^{(r)} = (V^{(r)}, E^{(r)}, X^{(r)}, A^{(r)}), \quad (3.33)$$

де \mathcal{G}_{spat} - багатопредставлена графова структура для аналізу вебспаму, r - індекс подання; s - подання на рівні відправок через вебформи, u - поведінково-гетерогенному поданню, t - семантичному поданню, $V^{(r)}$ - множина вузлів у поданні r , $E^{(r)}$ - множина ребер, $X^{(r)}$ - матриця ознак вузлів, $A^{(r)}$ - матриця суміжності графа. Вузли $V^{(r)}$ задають сутності відповідного подання, а ребра $E^{(r)}$ формуються за ознаками часової близькості, спільності джерела або структурної подібності:

$$V^{(r)} = v_1^{(r)}, v_2^{(r)}, \dots, v_{n_r}^{(r)}, n_r = |V^{(r)}|, \quad (3.34)$$

де $v_i^{(r)}$ - i -й вузол у графовому поданні r , а n_r - кількість вузлів у цьому поданні.

$$X^{(r)} = \begin{bmatrix} (x_1^{(r)})^\top \\ (x_2^{(r)})^\top \\ \vdots \\ (x_{n_r}^{(r)})^\top \end{bmatrix} \in \mathbb{R}^{n_r \times d_r}, \quad (3.35)$$

де $x_i^{(r)} \in \mathbb{R}^{d_r}$ - вектор ознак i -го вузла у поданні r , а d_r - кількість ознак одного вузла в цьому поданні.

$$A^{(r)} = (a_{ij}^{(r)})_{n_r \times n_r}, a_{ij}^{(r)} = \begin{cases} 1, & (v_i^{(r)}, v_j^{(r)}) \in E^{(r)}, \\ 0, & (v_i^{(r)}, v_j^{(r)}) \notin E^{(r)}. \end{cases} \quad (3.36)$$

де $A^{(r)}$ - матриця суміжності графа в поданні r , а елемент $a_{ij}^{(r)}$ показує наявність або відсутність зв'язку між вузлами $v_i^{(r)}$ та $v_j^{(r)}$.

У поданні s вузли відповідають окремим відправкам вебформ, а ребра формуються за часовою близькістю, спільністю джерела або структурною подібністю. У поданні u вузли відповідають користувачам, тобто пристроям, IP-префіксам, сторінкам і поданням, а ребра відображають факти взаємодії або спільності технічних атрибутів. У поданні t вузли пов'язані з текстовими представленнями, а ребра задаються на основі семантичної близькості. У поведінковому поданні ключовою є типізація відносин (різні види зв'язків мають різний семантичний зміст і ризик-вагу), що формалізується множиною відношень та розкладом суміжності за типами:

$$\mathcal{R}^{(u)} = submits, shares_{device}, shares_{IP}, located_{on}, \dots \quad (3.37)$$

Для кожного типу відношення $\rho \in \mathcal{R}^{(u)}$ може бути задана окрема матриця суміжності $A^{(u, \rho)}$, елементи якої показують, чи існує між вузлами відповідний тип зв'язку:

$$A^{(u, \rho)} = (a_{ij}^{(u, \rho)}), a_{ij}^{(u, \rho)} = \begin{cases} 1, & (v_i^{(u)}, v_j^{(u)}) \in E_{\rho}^{(u)}, \\ 0, & (v_i^{(u)}, v_j^{(u)}) \notin E_{\rho}^{(u)}. \end{cases} \quad (3.38)$$

Перевірка здійснюється на підмножині вузлів, що відповідають відправкам, для яких наявні бінарні мітки «спам/не спам»:

$$V_{lab} \subseteq V^{(s)}, y_i \in \mathcal{C}, \mathcal{C} = LEGIT, SUSPECT, SPAM, i \in V_{lab}, \quad (3.39)$$

де V_{lab} - підмножина вузлів-відправок, для яких відомі класові мітки, де $V^{(s)}$ - множина вузлів подання рівня відправок, i - індекс окремого вузла-відправки; y_i - істинна мітка для вузла i , $y_i = 1$ відповідає класу «спам», а $y_i = 0$ - класу «не спам». Задача зводиться до побудови відображення $f_{\theta}: V^{(s)} \rightarrow [0, 1]$, де θ - набір параметрів графово-нейромережевого класифікатора, а значення $f_{\theta}(i)$ задає оцінку ймовірності того, що відправка, представлена вузлом i , належить до класу спаму. У

кожному поданні використовується спеціалізований графовий енкодер, що виконує локальну агрегацію контексту сусідів. Для гетерогенних типізованих зв'язків доцільним є енкодер класу R-GCN, тоді як для більш однорідних графів відправок і семантичних зв'язків застосовні GCN/GAT-подібні шари. Параметризація пошарового перетворення для кожного подання визначається матрицями ваг:

$$W^{(r,l)} \in \mathbb{R}^{d_{l+1} \times d_l}, \quad (3.40)$$

де $W^{(r,l)}$ - матриця ваг для подання v на l -му шарі енкодера, \mathbb{R} - множина дійсних чисел, $r \in \{s, u, t\}$ - індекс подання, де s відповідає поданню рівня відправок, u - поведінково-гетерогенному поданню, t - семантичному поданню, $l = 0, 1, \dots, L - 1$ - номер шару енкодера, d_l - розмірність вхідного вектора ознак на l -му шарі, d_{l+1} - розмірність вихідного вектора ознак на наступному шарі. Таким чином, матриця $W^{(v,l)}$ виконує лінійне перетворення ознак вузла з простору \mathbb{R}^{d_l} у простір $\mathbb{R}^{d_{l+1}}$ для відповідного подання v .

Подальший етап - адаптивне злиття отриманих представлень, яке вводить залежні від вузла ваги внеску окремих каналів:

$$z_i = \sum_{r \in s, u, t} \alpha_i^{(r)} z_i^{(r)}, \quad (3.41)$$

де z_i - представлення вузла i після злиття подань, $z_i^{(r)}$ - представлення вузла i , отримане з подання r , $\alpha_i^{(r)}$ - вага внеску подання r для вузла i , а $r \in s, u, t$ визначає одне з трьох подань: рівень відправок, поведінково-гетерогенне або семантичне подання. Такий механізм дозволяє моделі динамічно надавати перевагу поведінковим, структурним або контентним сигналам залежно від характеру конкретної відправки, а також коректно працювати за часткової відсутності окремих каналів. Ваги внеску визначаються через нормовану експоненційну функцію:

$$\alpha_i^{(r)} = \frac{\exp(e_i^{(r)})}{\sum_{r' \in s, u, t} \exp(e_i^{(r')})}, \quad (3.42)$$

де $e_i^{(r)}$ - оцінка важливості подання r для вузла i .

Ваги злиття можуть інтерпретуватися як механізм атрибуції рішення, оскільки вони показують, яке з подань зробило домінуючий внесок у сформований висновок.

Щоб зменшити залежність від обсягу розмітки та підвищити робастність до дрейфу, навчання поєднує супервізовану компоненту на мічених відправках з контрастивним узгодженням між поданнями. Контрастивна складова спрямована на те, щоб подання однієї і тієї ж сутності (відправки) в різних графах були узгодженими, а випадкові або семантично далекі пари - розділеними в просторі ознак. У термінах експлуатації це знижує ризик деградації якості при зміні тактик спаму, коли один канал може бути системно обфускований, але поведінкові та структурні сигнали зберігають інформативність.

Окремо враховується вимога відсутності витоків при оцінюванні, де розділення на train/val/test здійснюється за часовими зрізами, а всі перетворення мають налаштовуватися лише на навчальному інтервалі. Такий протокол є критичним для задач із поточковими даними та ковзними вікнами, де несанкціонований доступ до майбутніх відомостей може штучно завищити метрики.

Для переходу від «моделі як класифікатора» до «моделі як компонента безпеки» необхідно забезпечити незмінність (або принаймні доказовість незмінності) ключових артефактів прийняття рішення. У запропонованій схемі до таких артефактів належать: версія моделі та її гіперпараметрів, поріг прийняття рішення, агреговані ознаки або їхні хеш-подання, атрибуція каналів (ваги злиття), часові мітки події та результат (ймовірність спаму і класове рішення). Механізм забезпечення цілісності реалізується як незмінний журнал для контрольного запису рішення. У журналі фіксується не весь сирий контент (що може суперечити вимогам приватності), а мінімально достатній набір полів у вигляді хешів або агрегатів: хешовані ідентифікатори, узагальнені IP-префікси, контрольні суми текстових полів і URL, маркери часових вікон, а також цифровий підпис контрольного запису рішення. Така фіксація забезпечує можливість незалежної перевірки того, що рішення було прийняте саме з тією моделлю, при тих самих налаштуваннях і на відповідному наборі сигналів, без розкриття чутливих даних.

Підхід узгоджується з принципами, такими як рішення інтелектуального модуля трактується як подія безпеки, що підлягає аудитному збереженню. Незмінність журналу забезпечує відтворюваність розслідувань інцидентів, а також

знижує ризики підміни результатів на етапі передачі між компонентами вебзастосунку.

У типовому випадку розрізняються принаймні три режими реагування: пропуск без додаткових дій для низькоризикових відправок, відкладене рішення (карантин/модерація/підвищений контроль) для середнього ризику, негайне блокування або вимога додаткової верифікації (CAPTCHA/2FA/повторне підтвердження) для високого ризику. Вибір порогів є не статичним, а операційно керованим: він має калібруватися за метриками, релевантними для дисбалансних даних і несиметричних втрат. На практиці доцільним є налаштування порога під обмеження на хибні спрацювання FP-rate з вимогою забезпечити заданий рівень Recall, або оптимізація за PR-AUC та похідними показниками на валідаційному часовому зрізі. Порогові значення та їх зміни також підлягають журналюванню в контурі цілісності, щоб забезпечити коректну інтерпретацію історичних рішень. Важливою особливістю моделі є інтерпретованість на рівні каналів через ваги злиття. Це дозволяє формувати правила реагування не лише за «ймовірністю спаму», а й за джерелом сигналу. Наприклад, домінування поведінкового каналу u може свідчити про координовану атаку на рівні пристроїв/IP/маршрутів, тоді як домінування семантичного каналу t - про контентні дублікати або семантичні варіації шаблонів. Така диференціація корисна для вибору найменш інвазивних контрзаходів і зменшення негативного впливу на легітимних користувачів.

Запропонований метод графово-нейромережевого виявлення вебспаму та підозрілої активності у вебсистемах орієнтований на роботу в реальному або квазіреальному часі, що накладає обмеження на побудову графів і виконання інференсу. Для забезпечення масштабованості застосовується інкрементальне оновлення графів у часовому вікні спостереження, а також локальне семплювання сусідів при проходженні GNN-шарів. Семантичне подання, побудоване як kNN-граф, може потребувати періодичного переіндексування; частота переіндексації визначається компромісом між актуальністю зв'язків і обчислювальною вартістю. Контур аудиту складається з фіксації події (submission), результату класифікації, обраного правила реагування та контрольного запису рішення у незмінному журналі.

У випадку інцидентів або апеляцій (хибне блокування) журнал дозволяє реконструювати ланцюг рішень без доступу до персональних даних, а також виявити дрифти в тактиках атак через агреговану статистику ваг злиття та типів зв'язків, що домінують у певні періоди.

Така схема поєднує сучасні методи машинного навчання з практичними інструментами керування контентом. Для того, щоб створити ефективну систему захисту вебзастосунків від спаму, було вирішено враховувати широкий набір параметрів, отриманих безпосередньо з форм, які користувачі заповнюють на сайтах (Табл. 3.1). Серед ключових показників виділяється IP-адреса відправника, яка дає можливість ідентифікувати його місцеперебування та виявляти регіони з високим рівнем активності спам-ботів. Записи зі статусом “Не визначено” використовуються як нерозмічені приклади для подальшої перевірки і входять до множини V_{lab} під час навчання, тільки після визначення адміністратором.

Фіксується не лише точний час відправлення, а й проміжок, що минув з моменту останнього повідомлення з тієї самої IP-адреси. Якщо система виявляє надто часті запити від одного джерела, це може вказувати на спробу спам-атаки. Такий підхід дозволяє швидко виявляти підозрілу активність і блокувати подальше розповсюдження небажаного контенту. Ще одним важливим аспектом аналізу є власне текст повідомлення. Система перевіряє повідомлення на наявність фраз і ключових слів, типових для спаму або шахрайських розсилок. Слова чи вирази на кшталт «безкоштовний приз», «натисніть сюди» або «ви виграли» вказують на підвищений ризик шахрайства та автоматично привертають увагу системи. Крім того, інформація про ім'я та телефонний номер користувача також аналізується. Імена перевіряються на наявність нетипових або підозрілих форматів, характерних для автоматичного заповнення форм ботами. Телефонні номери перевіряються на відповідність стандартним шаблонам і реальним телефонним кодам. Якщо користувач вказує номер телефону у неправильному форматі або явно вигадані дані, це може свідчити про спробу автоматизованої атаки.

Таблиця 3.1

Дані з вебформ, які використовуються для аналізу та класифікації спам-повідомлень

IP-адреса	Країна	Час відправлення	Час останнього відправлення з цього IP	Повідомлення з форми	Ім'я з форми	Телефон з форми	Статус
89.209.255.5	Україна	12.03.2024 14:00	12.03.2024 13:55	Привіт, хочу замовити послугу.	Олександр	380501234567	Не СПАМ
2.57.8.55	Польща	12.03.2024 14:05	11.03.2024 18:00	Клікніть тут для отримання призу!	ІванІван	123456	СПАМ
2.16.90.11	Німеччина	12.03.2024 14:10	10.03.2024 9:00	Прошу на зворотній зв'язок	Анна	490123456789	Не визначено
...

Після того, як інформація зібрана, її необхідно підготувати для подальшого аналізу. Цей етап складається з двох основних процесів: нормалізації та векторизації. Нормалізація спрямована на приведення отриманих даних, таких як час відправлення або номери телефонів, до одного стандартного формату. Завдяки цьому стає можливим ефективно порівняння та аналіз інформації, отриманої з різних джерел. Наступним кроком є векторизація, яка перетворює текстові повідомлення в числові масиви даних. Саме ці числові вектори надалі використовуються моделлю машинного навчання. Після завершення попередньої обробки дані потрапляють до графової нейронної мережі. Ця мережа використовує модель, що вже була натренована на основі великого обсягу попередньої інформації. Вона аналізує взаємозв'язки між елементами та виявляє характерні ознаки, притаманні спаму. Кожне нове повідомлення, що аналізується системою, не тільки проходить класифікацію, а й сприяє подальшому вдосконаленню моделі.

Отже, адаптація методу графово-нейромережевого виявлення вебспаму та підозрілої активності до зміни шаблонів загроз полягає у використанні багатопредставленого графового опису подій, динамічного оновлення зв'язків між поданнями, правил реагування та механізмів аудитної фіксації рішень. Такий підхід дозволяє розглядати вебформу не як ізольоване повідомлення, а як елемент

пов'язаного потоку подій, у якому враховуються контентні, технічні, часові та поведінкові сигнали. Завдяки цьому метод зберігає здатність розрізняти легітимні, підозрілі та шкідливі звернення навіть за зміни тактик спаму, а узгодження з механізмами моделі ІКДЦ забезпечує доказовість рішень, контроль цілісності журналу та можливість подальшого аудиту без розкриття чутливих даних.

Висновки до розділу 3

У третьому розділі сформовано дані та ознаки для аналізу вебконтенту і поведінки користувачів у межах задачі виявлення вебспаму та підозрілої активності у вебсистемах. Визначено підхід до підготовки вхідних даних, їх очищення, кодування, нормалізації та масштабування, а також описано формування ризикових компонентів, що враховують фінансові, географічні, часові та поведінкові характеристики подій. Це створює основу для подальшого використання підготовлених даних у графово-нейромережевому методі аналізу вебспаму та підозрілої активності.

Формалізовано багатопредставлений графовий опис подій вебсистеми, у якому подання через вебформи розглядаються не ізольовано, а як елементи пов'язаного потоку подій. У такому описі враховуються локальні ознаки повідомлення, контентні характеристики, поведінковий контекст, часові параметри, мережеві атрибути та зв'язки між об'єктами взаємодії. Це дозволяє аналізувати не лише зміст окремого звернення, а й структурні та поведінкові закономірності, що виникають між групами подій.

Вперше розроблено метод графово-нейромережевого виявлення вебспаму та підозрілої активності у вебсистемах, що ґрунтується на моделі ІКДЦ та багатопредставленому графовому описі подій. У межах методу події подаються через систему технічних, змістовних, часово-поведінкових і контекстних ознак із урахуванням зв'язків між подіями та результатами аналітичного оцінювання. Це забезпечує можливість розрізнення легітимних, підозрілих і шкідливих звернень,

підвищує обґрунтованість виявлення вебспау та сприяє зменшенню частки хибних спрацювань.

Обґрунтовано адаптацію методу до зміни шаблонів загроз, правил реагування та механізмів забезпечення цілісності. Показано, що використання контентних, часових, мережевих і поведінкових зв'язків між подіями підвищує стійкість методу до модифікації спам-тактик, а узгодження результатів класифікації з механізмами ІКДЦ забезпечує аудитну відтворюваність рішень, контроль цілісності журналу та можливість подальшої перевірки без розкриття чутливих даних.

Таким чином, у третьому розділі розроблено математичний та алгоритмічний апарат графово-нейромережевого виявлення вебспау та підозрілої активності у вебсистемах. Запропонований підхід поєднує підготовку даних, формування системи ознак технічного, змістовного, часово-поведінкового й контекстного характеру, багатопредставлений графовий опис подій та аналіз зв'язків між подіями й результатами аналітичного оцінювання.

РОЗДІЛ 4 МЕТОД ІНТЕГРОВАНОГО ЗАБЕЗПЕЧЕННЯ ДОВІРИ Й ЦІЛІСНОСТІ У ВЕБСИСТЕМАХ, ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА

4.1. Метод інтегрованого забезпечення довіри й цілісності у вебсистемах

Метод інтегрованого забезпечення довіри й цілісності у вебсистемах призначений для узгодженої обробки критичних подій у межах єдиного контуру, який поєднує криптографічну фіксацію події, формування ознак, графово-нейромережеве або транзакційне оцінювання ризику, прийняття рішення, вибір дії реагування та включення доказового запису до незмінного журналу. На відміну від ізольованих підходів, у яких окремо працюють детектор загроз, механізм контролю доступу або журнал подій, запропонований метод забезпечує наскрізний причинно-наслідковий зв'язок між подією, її оцінкою, прийнятим рішенням, дією системи та доказовим записом, придатним для подальшої аудиторної перевірки.

На схемі (Рис. 4.1) відображено повний цикл функціонування методу, тобто надходження та нормалізацію подій вебформ і транзакцій, їх криптографічну фіксацію, формування ознак, побудову багатопредставлених графів, графове кодування, інтеграцію сигналів, оцінювання ризику, класифікацію подій, застосування політики реагування, формування пакета доказовості рішення, його включення до незмінного журналу, а також контур аудиту, розмітки інцидентів, навчання та версіонування моделей.

Основою методу є композиція функціональних відображень критичної події у клас рішення, дію реагування, доказовий запис і результат верифікації. Метод спирається на модель ІКДЦ, у якій критична подія, її контекст, ознакове подання, політики реагування та незмінний журнал формалізовано відповідно до (2.1) - (2.36), а також використовує метод блокчейн-верифікованого журналювання критичних подій і контролю доступу відповідно до (2.37) - (2.47) та метод графово-нейромережевого виявлення вебспау й підозрілої активності (3.17) - (3.26).

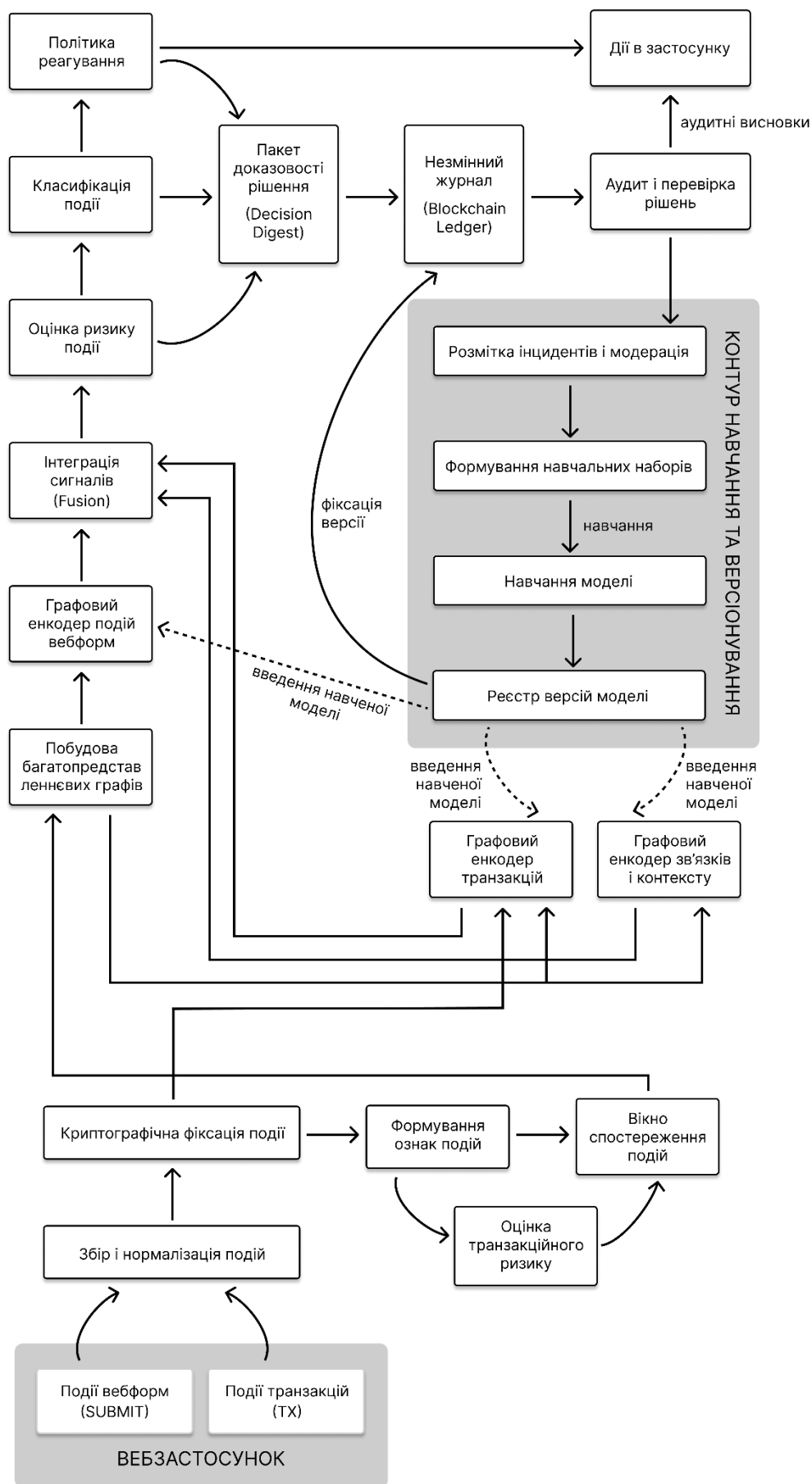


Рис. 4.1 Схема інтегрованого контуру довіри у вебзастосунку на основі блокчейн журналювання та графового навчання

Узагальнено метод інтегрованого забезпечення довіри й цілісності задається як відображення:

$$\mathcal{M}_{INT}: E \rightarrow \mathcal{Y} \times \mathcal{A}_{resp} \times \mathcal{Q} \times \mathcal{B}, \quad (4.1)$$

де \mathcal{M}_{INT} - метод інтегрованого забезпечення довіри й цілісності у вебсистемах, E - множина критичних подій, визначена в моделі ІКДЦ, $\mathcal{Y} = \text{SAFE, SUSPECT, THREAT}$ - множина класів рішення, $\mathcal{A}_{resp} = \text{ALLOW, VERIFY, QUARANTINE, BLOCK}$ - множина дій реагування, \mathcal{Q} - множина доказових записів, $\mathcal{B} = 0, 1$ - множина результатів верифікації, де 1 означає коректність запису, а 0 - порушення цілісності, автентичності або зв'язності журналу.

Для окремої критичної події $e_t \in E$, поданої відповідно до (2.2), результат роботи методу визначається як:

$$\mathcal{M}_{INT}(e_t) = (y_t, \rho_t, q_t, v_t), \quad (4.2)$$

де y_t - клас рішення щодо події, ρ_t - дія політики реагування, q_t - доказовий запис, що фіксує підстави прийнятого рішення, $v_t \in 0, 1$ - результат верифікації доказового запису та його включення до незмінного журналу. Для позначення дії реагування використовується ρ_t , щоб не змішувати її з аудитним записом a_i , визначеним у (2.31). Функціонально метод задається композицією відображень:

$$\mathcal{M}_{INT} = \mathcal{J}_\Lambda \circ \chi \circ \pi \circ \delta \circ \Omega \circ \eta, \quad (4.3)$$

де η - відображення криптографічної фіксації події, Ω - відображення формування інтегрованої оцінки ризику, δ - відображення оцінки ризику у клас рішення, π - відображення класу рішення та контексту у дію політики реагування, χ - відображення формування доказового запису, \mathcal{J}_Λ - відображення включення доказового запису до незмінного журналу Λ .

Першим етапом є криптографічна фіксація події:

$$\eta(e_t) = (c_t, h_t, sig_t), \quad (4.4)$$

де c_t - канонічне подання події, h_t - криптографічний хеш події, sig_t - цифровий підпис хешу події. Ці компоненти визначаються відповідно до формул (2.5) – (2.7), а перевірка підпису виконується відповідно до (2.8). Це забезпечує однозначне подання події та її криптографічно перевірювану прив'язку до джерела реєстрації.

Другим етапом є формування інтегрованої оцінки ризику:

$$\Omega(e_t) = p_t, p_t \in [0,1], \quad (4.5)$$

де p_t - нормалізована оцінка ризику критичної події. Значення p_t визначається залежно від типу події:

$$p_t = \begin{cases} p_v^{SPAM}, & type_t = SUBMIT, \\ R_t^{TX}, & type_t = TX, \\ p_t^{BASE}, & \text{для іншого типу події, якщо використовується загальна оцінка ризику,} \end{cases} \quad (4.6)$$

де $type_t$ - тип події, визначений у (2.2), p_v^{SPAM} - імовірність належності події вебформи до класу *SPAM*, R_t^{TX} - інтегральний ризик транзакції, обчислений відповідно до формул (3.1) – (3.16), p_t^{BASE} - загальна оцінка ризику події, визначена відповідно до (2.14). Метод використовує вже формалізовані оцінки методів блокчейн-верифікованого журналювання критичних подій і контролю доступу та методу графово-нейромережевого виявлення вебспау й підозрілої активності і приводить їх до єдиної шкали $[0,1]$.

Третім етапом є перетворення оцінки ризику у клас рішення:

$$y_t = \delta(p_t; \tau_1, \tau_2), \quad (4.7)$$

де $\delta(\cdot)$ - порогове правило класифікації, що переводить оцінку p_t у клас *SAFE*, *SUSPECT* або *THREAT*. Порогові значення τ_1 , τ_2 і правило класифікації використовуються відповідно до (2.26) – (2.30). Клас *SAFE* означає штатну обробку події, *SUSPECT* - необхідність контрольованої перевірки, а *THREAT* - запуск сценарію блокування.

Четвертим етапом є вибір дії політики реагування:

$$\rho_t = \pi(y_t, ctx_t, P), \quad (4.8)$$

де $\rho_t \in \mathcal{A}_{resp}$ - дія реагування, ctx_t - контекст події, визначений відповідно до (2.3), P - множина політик реагування, задана в моделі ІКДЦ, $\pi(\cdot)$ - функція політики реагування. У базовому випадку для класу *SAFE* може застосовуватися дія *ALLOW*, для класу *SUSPECT* - *VERIFY* або *QUARANTINE*, а для класу *THREAT* - *BLOCK*. Конкретна відповідність між класом рішення та дією системи задається політикою реагування P , що дозволяє адаптувати метод до вимог конкретної вебсистеми.

П'ятим етапом є формування доказового запису:

$$q_t = (h_t, sig_t, model_{id_t}, model_{hash_t}, policy_{hash_t}, window_{id_t}, p_t, y_t, \rho_t, \alpha_t, ts_t), \quad (4.9)$$

де q_t - доказовий запис події; h_t - хеш канонічного подання події, sig_t - цифровий підпис, $model_{id_t}$ - ідентифікатор версії моделі, яка використовувалася для прийняття рішення, $model_{hash_t}$ - контрольна сума параметрів цієї моделі, $policy_{hash_t}$ - контрольна сума політики реагування, $window_{id_t}$ - ідентифікатор часового вікна, у межах якого оцінювалася подія, p_t - оцінка ризику, y_t - клас рішення, ρ_t - дія реагування, α_t - вектор ваг внеску графових подань, ts_t - час формування рішення. Контрольні суми $model_{hash_t}$ та $policy_{hash_t}$ обчислюються криптографічною хеш-функцією $H(\cdot)$, визначеною у (2.6).

Шостим етапом є включення доказового запису до незмінного журналу:

$$\mathcal{J}_\Lambda(q_t) = (B_k, v_t), \quad (4.10)$$

де B_k - блок незмінного журналу, до якого включається доказовий запис q_t , $v_t \in 0,1$ - результат верифікації включення запису до журналу. Формування блоку, множини аудитних записів, кореневого хешу ієрархічного дерева хешів та хешу блоку виконується відповідно до (2.32) – (2.36), а умови верифікації запису та блоку - відповідно до (2.47).

Послідовність роботи методу можна подати як конвеєр:

$$e_t \rightarrow \eta(e_t) \rightarrow \Omega(e_t) \rightarrow \delta(p_t) \rightarrow \pi(y_t, ctx_t, P) \rightarrow \chi(\cdot) \rightarrow \mathcal{J}_\Lambda(q_t), \quad (4.11)$$

де вхідна критична подія e_t послідовно проходить криптографічну фіксацію, формування ризикової оцінки, класифікацію, вибір дії реагування, формування доказового запису та включення до незмінного журналу.

У прикладному аспекті метод працює таким чином, якщо подія є поданням через вебформу, то її ризик оцінюється графово-нейромережевим методом з урахуванням контентних, часових, мережевих і поведінкових зв'язків між подіями. Якщо подія є транзакційною, то для неї використовується інтегральна оцінка транзакційного ризику, яка враховує суму платежу, географічний фактор, час виконання операції та тип платіжного інструмента. Після цього отримана оцінка ризику переходить у спільне порогове правило класифікації, а результат класифікації використовується політикою реагування для вибору дії системи.

Доказовий запис q_t є ключовим елементом інтегрованого методу, оскільки поєднує технічний факт події, криптографічну фіксацію, версію моделі, політику реагування, оцінку ризику, клас рішення, дію системи та атрибуцію графових подань. Завдяки цьому під час аудиту можна перевірити не лише сам факт події, а й те, якою моделлю, за якими параметрами, у межах якого часового вікна та за якою політикою було прийнято відповідне рішення. Метод також передбачає підтримку еволюції моделей і політик реагування, нові версії моделей, зміни порогів, оновлення правил реагування та зміни параметрів графового аналізу повинні супроводжуватися оновленням відповідних контрольних сум у доказових записах. Це забезпечує простежуваність не лише окремих подій, а й еволюції самої системи прийняття рішень.

Таким чином, вперше розроблено метод інтегрованого забезпечення довіри й цілісності у вебсистемах, який є композицією функціональних відображень критичної події у клас рішення, дію реагування, доказовий запис і результат аудитної верифікації. Метод ґрунтується на моделі ІКДЦ, методі блокчейн-верифікованого журналювання критичних подій і контролю доступу та методі графово-нейромережевого виявлення вебспаму й підозрілої активності. Його відмінність полягає в тому, що критична подія проходить єдиний цикл обробки, канонізацію та криптографічну фіксацію, формування ознак, графово-нейромережеве оцінювання ризику, класифікацію за рівнем небезпеки, вибір політики реагування та включення доказового запису до незмінного журналу.

4.2. Адаптація моделі ІКДЦ та інтегрованого методу для побудови прототипу захисту платіжного контуру у вебсередовищі

Електронні платежі є базовим механізмом фінансових операцій у сервісних і торговельних вебсистемах та зазвичай інтегруються з CRM- і бухгалтерськими платформами для наскрізного обліку транзакцій, автоматизації звітності й аудиту подій. Водночас зростання кількості інтеграційних зв'язків розширює поверхню атаки, а особливо небезпечними стають загрози, спрямовані на порушення цілісності

й автентичності платіжних даних. Одним із найкритичніших сценаріїв є атака типу «людина посередині» (MITM), за якої зловмисник перехоплює або модифікує повідомлення між компонентами в режимі реального часу, що може призводити до підміни суми чи реквізитів, фальсифікації статусів оплати та спотворення облікових даних у CRM і бухгалтерії (Рис. 4.2). Типову взаємодію в такому середовищі доцільно подати у вигляді схеми потенційної MITM-атаки в платіжній системі.



Рис. 4.2 Схема MITM-атаки в платіжній системі

Прототипний контур довіри, побудований на основі моделі ІКДЦ та інтегрованого методу забезпечення довіри й цілісності, розглядається як контрольний шар, що вбудовується у наявний процес обробки платежів без повної перебудови платіжної інфраструктури. Основний принцип побудови полягає у багаторівневій перевірці критичних атрибутів транзакції на етапах: формування запиту, верифікація, контроль цілісності та «свіжості», детекція аномалій, підтвердження й передача результатів у CRM/бухгалтерію.

Функціональна схема розгортання прототипу включає такі логічні блоки: формування платіжного запиту, первинна верифікація транзакції, модуль протидії MITM, підтвердження платежу з подальшою синхронізацією з CRM/бухгалтерією, підсистема безпеки й журналювання, яка накопичує дані для аудиту та подальшого аналізу загроз. Узгоджену взаємодію цих елементів доцільно подати у вигляді структурно-функціональної схеми прототипного контуру довіри (Рис. 4.3).

У архітектурі враховано багатоканальність оплати: готівкові операції через пристрій сплати, безготівкові платежі через POS-термінал, а також онлайн-оплату за

QR-кодом через платіжний сервіс. Додатковим компонентом є сервіс реєстрації чеків, що генерує та реєструє електронний чек, надаючи користувачеві посилання на підтверджувальний документ. CRM-система отримує дані про транзакцію й чек, забезпечуючи зв'язок між оплатою та обліковими операціями, включно з пошуком платежу за номером чека, процедурами повернення та коригування.

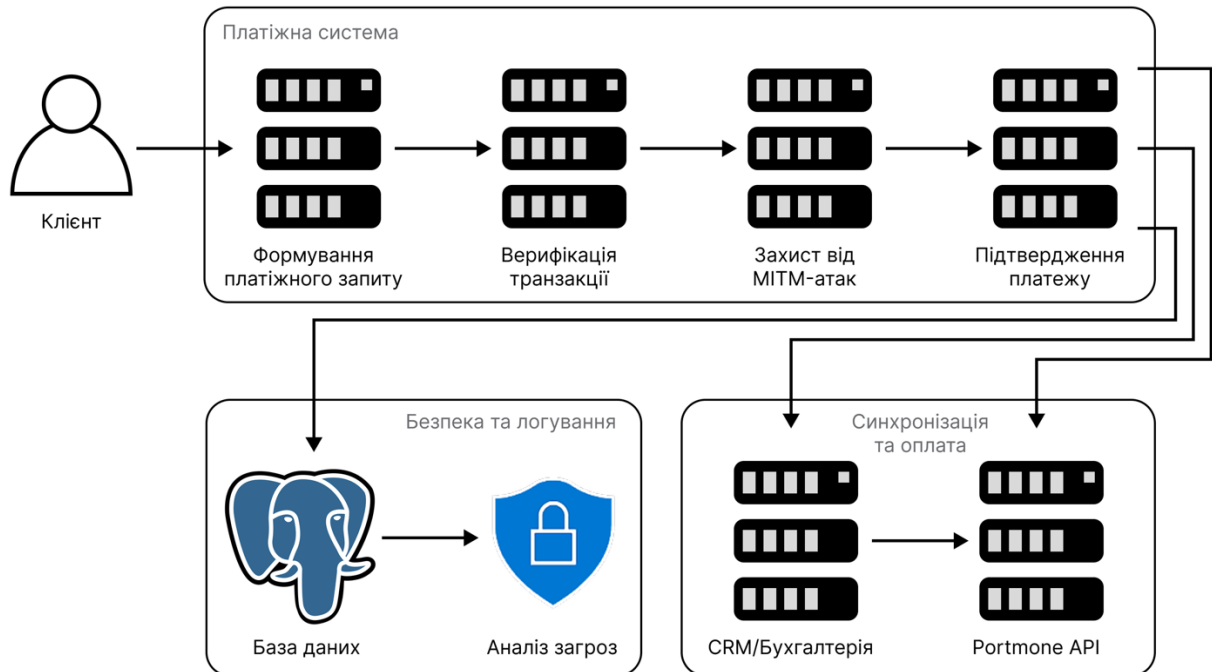


Рис. 4.3 Структурно-функціональна схема прототипного контуру довіри для захисту платіжного процесу та інтеграції з CRM/бухгалтерією

Важливо, що підсистема довіри у прототипі не підміняє платіжні сервіси та бізнес-логіку CRM, а інтегрується поверх наявного процесу як контрольний шар. Її призначення полягає у тому, щоб забезпечувати узгодженість ключових атрибутів транзакції на всіх етапах життєвого циклу платежу, починаючи від формування запиту й завершуючи передачею підтверджених результатів у CRM та бухгалтерську підсистему. У межах цього шару реалізовано механізми протидії підміні даних і повторному відтворенню повідомлень, що дає змогу мінімізувати ризики несанкціонованого повтору операцій або маніпуляцій параметрами транзакції. Результати перевірок і виявлені підозрілі ситуації перетворюються на події безпеки, які фіксуються у журналах і надалі використовуються для аудиту, аналізу інцидентів та вдосконалення механізмів захисту.

Для коректної інтеграції механізмів довіри необхідно уніфікувати представлення транзакції як об'єкта контролю, тобто задати її формалізований опис, який однаково інтерпретується всіма компонентами платіжного контуру та інтеграційними модулями. У межах адаптації моделі ІКДЦ до платіжного вебсередовища кожна транзакція подається як структурований набір параметрів, що охоплює не лише платіжні реквізити, а й супровідні атрибути електронного чека, верифікаційні поля та результати взаємодії із зовнішніми сервісами. Такий підхід забезпечує контроль цілісності не на рівні окремого повідомлення, а в межах усього ланцюга обробки «ініціація → підтвердження → облік», де розбіжності між етапами можуть бути ознакою підміни даних або несанкціонованого втручання. Повний опис даних для i -ї транзакції задається кортежем:

$$X_i = (T_i, P_i, C_i, V_i, R_i), \quad (4.12)$$

де $i \in \mathbb{N}$ - індекс транзакції, X_i - узагальнений профіль i -ї транзакції, який використовується як вхідний об'єкт для процедур контролю довіри та подальшого аналізу, T_i - блок ідентифікаційних атрибутів транзакції, що містить унікальний ідентифікатор транзакції та канал її виконання, P_i - блок платіжних реквізитів, що включає суму платежу, час ініціації транзакції та спосіб оплати, C_i - блок атрибутів електронного чека, який містить номер чека та ідентифікатор ПРРО, V_i - блок верифікаційних даних, що включає номер телефону або інший ідентифікатор платника та одноразовий код підтвердження, R_i - блок відповіді зовнішнього сервісу, який містить статус виконання операції та службове повідомлення або код відповіді. Таким чином, кортеж X_i охоплює всі ключові атрибути транзакції, необхідні для контролю цілісності, підтвердження платежу та подальшого виявлення аномалій.

Спираючись на сформований опис даних X_i , у межах прикладної реалізації ІКДЦ використовується функція аномальності $D(X_i)$, яка кількісно відображає, наскільки характеристики поточної транзакції відхиляються від нормальної поведінки, зафіксованої у системі на основі історичних даних та типових патернів виконання платежів:

$$D(X_i) = \sum_{j=1}^m w_j \Delta_j(x_{ij}, x_j^{ref}), \quad (4.13)$$

де X_i - профіль i -ї транзакції, x_{ij} - значення j -го контрольованого атрибута у профілі X_i , x_j^{ref} - референтне (очікуване або типове) значення j -го атрибута, визначене на основі історичних даних, m - кількість атрибутів, що враховуються під час оцінювання аномальності, $w_j \geq 0$ - ваговий коефіцієнт j -го атрибута, $\Delta_j(x_{ij}, x_j^{ref})$ - функція локального відхилення:

$$\Delta_j(x_{ij}, x_j^{ref}) = \begin{cases} \min\left(1, \frac{|x_{ij} - x_j^{ref}|}{s_j + \varepsilon}\right), & \text{для числового атрибута,} \\ 0, & \text{для категоріального атрибута, якщо } x_{ij} = x_j^{ref}, \\ 1, & \text{для категоріального атрибута, якщо } x_{ij} \neq x_j^{ref}. \end{cases} \quad (4.14)$$

де $s_j > 0$ - допустимий масштаб відхилення для j -го числового атрибута, який може визначатися як стандартне відхилення або допустимий інтервал зміни цього атрибута за історичними даними; $\varepsilon > 0$ - мале число для уникнення ділення на нуль.

Чим більшим є значення $D(X_i)$, тим більш нетиповими вважаються поєднання реквізитів платежу, чекових атрибутів, верифікаційних даних і відповіді зовнішніх сервісів, а отже - тим вища ймовірність наявності аномальних або потенційно небезпечних характеристик, зокрема таких, що можуть бути зумовлені МІТМ-втручанням. На основі значення $D(X_i)$ встановлюється динамічний поріг δ ($\delta \geq 0$), який визначає межу допустимого відхилення для поточного режиму роботи системи та властивостей платіжного потоку. Рішення щодо статусу транзакції подається індикаторною функцією $u(T_i)$, що відображає результат перевірки для транзакції з ідентифікатором T_i .

$$u(T_i) = \begin{cases} 1, & |D(X_i) - D^{ref}| < \tau_D, \\ 0, & |D(X_i) - D^{ref}| \geq \tau_D, \end{cases} \quad (4.15)$$

де D^{ref} - базовий рівень аномальності для валідних транзакцій, а $\tau_D \geq 0$ - допустиме відхилення від цього рівня:

$$\tau_D = \mu_D + s_D, \quad (4.16)$$

де μ_D - середнє значення $D(X_i)$ для валідних транзакцій у валідаційній вибірці, а s_D - стандартне відхилення цих значень.

Отже, значення $u(T_i)=1$ означає, що транзакція не перевищує встановлену межу аномальності й розглядається як умовно безпечна (допускаючи, за потреби, додаткові

перевірки на інших етапах). Натомість $u(T_i)=0$ вказує, що рівень аномальності є надто високим, транзакції присвоюється статус підозрілої, після чого активуються механізми посиленої реакції блокування операції або додаткова верифікація платника залежно від політики безпеки та сценарію використання.

Оскільки МІТМ-атака передбачає втручання у канал обміну та, як наслідок, змінює принаймні один із суттєвих параметрів транзакції (наприклад, суму, часові атрибути, реквізити отримувача або чекові дані), така транзакція, як правило, демонструє зростання міри відхилення $D(X_i)$. За умови коректного налаштування процедури оцінювання аномальності та динамічного порога δ , ймовірність того, що модифікований платіж вийде за межі допустимого відхилення і буде позначений як підозрілий, є високою.

Для кількісного опису ефективності виявлення використовується показник рівня детектування P_d , який належить інтервалу $(0,1)$ і відображає ймовірність того, що спроба атаки буде виявлена системою:

$$P_d = 1 - (P_f)^N, \quad (4.17)$$

де $P_f \in [0,1)$ - ймовірність хибного негативу, тобто ймовірність того, що система не розпізнає реальну атаку та помилково класифікує її як нормальну операцію. На практиці P_f визначається за результатами тестування моделі або прототипу як відношення кількості хибнонегативних рішень до загальної кількості реальних атак:

$$P_f = \frac{FN}{TP+FN}, \quad (4.18)$$

де FN - кількість атак, які система не виявила, а TP - кількість атак, які система коректно виявила. Параметр $N \in \mathbb{N}$, $N \geq 1$, задає кількість незалежних або квазінезалежних перевірок, які виконуються в межах транзакційного ланцюга, на практиці це відповідає послідовному контролю різних складових транзакції, зокрема верифікаційного коду, відповіді платіжного сервісу, часових міток і маркерів «свіжості» повідомлень. З формули випливає, що зі зростанням N та зі зменшенням P_f (тобто з підвищенням якості моделі аномальності й точності перевірок) величина P_d монотонно наближається до 1, що відповідає практично гарантованому виявленню спроб фальсифікації даних у каналі платежу.

Разом із пороговою логікою та правилами цілісності, для підвищення точності розпізнавання складних і «маскованих» сценаріїв шахрайства у межах прикладної реалізації ІКДЦ використовується нейромережева модель бінарної класифікації транзакцій, яка перетворює вектор ознак x_i на оцінку ризику $r(x_i) \in [0,1]$, яка оперує векторизованим представленням транзакції x_i (отриманим із множини ознак X_i після кодування категоріальних параметрів і масштабування числових значень). Результатом обчислення є ризик-функція, значення якої лежить в інтервалі $(0,1)$:

$$r(x_i) = f_\theta(x_i), f_\theta(x_i) = g(Wx_i + b), g(z) = \frac{1}{1+e^{-z}}, f_\theta: R^d \rightarrow [0,1], \quad (4.19)$$

де $r(x_i)$ інтерпретується як імовірність того, що транзакція є атакованою або містить ознаки втручання, $f_\theta(\cdot)$ - нейромережева функція бінарної класифікації, яка перетворює вектор ознак транзакції x_i на скалярну оцінку ризику, θ - сукупність параметрів нейронної мережі (ваг і, за потреби, зсувів), що формують зв'язки між нейронами суміжних шарів у вигляді векторів і матриць, R^d - простір векторів ознак розмірності d . Під час навчання моделі параметри θ оптимізуються так, щоб мінімізувати вибрану функцію втрат і забезпечити максимальну узгодженість між прогнозом мережі та фактичними мітками класів, тобто підвищити точність розрізнення класів «нормальна транзакція» та «атака».

Для прийняття рішення про наявність загрози встановлюється порогове значення τ :

$$d_i^{MITM} = \begin{cases} NORMAL, & r(x_i) < \tau_{MITM}, \\ ATTACK, & r(x_i) \geq \tau_{MITM}. \end{cases} \quad (4.20)$$

де d_i^{MITM} - рішення моделі щодо i -ї транзакції, $r(x_i) \in [0,1]$ - оцінка ризику MITM-втручання, а $\tau_{MITM} \in [0,1]$ - поріг прийняття рішення. Порогове значення τ_{MITM} визначається на валідаційній вибірці як значення, що забезпечує прийнятний компроміс між пропущеними атаками та хибними спрацюваннями.

Логіка прототипу реалізована як послідовний конвеєр перевірок, що виконується до підтвердження оплати та передачі результатів у CRM і бухгалтерський контур. Після надходження платіжного запиту система перевіряє автентичність критичних полів, зокрема суми, способу оплати та часу ініціації, використовуючи криптографічний токен для зв'язування запиту з легітимним

джерелом. Далі виконується валідація структури повідомлення, допустимості значень і узгодженості реквізитів. Якщо запит не проходить перевірку, активується режим реагування з повторною верифікацією або блокуванням, формуванням події безпеки та журналюванням інциденту. Якщо запит є коректним, запускається модуль детекції аномалій на основі Autoencoder, який виявляє приховані відхилення в ознаках транзакції, характерні для MITM-втручання. За відсутності аномальності додатково перевіряється повторне відтворення повідомлення за допомогою timestamp-based nonce, що дозволяє оцінити його свіжість і унікальність у межах допустимого часового вікна. Якщо ознак повтору не виявлено, транзакція вважається такою, що пройшла захисний контур, і передається далі для підтвердження та синхронізації результатів. Узагальнена послідовність цих кроків подана на блок-схемі процесу перевірки та захисту транзакцій на Рис. 4.4.

Якщо на етапі шаблонної перевірки, аналізу аномалій або контролю replay транзакція визначається як підозріла, система переходить до режиму активного реагування. У такому режимі виконується повторна верифікація платника, блокування або скасування операції, формується повідомлення для адміністратора й фіксується подія безпеки в журналі. За потреби система генерує нове платіжне середовище з новим QR-кодом або новим платіжним посиланням, що унеможливорює повторне використання скомпрометованих даних. Журнали використовуються не лише для аудиту, а й для накопичення інцидентів, уточнення параметрів функції аномальності та перенавчання нейромережевої моделі. Для аналізу послідовностей подій у часі використовується LSTM, яка дає змогу виявляти серійні патерни атак у потоках Portmone, Checkbox та CRM. Підсистема довіри реалізується як окремий проміжний сервіс між платіжними каналами та CRM або бухгалтерією, що дозволяє інтегрувати її без зміни основної бізнес-логіки. Усі зовнішні взаємодії проходять через контрольовані інтерфейси з перевіркою цілісності, часу та контексту транзакції. Будь-яка розбіжність між підтвердженням оплати та реєстрацією електронного чека за сумою, ідентифікаторами, часовими мітками або статусами відповіді розглядається як ознака MITM-втручання і переводить процес у режим посиленої перевірки.

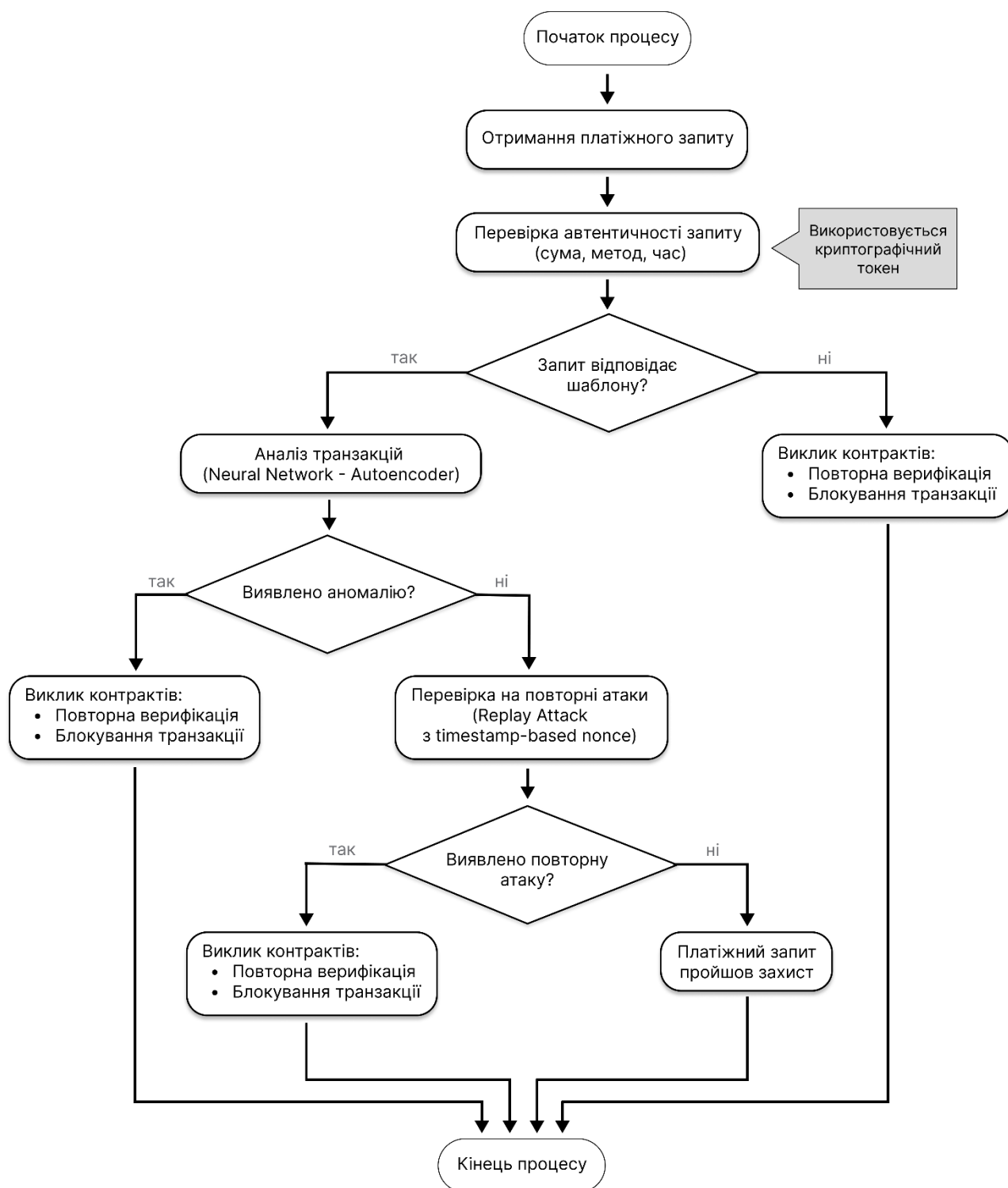


Рис. 4.4 Блок-схема алгоритму перевірки та захисту платіжних транзакцій від MITM і повторних атак

Для стабільної роботи детектор використовує розширений набір вхідних параметрів, що описують транзакцію як ланцюг взаємопов'язаних подій у платіжному контурі. До нього входять ідентифікатор транзакції, канал і спосіб

оплати, сума платежу, часові атрибути, параметри верифікації та відповідь зовнішніх сервісів. На виході модуль формує числову оцінку підозрілості в інтервалі від 0 до 1 і клас події, який відображає тип загрози. Це спрощує автоматичне реагування, журналювання та подальший аналіз інцидентів. Експериментальна перевірка виконувалась у контрольованому середовищі з розміченими прикладами атак. Оскільки в початковій інфраструктурі окремий механізм виявлення MITM був відсутній, пряме порівняння з попередньою конфігурацією не проводилося. Для тестування використано відкритий набір Kitsune Network Attack Dataset [131], який містить мережеві потоки, ознаки атак, значення аномальності та еталонну розмітку MITM-активності. Дані з типом атаки, характеристиками потоку, еталонним класом, оцінкою аномальності та рішенням моделі наведено в табл. 4.1.

Таблиця 4.1

Класифікації мережевих потоків/подій за оцінкою аномальності та рішенням моделі

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

Оскільки в початковій конфігурації платіжної системи до впровадження запропонованого прототипу не було спеціалізованого механізму детекції MITM-атак, коректне порівняння з «попередньою версією» є неможливим. Для кількісної оцінки результатів класифікації використовуються стандартні показники якості детектора, зокрема *TP* (True Positive) кількість атак, які модель коректно ідентифікувала як «атака», та *FN* (False Negative) кількість атак, які система пропустила, помилково

віднісши їх до «нормальних» подій. На цій основі обчислюється коефіцієнт повноти *Recall*, що відображає частку виявлених атак серед усіх реальних атак у вибірці:

$$\text{Recall} = \frac{TP}{TP + FN}, \quad (4.21)$$

де: *TP* - кількість правильно виявлених MITM-атак, *FN* - кількість MITM-атак, які система пропустила.

Таким чином, адаптація моделі ІКДЦ та інтегрованого методу забезпечення довіри й цілісності до платіжного вебсередовища дозволяє побудувати прототипний контур контролю транзакцій, у якому поєднуються формалізований опис платіжної події, перевірка критичних атрибутів, оцінювання аномальності, нейромережеве розпізнавання MITM-втручання, replay-контроль і незмінне журналювання результатів. Такий підхід забезпечує не лише виявлення підміни суми, статусу, чекових даних або відповіді зовнішнього сервісу, а й формування доказової бази для подальшого аудиту, розслідування інцидентів і коригування політик реагування в межах моделі ІКДЦ.

4.3. Архітектура та програмна реалізація інтегрованого контуру довіри й цілісності

Програмне представлення інтегрованого контуру довіри й цілісності ґрунтується на узгодженні моделі ІКДЦ, методу блокчейн-верифікованого журналювання критичних подій і контролю доступу та методу графово-нейромережевого виявлення вебспаму й підозрілої активності. Такий контур забезпечує єдиний маршрут обробки критичної події, від її надходження у вебсистему, нормалізації та формування ознак до оцінювання ризику, прийняття рішення, застосування політики реагування, незмінного журналювання та подальшої аудитної перевірки. Архітектурне подання через UML-схеми дозволяє показати, як окремі теоретичні компоненти роботи перетворюються на взаємопов'язані програмні модулі, здатні забезпечувати цілісність даних, простежуваність рішень і відтворюваність перевірки у вебсередовищі.

Першим архітектурним рівнем є контур контролю доступу та блокчейн-верифікованого журналювання, оскільки саме він забезпечує фіксацію рішень, перевірку їх цілісності та подальшу аудитну відтворюваність. Цей рівень відображає програмну інтерпретацію методу блокчейн-верифікованого журналювання критичних подій і контролю доступу. Для опису взаємодії компонентів системи контролю доступу та блокчейн-реєстру була розроблена UML-схема (Рис. 4.5), що відображає основні класи та їх ролі.

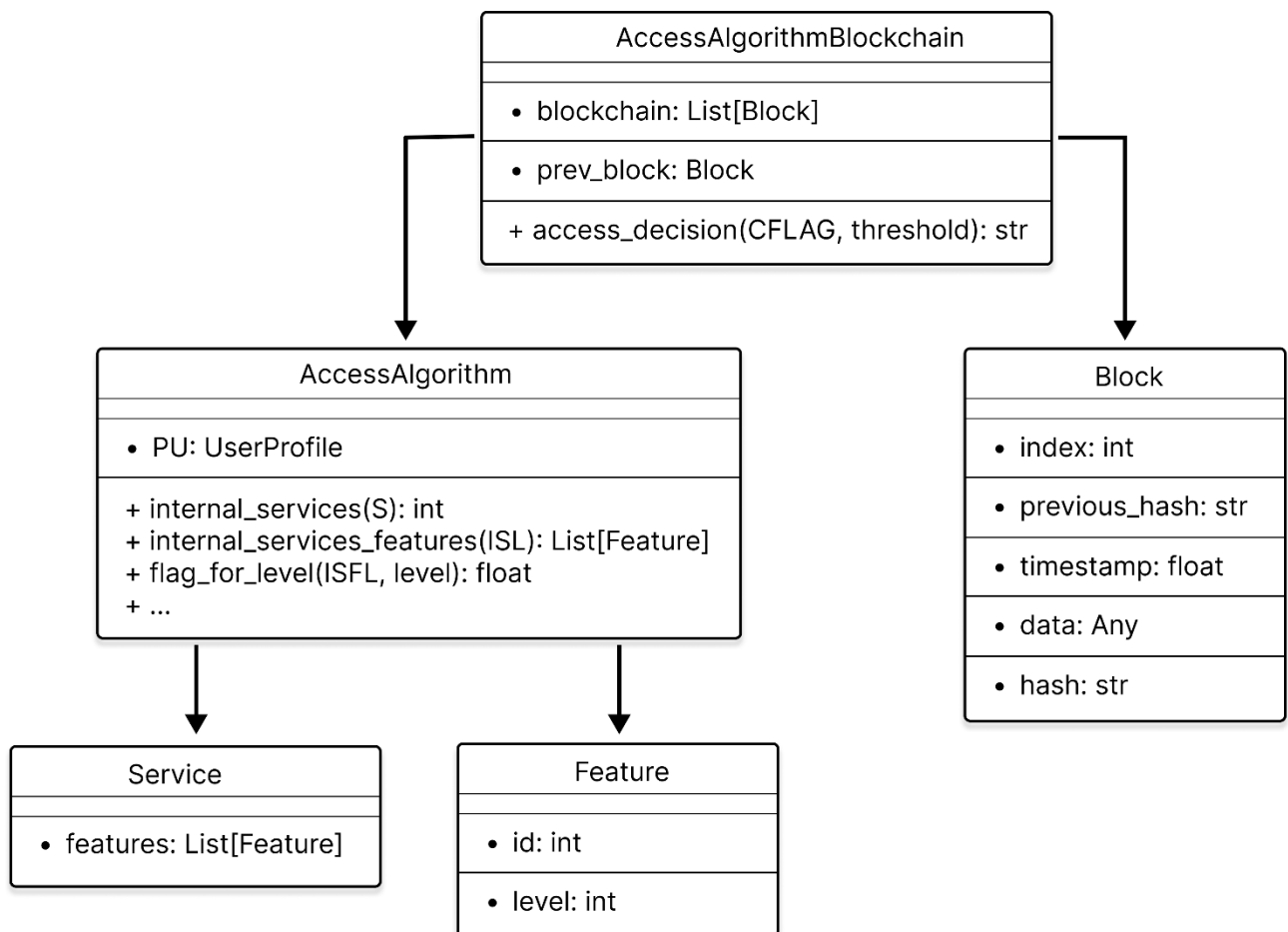


Рис. 4.5. UML-схема класів для опису методу блокчейн-верифікованого журналювання критичних подій і контролю доступу у вебсистемах

Ключовою ланкою у механізмі виступає клас **AccessAlgorithm**, який опрацьовує профіль користувача та специфікації сервісу, формує показники доступу та приймає рішення, що надалі фіксується як критична подія.

Основні компоненти системи:

- клас `Service` відображає окремий сервіс із набором функцій (`Feature`), доступ до яких підлягає контролю;
- клас `Feature` визначає окремі функції або ресурси з атрибутами, що впливають на рівень доступу;
- інтеграція блокчейну реалізується через клас `Block`, що містить індекс, попередній хеш, часову мітку, збережені дані (рішення про доступ) та власний хеш;
- розширений клас `AccessAlgorithmBlockchain` забезпечує збереження рішень у блокчейні та операції керування ланцюгом.

UML-діаграма (Рис. 4.6) деталізує програмну структуру незмінного журналювання критичних подій, показує, як рішення доступу, SQL-операції та інші критичні дії подаються як транзакції журналу, групуються у блоки та перевіряються через механізм зв'язності ланцюга. Центральним компонентом є клас `'Blockchain'`, який інкапсулює ланцюг блоків і забезпечує операції додавання та перевірки записів. Клас `'Block'` містить часову мітку, множину транзакцій і службові атрибути, необхідні для підтримки зв'язності ланцюга. Клас `'Transaction'` подає окрему критичну подію, зокрема SQL-операцію або запит доступу до ресурсу. Клас `'User'` представляє суб'єкта доступу, а клас `'WebResource'` - цільовий вебресурс. Клас `'DataAnalytics'` реалізує обробку накопичених транзакцій і формування аудитних показників. Така модульна структура демонструє, що метод блокчейн-верифікованого журналювання критичних подій і контролю доступу може бути реалізований як єдина підсистема, у якій функції контролю SQL-операцій, журналювання подій доступу, аналітичного аудиту та перевірки цілісності взаємопов'язані в межах моделі ІКДЦ.

Після формалізації контуру журналювання необхідно описати програмну частину, яка забезпечує підготовку даних та ознак для аналітичних компонентів системи. Саме цей рівень відповідає за перетворення подій вебформ і транзакційних записів у числові представлення, придатні для ризикового оцінювання та подальшого використання графово-нейромережевою моделлю. Для забезпечення відтворюваності, розширюваності та зручності інтеграції методу у вебсистеми

використано модульну архітектуру, яка представлена UML-схемою (Рис. 4.7). Така архітектура спрощує вбудовування алгоритмів машинного навчання у вебсистеми.

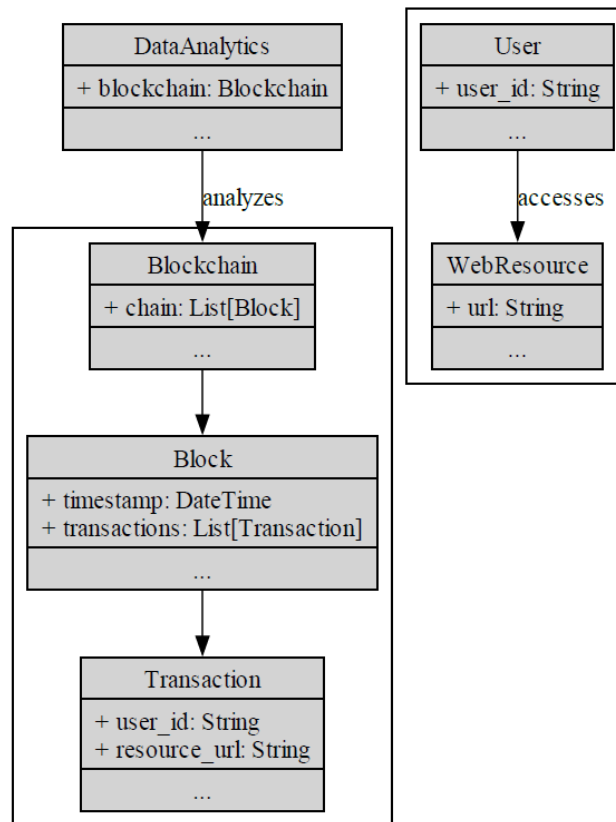


Рис. 4.6. UML-діаграма класів програмної реалізації методу блокчейн-верифікованого журналювання критичних подій і контролю доступу у вебсистемах

DataLoader відповідає за завантаження вхідних даних, приймає дані, отримані через POST-запити, та готує їх до подальшої обробки. Цей етап забезпечує відповідність отриманих даних необхідним форматам і їх придатність до подальшого аналізу. DataPreprocessor виконує попередню обробку даних для їх подальшого використання в методі графово-нейромережевого виявлення вебспаму та підозрілої активності у вебсистемах. До його функцій належать перетворення вхідних даних у DataFrame, який є зручним форматом для роботи з великими масивами інформації, а також застосування таких технік, як логарифмічне перетворення для коректної оцінки фінансових параметрів. Крім того, цей компонент відповідає за визначення географічних ризиків, що дає змогу враховувати місце здійснення транзакції як важливий фактор ризику.

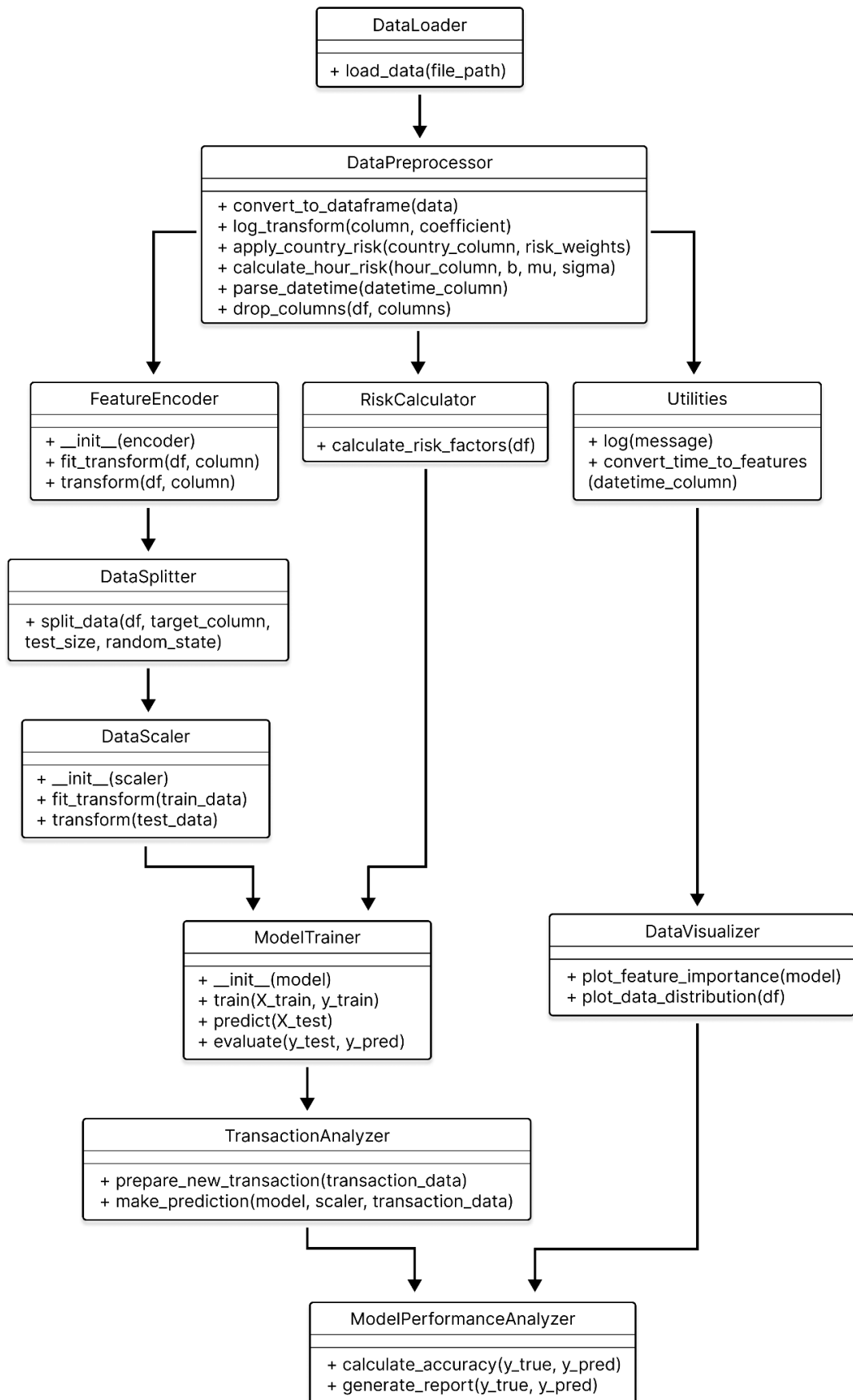


Рис. 4.7 UML-схема класів модуля обробки та аналізу даних

FeatureEncoder забезпечує кодування ознак і перетворює такі дані, як географічне положення та тип платіжної картки, у числові значення, придатні для використання в алгоритмах машинного навчання.

DataSplitter, DataScaler та ModelTrainer є ключовими компонентами на етапі побудови та використання моделі ІКДЦ. DataSplitter виконує поділ даних на навчальні та тестові набори, що є необхідним для формування стійкої та точної моделі ІКДЦ.

DataScaler відповідає за нормалізацію даних, що дозволяє привести ознаки до єдиного масштабу та підвищити ефективність навчання моделі ІКДЦ.

ModelTrainer здійснює навчання графово-нейромережевої або класифікаційної моделі на основі підготовлених даних і оцінює її якість на тестовій вибірці.

TransactionAnalyzer використовує навчену аналітичну модель у межах контуру ІКДЦ для визначення того, чи є транзакція підозрілою або безпечною.

RiskCalculator аналізує різні фактори ризику, пов'язані з транзакціями, використовуючи набір параметрів для оцінки ризику, таких як географічне положення, сума транзакції, час проведення операції та інші показники, щоб зробити точний висновок про рівень загрози. Utilities містить допоміжні інструменти, таких як журналювання подій, що дозволяє фіксувати всі дії системи, або конвертація часових міток, що допомагає точніше відстежувати часові аспекти транзакцій.

ModelPerformanceAnalyzer та DataVisualizer відповідають за візуалізацію даних, аналіз результативності моделі ІКДЦ та оцінювання ефективності методу графово-нейромережевого виявлення вебспау та підозрілої активності у вебсистемах, допомагаючи адміністраторам краще розуміти поведінку системи, аналізувати тенденції та виявляти можливості для оптимізації.

Окремим спеціалізованим компонентом аналітичного рівня є графово-нейромережевий класифікатор вебспау та підозрілої активності. Якщо попередня UML-схема (Рис. 4.7) описує загальний модуль підготовки даних і навчання, то наступна схема (Рис. 4.8) деталізує саме графову структуру класифікаційного компонента, у якому повідомлення, подані через вебформи, розглядаються як вузли графа, а зв'язки між ними - як підстава для контекстного аналізу. Реалізація системи

фільтрації вебспаму з використанням графових нейронних мереж є складною і багаторівневою задачею. Основу цієї системи становить класифікаційний компонент моделі ІКДЦ, реалізований у вигляді графово-нейромережевої моделі, яка базується на сучасних методах глибокого навчання і відповідає за ідентифікацію спам-повідомлень. Особливістю архітектури цієї моделі є використання графів, що складаються з вузлів та зв'язків між ними. Кожен вузол і зв'язок графа відповідає певному параметру або характеристиці, отриманій з аналізованих вебформ, що дозволяє ефективно знаходити й аналізувати приховані закономірності у даних.

Основою графової моделі є вузли, які реалізуються за допомогою класу `Node`, та зв'язки між ними, представлені класом `Edge`. У цій архітектурі кожен вузол відповідає конкретному повідомленню, яке користувач надіслав через вебформу. Цей вузол описується спеціальним набором характеристик (вектором ознак), який містить інформацію про ключові параметри подання. Зв'язки ж відображають різноманітні відносини між окремими повідомленнями - наприклад, подібність у вмісті текстів чи близькість у часі надсилання.

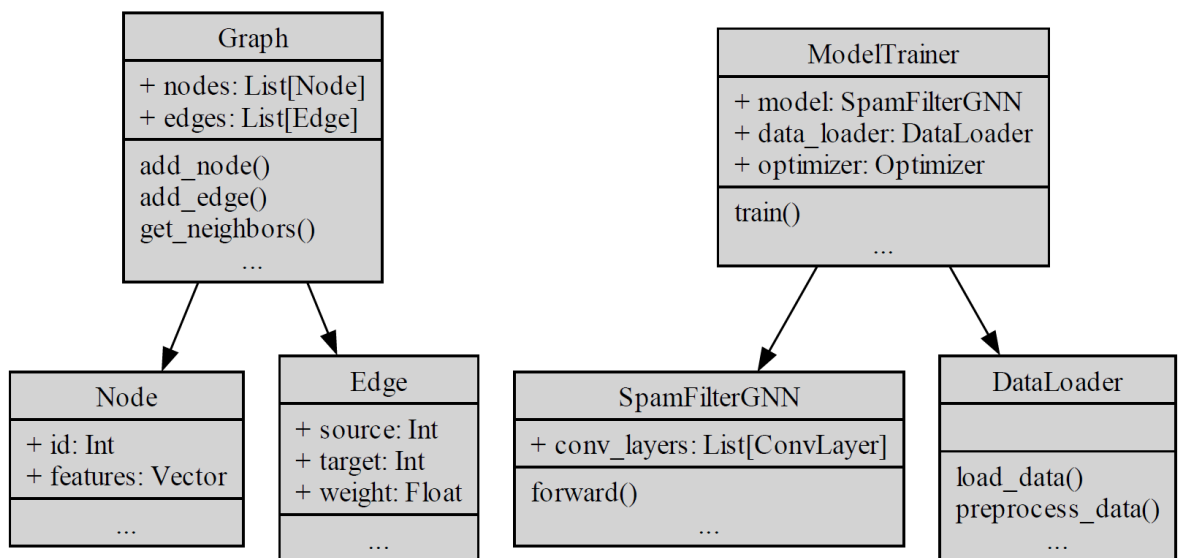


Рис. 4.8. Архітектурна схема класів класифікаційного компонента моделі ІКДЦ для фільтрації вебспаму

Управління цією структурою здійснюється за допомогою класу `Graph`, який відповідає за збереження і зв'язування вузлів та зв'язків у єдину цілісну систему. Це дає змогу зручно та ефективно обробляти дані, що надходять, і використовувати їх

під час подальшого тренування моделі. Для безпосередньої роботи нейронної мережі передбачено класи SpamFilterGNN, що формує власне нейронну архітектуру, і ModelTrainer, який виконує навчання моделі, взаємодіючи із вхідними даними через клас DataLoader. Така організація класів дозволяє не тільки ефективно керувати великими масивами інформації, але й забезпечує точність у виявленні спам-повідомлень завдяки врахуванню всіх важливих взаємозв'язків між поданнями.

Ключовим компонентом запропонованої системи є клас SpamFilterGNN, основним призначенням якого є класифікація окремих повідомлень на предмет того, чи належать вони до спаму. Цей клас являє собою нейронну мережу, яка аналізує структуру графа, оцінюючи зв'язки між вузлами. При цьому використовуються графові згорткові шари (Graph Convolutional Layers, GCN), які дозволяють вузлам враховувати інформацію своїх сусідів під час класифікації.

Основні методи класу SpamFilterGNN включають ініціалізацію шарів моделі, зокрема два графові згорткові шари. Конструктор відповідає за створення та налаштування шарів, які спочатку отримують початкові дані, далі формують приховане представлення, а потім генерують результат класифікації. Конструктор класу виглядає наступним чином:

```
class SpamFilterGNN(nn.Module):
    def __init__(self, in_features, out_features, hidden_features):
        super(SpamFilterGNN, self).__init__()
        self.conv1 = GCNConv(in_features, hidden_features)
        self.conv2 = GCNConv(hidden_features, out_features)
```

Таким чином, завдяки використанню GCN-шарів модель здатна ефективно враховувати контекстну інформацію з сусідніх вузлів, що забезпечує високу точність класифікації спамових повідомлень.

Функція `forward()` реалізує прямий прохід через мережу, де кожен вузол отримує інформацію від своїх сусідів через графові конволюційні шари.

```
def forward(self, x, edge_index):
    # Перший графовий конволюційний шар
    x = F.relu(self.conv1(x, edge_index))
    # Другий графовий конволюційний шар
    x = self.conv2(x, edge_index)
    return F.log_softmax(x, dim=1)
```

Процес навчання включає дві основні стадії: прямий прохід, коли кожен вузол оновлює свої характеристики, та зворотний прохід, під час якого відбувається

оптимізація параметрів мережі. Функція `forward()` забезпечує проходження інформації через нейронну мережу і виконує головну роль у класифікації вузлів графа. На цьому етапі кожен вузол отримує та обробляє дані не лише свої власні, але й інформацію з сусідніх вузлів, використовуючи графові згорткові шари.

Клас `DataLoader` відповідає за ефективну підготовку даних перед початком навчання нейронної мережі. Його основне завдання - отримати вихідну інформацію з вебформ, виділити важливі характеристики та перетворити їх у формат, придатний для обробки графовою нейронною моделлю. Клас `DataLoader` допомагає системі перетворювати початкові дані на оптимальний формат, забезпечуючи швидке й точне тренування моделі, цей клас має дві головні функції. Перша функція `load_data()` - призначений для зчитування даних, які будуть використовуватися для побудови графа (включаючи вузли та зв'язки між ними). Реалізація цієї функції буде наступною:

```
class DataLoader:
    def load_data(self, path):
        raw_data = read_data(path)
        return raw_data
```

Друга функція - `preprocess_data()` - виконує попередню підготовку даних, тобто очищення, нормалізацію та векторизацію. Після цих операцій отримані дані готові для подачі безпосередньо на вхід нейронної мережі:

```
def preprocess_data(self, raw_data):
    processed_data = vectorize(raw_data)
    return processed_data
```

Клас `ModelTrainer` відповідає за ефективне навчання моделі, взаємодіючи безпосередньо з нейронною мережею, даними, а також забезпечуючи налаштування її параметрів. Цей клас виконує дві ключові функції - тренування нейронної мережі та оцінку її результативності на тестових даних. Функція `train()` реалізує повний цикл навчання моделі, здійснюючи послідовні проходи через дані. В процесі тренування виконується прямий прохід, обчислюється похибка моделі, після чого параметри оновлюються за допомогою обраного оптимізатора. Ось як виглядає типова реалізація цієї функції:

```
class ModelTrainer:
    def __init__(self, model, data_loader, optimizer):
        self.model = model
        self.data_loader = data_loader
```

```

self.optimizer = optimizer

def train(self, epochs):
    for epoch in range(epochs):
        # Завантажуємо дані та виконуємо попередню обробку
        raw_data = self.data_loader.load_data()
        processed_data = self.data_loader.preprocess_data(raw_data)

        # Прямий прохід (отримуємо результат роботи моделі)
        output = self.model(processed_data['features'], processed_data['edge_index'])

        # Обчислюємо похибку
        loss = F.nll_loss(output, processed_data['labels'])

        # Виконуємо зворотний прохід і оновлюємо параметри моделі
        self.optimizer.zero_grad()
        loss.backward()
        self.optimizer.step()

```

Другий метод, `evaluate()`, дозволяє перевірити точність роботи моделі, застосовуючи її до спеціально виділеного набору тестових даних. Він обчислює похибку класифікації, завдяки чому можна визначити, наскільки ефективно модель розпізнає спам. Приклад методу:

```

def evaluate(self, test_data):
    self.model.eval()
    with torch.no_grad():
        output=self.model(test_data['features'], test_data['edge_index'])
        loss = F.nll_loss(output, test_data['labels'])
    return loss

```

Попередні UML-схеми (Рис. 4.5–4.8) розкривають окремі частини інтегрованого контуру: контроль доступу, незмінне журналювання, підготовку даних і графово-нейромережеву класифікацію. Для відображення їх узгодженої роботи в межах єдиної програмної архітектури подано зведену UML-діаграму конвеєра обробки подій і журналювання (Рис. 4.9). Вона показує, як критична подія надходить у систему, проходить канонізацію, формування ознак, побудову контекстного графа, оцінювання ризику, застосування політики реагування, формування рішення та запис результату в незмінний журнал.

Архітектура програмної реалізації підсистеми довіри ІКДЦ спроектована як керований конвеєр обробки подій, у якому події вебформ і транзакційні події проходять однакові етапи нормалізації, оцінювання ризику та незмінного закріплення результатів. Такий підхід дозволяє розглядати рішення модуля виявлення загроз не як локальний висновок моделі машинного навчання, а як формалізовану подію безпеки, що підлягає доказовій фіксації та подальшій перевірці. Реалізація виконана

мовою Python, а структура компонентів організована навколо чітко визначених доменних об'єктів, ядра конвеєра, підсистеми незмінного журналювання та циклу навчання і версіонування моделі. Базовим елементом потоку є об'єкт `SecurityEvent`, який задає уніфіковане подання події незалежно від її типу. У структурі події фіксуються ідентифікатор, тип події (`SUBMIT` або `TX`), час, суб'єкт ініціації, сесія, ресурс звернення, контекст та корисне навантаження. Таке подання забезпечує, з одного боку, придатність до інтеграції у вебсередовище, а з іншого, визначає мінімально достатній набір атрибутів для формування ознак, побудови графових зв'язків та проведення аудиту. Подальша обробка події спирається на принцип однозначності, тому наступним доменним артефактом виступає `CanonicalEvent`, який містить канонічний рядок події та версію правил канонізації. Версія канонізації є суттєвою для відтворюваності, оскільки саме вона визначає, які перетворення застосовано до часових міток, категоріальних значень та структури полів, і тим самим задає стабільну основу для криптографічного відбитка.

Центральним елементом зведеної архітектури є клас `SecurityPipeline`, який координує повний маршрут обробки критичної події. На вхід конвеєра надходить об'єкт `SecurityEvent`, що задає уніфіковане подання події вебсистеми незалежно від її типу. Метод `process(e: SecurityEvent): PolicyDecision` відображає основний сценарій роботи контуру: подія приймається на обробку, проходить послідовні етапи аналізу, після чого система формує рішення політики реагування.

Клас `EventCanonicalizer` відповідає за канонізацію події, тобто приведення її до однозначного структурованого подання. Це необхідно для того, щоб одна й та сама подія мала стабільне представлення під час подальшого формування ознак, оцінювання ризику та журналювання. Клас `FeatureProcessor` формує ознаковий опис події через метод `features(e): object`, перетворюючи вхідні атрибути події на структуру, придатну для подальшого аналізу.

Контекстна складова події відображена класом `ContextGraph`, який формує графовий знімок або контекст події через метод `snapshot(e): object`. Цей компонент забезпечує врахування не лише локальних атрибутів події, а й її зв'язків з іншими подіями, користувачами, ресурсами або транзакційними сценаріями. Отримані

ознаки та контекст передаються до класу `RiskModel`, який виконує оцінювання ризику за допомогою методу `infer(ctx, feat): object`.

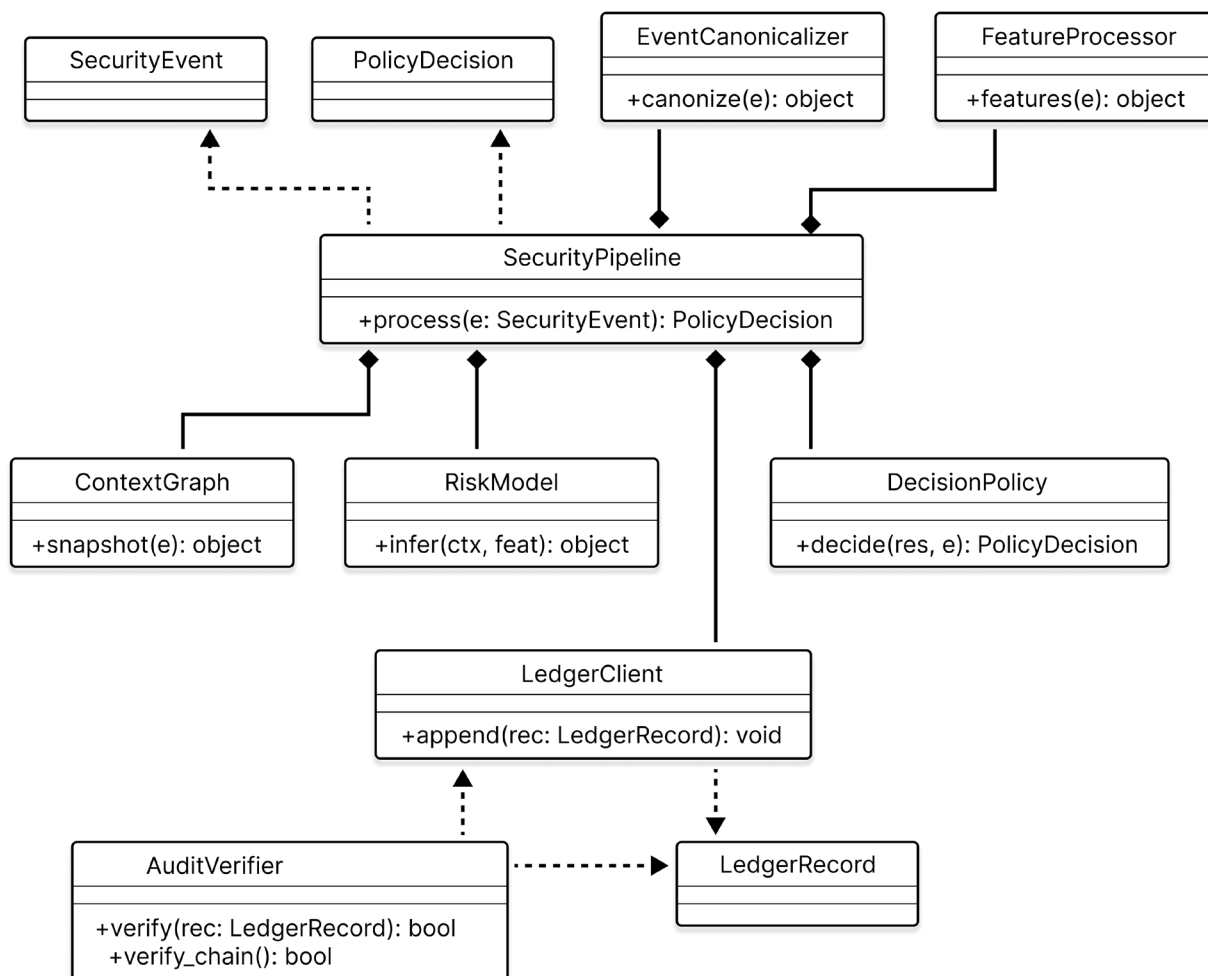


Рис. 4.9 UML-діаграма зведеного конвеєра обробки подій, прийняття рішень та журналювання в інтегрованому контурі ІКДЦ

Результат оцінювання ризику передається до класу `DecisionPolicy`. Цей клас реалізує правила реагування та перетворює результат аналізу у кероване рішення через метод `decide(res, e)` в `PolicyDecision`. Об'єкт `PolicyDecision` фіксує підсумковий результат обробки події, наприклад дозвіл, перевірку, блокування або іншу дію відповідно до політик безпеки.

Для доказової фіксації результатів використовується клас `LedgerClient`, який додає сформований запис до журналу через метод `append(rec: LedgerRecord): void`. Об'єкт `LedgerRecord` подає журналізований запис, що пов'язує подію, результат оцінювання ризику, прийняте рішення та службові дані, необхідні для подальшої

перевірки. Контроль коректності журналювання забезпечує клас AuditVerifier, який підтримує перевірку окремого запису за допомогою `verify(rec: LedgerRecord): bool` та перевірку цілісності всього ланцюга через `verify_chain(): bool`.

Таким чином, архітектура та програмна реалізація інтегрованого контуру довіри й цілісності демонструють, як модель ІКДЦ і розроблені методи можуть бути узгоджені в єдиному програмному конвеєрі обробки критичних подій. Запропонована структура поєднує доменне подання події, канонізацію, криптографічне закріплення, формування ознак, графово-нейромережевий інференс, політики реагування, незмінне журналювання, аудитну перевірку та версіонування моделей. Завдяки цьому рішення системи безпеки не залишається лише результатом роботи класифікатора, а перетворюється на відтворюваний і перевірюваний артефакт, пов'язаний із конкретною подією, версією моделі, станом ознак, політикою реагування та записом у журналі. Це забезпечує цілісність програмного контуру, простежуваність рішень і можливість подальшої аудитної перевірки у вебсистемах.

4.4. Експериментальна перевірка моделі ІКДЦ та запропонованих методів

Експериментальну перевірку підсистеми довіри ІКДЦ організовано як відтворюване порівняльне дослідження, у якому одночасно оцінюються дві групи властивостей. Перша група характеризує якість виявлення загроз у потоці подій вебформ SUBMIT та транзакцій TX за наявності графового контексту. Друга група характеризує доказовість і незмінність рішень, коли кожний висновок супроводжується криптографічними артефактами й фіксується в ledger так, щоб під час аудиту можна було відтворити подію, версію моделі, параметри політики та підсумковий клас ризику. Експериментальне середовище відповідає конвеєру обробки подій, у якому виконуються канонізація, хешування та підпис, формування ознак, побудова графового знімка у часовому вікні спостереження, інференс графової моделі та застосування політики реагування з формуванням незмінного запису. Для подій типу TX використано відкритий набір PaySim [132], який містить синтетичні мобільні грошові транзакції на основі реальних журналів сервісу мобільних платежів.

Для перевірки поведінки графової частини на транзакційних зв'язках застосовано Elliptic Data Set [133], який надає транзакційний граф і мітки класів для задачі виявлення нелегітимних операцій у криптовалютних мережах. Для валідації стійкості метрик на табличних ознаках транзакцій використано публічно описаний датасет операцій платіжних карт із мітками шахрайства, що є типовим еталоном для задач fraud у високодисбалансованих вибірках. Потік SUBMIT-подій сформовано як журнал взаємодії з вебформами, де кожне відправлення має текстовий вміст, технічний контекст і часові характеристики, достатні для відновлення зв'язків між подіями в межах одного користувацького оточення. Така постановка дозволяє оцінити не лише точність виявлення загроз, а й практичну придатність інтегрованого контуру ІКДЦ з погляду доказовості, затримки, пропускну здатності та можливості аудиторної перевірки.

Підготовка даних виконана зі збереженням причинно-часового характеру потокового застосування. Розділення на train, validation і test виконано часовими зрізами, коли навчання відбувається на більш ранніх інтервалах, а оцінювання на пізніших, що усуває витік інформації між майбутніми подіями та параметрами моделі.

На етапі побудови графа кожна подія є вузлом відповідного типу, а ребра формуються за правилами спільності учасників, сесій, технічних маркерів і близькості в часі у часовому вікні спостереження. Така постановка перевіряє ключову гіпотезу, що ризик визначається не лише локальними ознаками події, а й її положенням у структурі взаємодій. Якість детекції оцінюється метриками precision, recall і F1-міра, але інтерпретація виконується через рішення політики реагування. Для кожної події формується одна з трьох керованих дій дозволити, перевести в контрольовану перевірку, заблокувати. Пропуск загрози означає, що подія, яка є загрозою, отримала дію дозволити або була лише відкладена на перевірку без блокування. Хибне блокування означає, що легітимна подія була заблокована. Для узагальненої оцінки використано F1-міру, яка набуває значень у діапазоні [0,1], де 1 відповідає ідеальній якості. F1-міра обчислюється через кількості рішень:

$$F1 = \frac{2TP}{2TP + FP + FN}, \quad (4.23)$$

де TP , FP , FN є цілими невід’ємними числами, TP - це кількість подій загрози, які коректно віднесено до загрози або коректно заблоковано. FP - кількість легітимних подій, які помилково віднесено до загрози або помилково заблоковано. FN - кількість подій загрози, які не були виявлені як загроза, тобто не були заблоковані. За цією постановкою позитивним класом вважається подія загрози, а всі інші стани трактується як “не загроза” на рівні підрахунку TP, FP, FN . Оскільки в контурі безпеки одна й та сама помилка може мати різну практичну ціну залежно від дії політики, підсумкову ефективність додатково оцінено через інтегральну функцію очікуваних втрат. Вона поєднує втрати від пропущених загроз, втрати від хибних блокувань, вартість контрольованої перевірки, втрати від невиявленого втручання в журнал та штраф за додаткову затримку, пов’язану з канонізацією, криптографією і записом у ledger. Узагальнений вигляд такої функції зберігає лінійну інтерпретованість у термінах операційних витрат:

$$C = c_{FN}^{allow} FN_{allow} + c_{FN}^q FN_q + c_{FP}^{block} FP_{block} + c_q Q + c_{tamper} (T_{total} - T_{det}) + c_{lat} N \Delta \tau, \quad (4.24)$$

де C є сумарним показником втрат у вибраній шкалі, а більші значення відповідають гіршому результату. Усі коефіцієнти c_{FN}^{allow} , c_{FN}^q , c_{FP}^{block} , c_q , c_{tamper} , c_{lat} є невід’ємними дійсними числами, що задаються як параметри експерименту для приведення різних типів втрат до спільної шкали. Коефіцієнт c_{tamper} визначає вагу втрат від одного невиявленого втручання в журнал, а множник $(T_{total} - T_{det})$ задає кількість таких невиявлених втручань. Величини FN_{allow} , FN_q , FP_{block} , Q , T_{total} , T_{det} , N є цілими невід’ємними числами, де FN_{allow} - кількість загроз, які помилково пройшли з дією дозволу, FN_q - кількість загроз, які не були заблоковані і потрапили лише в режим контрольованої перевірки, FP_{block} - кількість легітимних подій, які помилково заблоковано, Q - кількість подій, переведених у режим контрольованої перевірки, T_{total} - кількість змодельованих спроб модифікації журналу, T_{det} - кількість виявлених втручань, а N - загальна кількість оброблених подій у тестовому інтервалі. Величина $\Delta \tau$ є додатковою затримкою на одну подію, спричиненою контуром доказовості, і вимірюється в одиницях часу. Добуток $N \Delta \tau$ характеризує сумарний додатковий час, а коефіцієнт c_{lat} переводить його у втрати в тій самій шкалі, що й

інші компоненти. Покращення визначається як відносне зменшення очікуваних втрат порівняно з базовим підходом, де базою виступає система без графового контексту або без ledger-компонента:

$$Imp = \frac{C_{base} - C_{model}}{C_{base}} \cdot 100, \quad (4.25)$$

де Imp - відносне покращення у відсотках, $C_{base} > 0$ - значення зведеної функції операційних втрат для базової конфігурації, $C_{model} \geq 0$ - значення цієї самої функції для досліджуваної конфігурації. Обидві величини обчислюються за формулою (4.24) в однаковій шкалі втрат. Додатне значення Imp означає зменшення втрат відносно бази, нульове - відсутність покращення, а від'ємне - погіршення результату.

Перевірку властивостей незмінності та відтворюваності виконано як окремий блок експерименту, орієнтований на аудит. Для кожного запису ledger відновлюється канонічне подання події, повторно обчислюється її хеш і перевіряється підпис, після чого контролюється відповідність ідентифікаторів версії моделі та політики, а також узгодженість ланцюга записів у механізмі зв'язування блоків. Кількісним показником чутливості журналу до модифікацій використано міру:

$$TDR = \frac{T_{det}}{T_{total}}, \quad (4.26)$$

де TDR - коефіцієнт виявлення порушень цілісності журналу, T_{det} - кількість виявлених втручань, T_{total} - загальна кількість змодельованих спроб модифікації журналу, причому $T_{total} > 0$ і $0 \leq T_{det} \leq T_{total}$. Отже, $TDR \in [0,1]$. Значення $TDR = 1$ означає, що всі змодельовані втручання були виявлені, а $TDR = 0$ - що жодне втручання не було виявлене.

Порівняльне оцінювання виконувалось для кількох конфігурацій, які відображають послідовне ускладнення контуру прийняття рішень. Базовою лінією виступала правилна конфігурація без графового контексту, що формує рішення переважно за локальними ознаками одиначної події. Наступний рівень представляла графова модель, яка враховує контекст зв'язків між подіями, але не використовує окремий інтегральний сигнал транзакційного ризику. Третя конфігурація доповнювала ознаковий опис транзакцій числовим сигналом R_{TX} , який відображає агрегований ризик операції та підсилює транзакційний канал у спільному просторі

ознак. Повний контур ІКДЦ поєднував контекстне графове оцінювання ризику з доказовою фіксацією результатів у незмінному журналі, включно з прив'язкою рішення до ідентифікатора версії моделі та параметрів політики реагування. Для забезпечення коректності порівняння пороги реагування τ_1 і τ_2 фіксувалися та не змінювалися між конфігураціями (Табл. 4.2).

Таблиця 4.2.

Параметри експерименту та узагальнені показники ефекту для порівнюваних конфігурацій

Показник	Значення
τ_1	0.350000
τ_2	0.750000
Приріст F1-міри для SUBMIT відносно правиллової конфігурації, %	17.948718
Приріст F1-міри для TX відносно правиллової конфігурації, %	21.621622
Зростання p95 відносно GNN, %	23.076923
Приріст F1 для TX після додавання R_{TX} , %	3.529412

Якість детекції оцінювали через precision, recall і F1-міра, а також через частку хибних спрацювань FPR , оскільки для прикладної безпеки важливо одночасно зменшувати пропуски загроз і не створювати надмірного обсягу помилкових блокувань та перевірок. Підсумкові метрики подано окремо для контуру подій SUBMIT і TX, що дозволяє порівнювати поведінку системи на “комунікаційних” і “операційних” подіях в однаковій шкалі показників. Як видно з результатів, перехід від конфігурації на основі правил до графової суттєво підвищує F1 і одночасно зменшує FPR в обох доменах, що відповідає гіпотезі про користь контекстного аналізу для виявлення скоординованої активності. Повний контур ІКДЦ демонструє додаткове покращення метрик і стабільніше зниження частки хибних спрацювань, що є важливим для зменшення навантаження на ручну верифікацію (Табл. 4.3).

Таблиця 4.3.

Порівняння якості детекції для класів подій SUBMIT і TX за різними конфігураціями

Тип події	Конфігурація	Точність	Повнота	F1-міра	Частка хибних спрацювань, %
SUBMIT	Rule WAF	0.83	0.74	0.78	2.6
SUBMIT	GNN	0.91	0.89	0.90	1.2
SUBMIT	ІКДЦ	0.93	0.91	0.92	0.9
TX	Rule WAF	0.79	0.70	0.74	1.9
TX	GNN	0.89	0.87	0.88	1.0
TX	ІКДЦ	0.91	0.89	0.90	0.8

Дані, наведені в Табл. 4.3, показують, що для обох типів подій спостерігається послідовне покращення показників детекції при переході від правилкової конфігурації до графової моделі та повного контуру ІКДЦ. Для подій SUBMIT ця тенденція візуалізована на Рис. 4.10, що дозволяє порівняти зміну точності, повноти та F1-міри між конфігураціями.

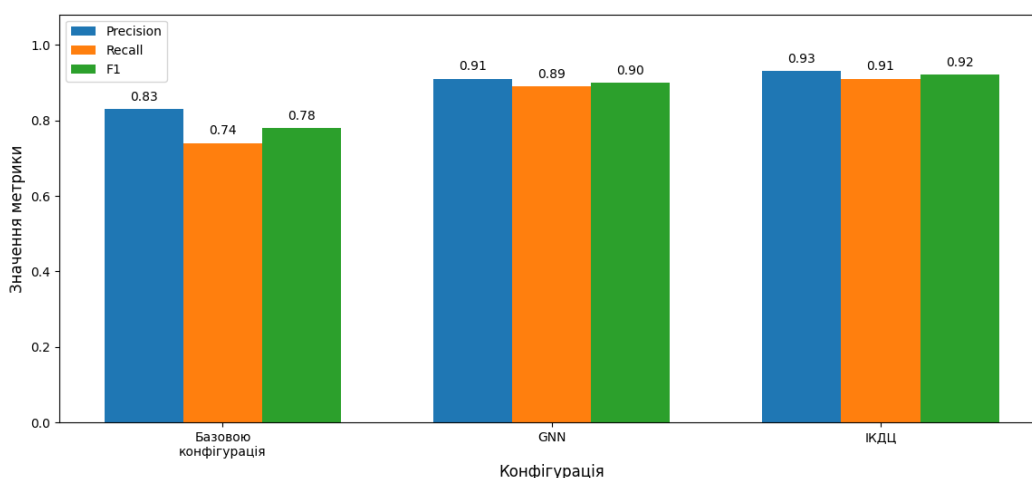


Рис. 4.10. Порівняння Precision, Recall і F1 для подій SUBMIT за різними конфігураціями

Як видно з Рис. 4.10, у контурі подій SUBMIT найбільше покращення відбувається при переході від базової правилкової конфігурації до GNN, тоді як повний контур ІКДЦ забезпечує додаткове підвищення всіх трьох метрик. Аналогічне порівняння для транзакційних подій TX наведено на Рис. 4.11.

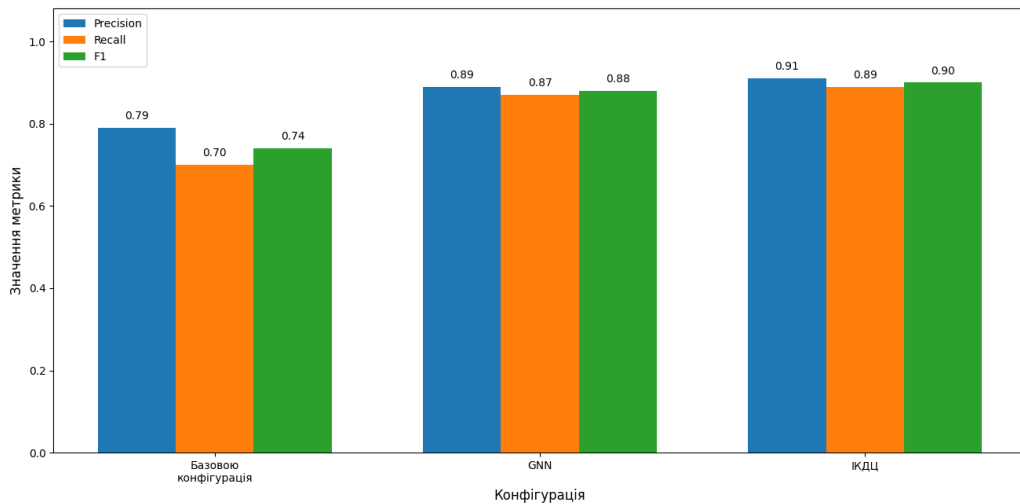


Рис. 4.11. Порівняння Precision, Recall і F1 для подій TX за різними конфігураціями

Результати, подані на Рис. 4.11, підтверджують, що для транзакційних подій TX повний контур ІКДЦ також демонструє найкраще поєднання Precision, Recall і F1-міри. Однак для безпекового контуру важливо оцінювати не лише якість виявлення загроз, а й рівень помилкового реагування на легітимні події, тому порівняння частки хибних спрацювань подано окремо на Рис. 4.12.

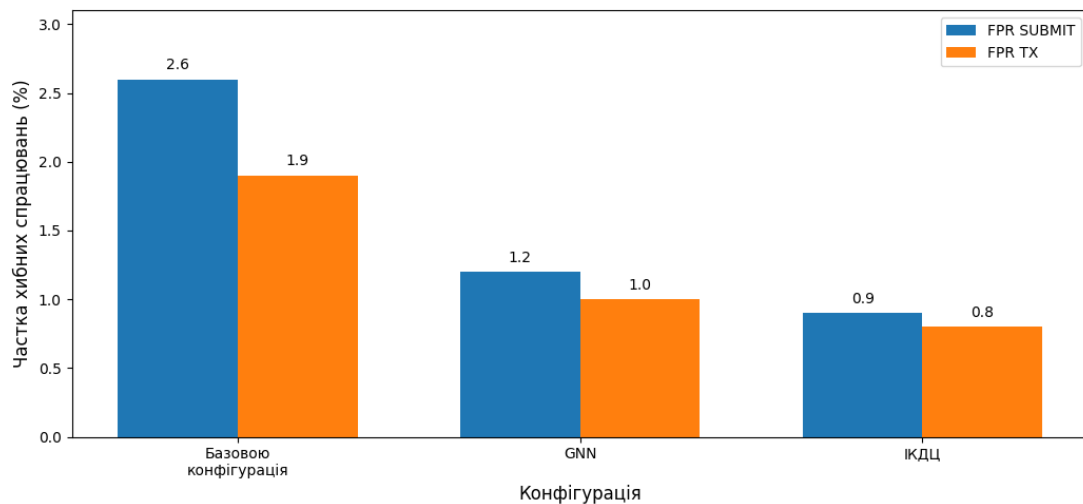


Рис. 4.12 Порівняння частки хибних спрацювань FPR для SUBMIT і TX за різними конфігураціями

За результатами, наведеними на Рис. 4.12, частка хибних спрацювань зменшується як для подій SUBMIT, так і для подій TX при переході до складніших конфігурацій. Це означає, що підвищення F1-міри в Табл. 4.3 досягається не за

рахунок надмірного блокування легітимних подій, а завдяки точнішому розмежуванню загроз і нормальної активності.

Для узагальнення впливу отриманих результатів на рівень довіри до рішень введено похідний індекс довіри до рішення, який поєднує якість виявлення загроз та рівень помилкового реагування на легітимні події. Оскільки $F1$ -міра характеризує збалансованість точності та повноти, а FPR відображає частку хибних спрацювань, індекс довіри до рішення для подій типу x подано у вигляді:

$$D_x = F1_x \cdot \left(1 - \frac{FPR_x}{100}\right),$$

де D_x - індекс довіри до рішення для подій типу x ; $F1_x$ - $F1$ -міра для відповідного типу подій, FPR_x - частка хибних спрацювань у відсотках. Такий показник використовується як додаткове узагальнення для інтерпретації довіри до результату, його зміст полягає в тому, що рішення вважається більш довіреним тоді, коли система одночасно краще виявляє загрози та рідше створює помилкові спрацювання для легітимних подій. Для подій SUBMIT за отриманими даними (Табл. 4.3) отримаємо:

$$D_{SUBMIT}^{Rule} = 0.78 \cdot \left(1 - \frac{2.6}{100}\right) = 0.78 \cdot 0.974 = 0.760,$$

$$D_{SUBMIT}^{IKDC} = 0.92 \cdot \left(1 - \frac{0.9}{100}\right) = 0.92 \cdot 0.991 = 0.912.$$

Відносне підвищення індексу довіри для подій SUBMIT становить:

$$\Delta D_{SUBMIT} = \frac{D_{SUBMIT}^{IKDC} - D_{SUBMIT}^{Rule}}{D_{SUBMIT}^{Rule}} \cdot 100\% = \frac{0.912 - 0.760}{0.760} \cdot 100\% = 20.0\%.$$

Для подій TX маємо:

$$D_{TX}^{Rule} = 0.74 \cdot \left(1 - \frac{1.9}{100}\right) = 0.74 \cdot 0.981 = 0.726,$$

$$D_{TX}^{IKDC} = 0.90 \cdot \left(1 - \frac{0.8}{100}\right) = 0.90 \cdot 0.992 = 0.893.$$

Відносне підвищення індексу довіри для подій (TX) становить:

$$\Delta D_{TX} = \frac{D_{TX}^{IKDC} - D_{TX}^{Rule}}{D_{TX}^{Rule}} \cdot 100\% = \frac{0.893 - 0.726}{0.726} \cdot 100\% = 23.0\%.$$

У порівнянні з типовою конфігурацією індекс довіри до рішення підвищено з 0.760 до 0.912, що відповідає приросту 20.0% для подій SUBMIT та з 0.726 до 0.893,

що відповідає приросту 23.0% для подій TX (Рис.4.13). Це підтверджує, що підвищення довіри до результатів роботи моделі ІКДЦ досягається не лише завдяки зростанню F1-міри, а й завдяки одночасному зменшенню частки хибних спрацювань. Саме поєднання цих двох факторів знижує ризик помилкового рішення адміністратора та підвищує обґрунтованість реакції системи на критичні події.

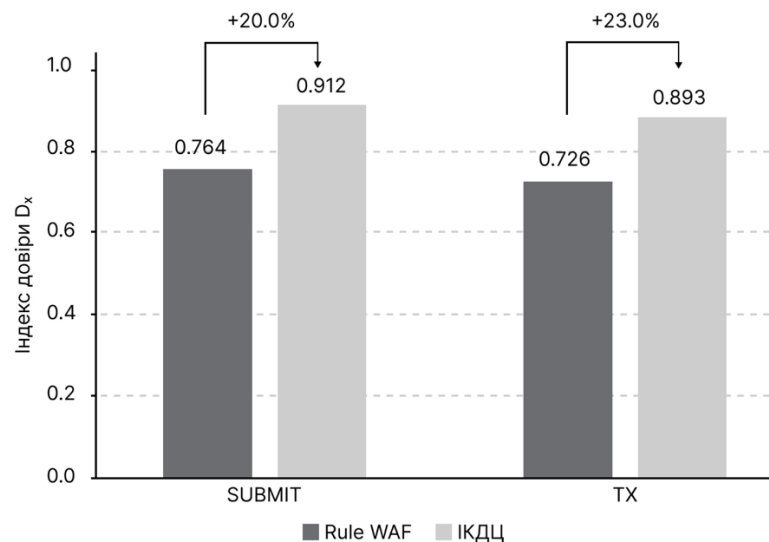


Рис. 4.13 Індекс довіри до рішення для подій SUBMIT, TX

Внесок інтегрального транзакційного ризику R_{TX} оцінювали шляхом порівняння конфігурацій із цим сигналом і без нього. Результати показують, що введення R_{TX} підсилює транзакційний канал навіть без підключення незмінного журналювання. Це проявляється у зростанні якості виявлення подій класу загрози THREAT у транзакціях і в зменшенні частки хибних спрацювань. Повний контур ІКДЦ додатково підвищує значення F1 і забезпечує найнижчий FPR серед порівнюваних варіантів. Така тенденція узгоджується з постановкою, де транзакції розглядаються не як паралельний незалежний потік, а як повноцінне джерело сигналів у спільному графовому контексті, тому зі зростанням складності конфігурації спостерігається послідовне підвищення F1 для класу загрози у транзакціях (Рис. 4.13). Додатково було виконано окремий зріз для транзакційних подій, який прямо демонструє внесок інтегрального ризикового сигналу R_{TX} . Такий зріз дозволяє відокремити ефект графового контексту від ефекту посилення транзакційного каналу через введення інтерпретованої ознаки ризику. Результати

показують, що конфігурація GNN без R_{TX} підвищує якість детекції порівняно з базовою конфігурацією на основі правил, однак додавання R_{TX} дає додатковий стабільний приріст метрик і водночас зменшує частку хибних спрацювань. Повний контур ІКДЦ демонструє найкраще співвідношення між F1 і FPR, що узгоджується з прийнятим підходом, у межах якого транзакції інтегруються в один графовий контекст із подіями вебформ, а не обробляються як ізольований канал (Табл. 4.4).

Таблиця 4.4.

Вплив інтегрального транзакційного ризику R_{TX} на якість детекції подій TX

Система	Використання R_{TX}	Точність	Повнота	F1-міра	Частка хибних спрацювань, %
Rule WAF	False	0.79	0.70	0.74	1.9
GNN No RTX	False	0.86	0.84	0.85	1.2
GNN With RTX	True	0.89	0.87	0.88	1.0
ІКДЦ	True	0.91	0.89	0.90	0.8

Як видно з табл. 4.4, використання інтегрального транзакційного ризику R_{TX} забезпечує додаткове покращення якості детекції подій TX порівняно з графовою моделлю без цього сигналу. Найвище значення F1-міри досягається у повному контурі ІКДЦ, що свідчить про послідовне підсилення транзакційного каналу. Наочне порівняння цих значень наведено на Рис. 4.14.

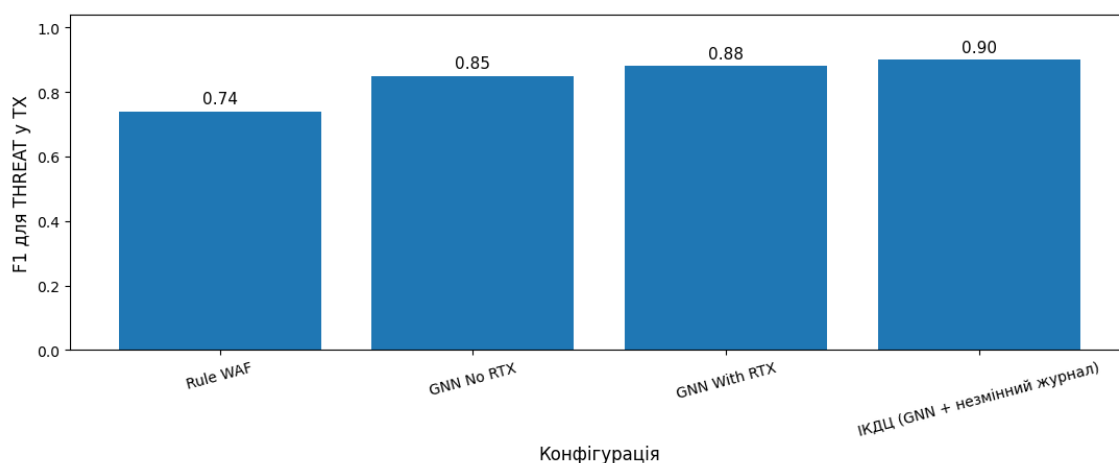


Рис. 4.14 Якість виявлення загроз у транзакціях TX за метрикою F1 для порівнюваних конфігурацій

Окрім точності детекції, критичною характеристикою є часові параметри роботи контуру, оскільки підсилення доказовості неминуче додає накладні витрати на формування криптографічних артефактів і запис результату в незмінний журнал.

Для оцінювання затримок використовувалися перцентилі, де р50 відповідає медіанному часу обробки, а р95 характеризує “верхню” затримку, в яку вкладаються 95% подій. Такий показник важливий для потокових систем, оскільки описує стабільність реакції не тільки в середньому, а й у навантажених режимах. Згідно з вимірами, перехід від GNN до повного контуру збільшує р95 з 52 до 64 мс, що відповідає оціненим накладним витратам на рівні близько 23%. При цьому час, що безпосередньо витрачається на запис у журнал і перевірку запису під час аудиту, залишається в межах одиниць мілісекунд на операцію, що пояснює керований характер загальних накладних витрат. Зниження пропускної здатності є очікуваним наслідком більш складної обробки, однак отримані значення залишаються придатними для потокових сценаріїв, де ключовим є контроль р95 і передбачуваність реакції (Табл. 4.5).

Таблиця 4.5.

Часові характеристики та пропускна здатність порівнюваних конфігурацій

Система	р50, мс	р95, мс	Швидкість обробки, подій/с	Запис у журнал, мс	Перевірка в аудиті, мс
Rule WAF	12	28	2400	0	0
GNN	18	52	1600	0	0
ІКДЦ (GNN + незмінний журнал)	23	64	1350	8	6

Окремим етапом перевірено поведінку системи в різних режимах експлуатації, які імітують типові стани навантаження та атаки. Розглянуто нормальний потік подій, хвилю бот-активності у вебформах, сплеск шахрайської транзакційної активності та комбінований сценарій, у якому одночасно присутні обидва типи загроз. У таких умовах важливо, щоб рішення залишалися не лише точними, а й вкладалися в допустимі часові межі, оскільки затримка напряму впливає на користувацький досвід і на можливість оперативного блокування. Порівняння за р95 демонструє монотонне зростання часу рішення при переході від конфігурації на основі правил до графової і далі до повного контуру, але збереження передбачуваності в усіх сценаріях (Табл.4.6). Найскладнішим є комбінований режим, де р95 для повного контуру досягає 98 мс, що

відображає підвищене навантаження на контекстний аналіз і журналювання в умовах одночасних атак у двох каналах (Рис. 4.15).

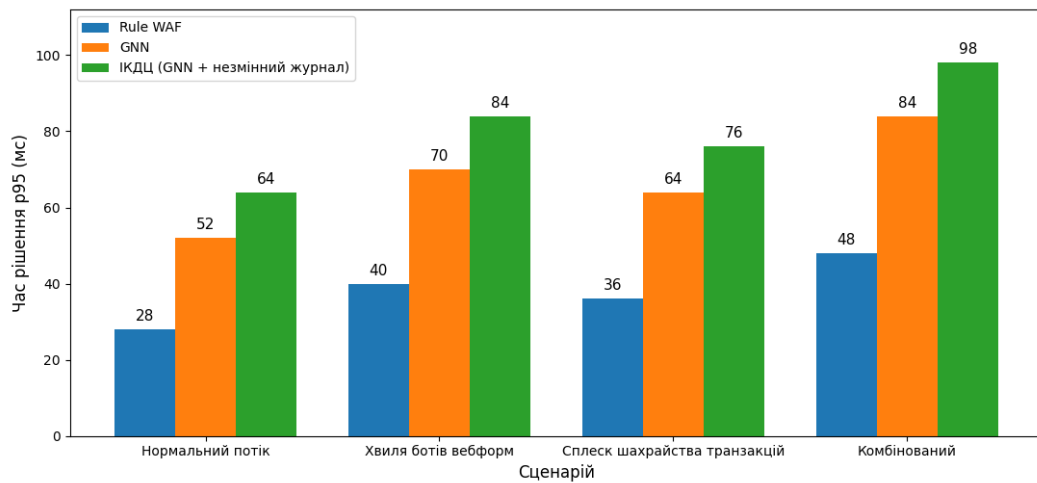


Рис. 4.15. Порівняння p95 затримки рішення за сценаріями навантаження для Rule WAF, GNN та ІКДЦ (GNN + незмінний журнал)

Загалом результати узгоджуються з очікуваним розподілом внеску компонентів, оскільки контекстний графовий аналіз забезпечує основний внесок у підвищення F1 і одночасно зменшує FPR відносно базової конфігурації на основі правил, що скорочує як пропуски загроз, так і навантаження на ручну верифікацію. Додавання R_{TX} підсилює транзакційний канал і дає додаткове покращення порівняно з графовою основою. Незмінне журналювання додає помірні накладні витрати за затримкою та пропускну здатністю, але натомість забезпечує відтворюваність і можливість аудиторної перевірки через фіксацію хешу події, підпису, версії моделі та параметрів політики, що є критичним для розслідування інцидентів і контролю змін моделі в часі.

Таким чином, експериментальна перевірка підтвердила доцільність поєднання графово-нейромережевого оцінювання, інтегрального транзакційного ризику та незмінного журналювання в межах інтегрованого контуру ІКДЦ. Порівняння з конфігурацією правил показало зростання точності, повноти та F1-міри для подій SUBMIT і TX, а також зменшення частки хибних спрацювань. Додавання ризикового сигналу R_{TX} посилює транзакційний канал, тоді як незмінне журналювання

забезпечило відтворюваність і доказовість прийнятих рішень. Отримане збільшення затримки залишається керованим і компенсується можливістю аудитної перевірки, фіксації версій моделі, параметрів політики реагування та контролю цілісності журналу. Це підтверджує практичну придатність запропонованого інтегрованого методу для задач виявлення загроз, простежуваності подій і забезпечення довіри у вебсистемах.

Висновки до розділу 4

У четвертому розділі вперше розроблено метод інтегрованого забезпечення довіри й цілісності у вебсистемах, що ґрунтується на моделі ІКДЦ, методі блокчейн-верифікованого журналювання критичних подій і контролю доступу та методі графово-нейромережевого виявлення вебспаму й підозрілої активності. Метод також базується на теорії композиції функціональних відображень критичних подій у клас рішень, що забезпечує узгоджене перетворення події у ризикову оцінку, рішення, дію реагування та доказовий запис.

Розроблений метод забезпечує узгоджену взаємодію між контуром криптографічної фіксації подій, контуром інтелектуального виявлення загроз, політиками реагування та незмінним журналюванням. За рахунок цього результати класифікації, ознаковий опис події, версія моделі, параметри політики реагування та підсумкове рішення фіксуються як відтворювані й перевірювані артефакти. Це дозволяє одночасно забезпечити контроль цілісності даних, простежуваність критичних подій, доказовість журналів та підвищення точності прийняття рішень у вебсистемі.

Показано, що поєднання канонізації подій, хешування, цифрового підпису, графового контекстного оцінювання, інтегрального аналізу транзакційного ризику, трирівневої класифікації та незмінного журналювання формує єдиний контур безпекової обробки. У межах такого контуру критична подія проходить повний цикл, від первинної фіксації та формування ознак до оцінювання ризику, вибору політики реагування, запису в незмінний журнал і подальшої аудитної перевірки.

Експериментальна перевірка підтвердила ефективність використання інтегрованого контуру ІКДЦ для виявлення загроз у потоках подій типу SUBMIT і TX. Для подій SUBMIT значення F1-міри зросло з 0.78 у базовій конфігурації правил до 0.90 для графово-нейромережевої конфігурації та до 0.92 для повного інтегрованого контуру ІКДЦ. Водночас частка хибних спрацювань зменшилася з 2.6% до 1.2% і 0.9% відповідно. Для подій TX значення F1-міри зросло з 0.74 до 0.88 і 0.90, а частка хибних спрацювань зменшилася з 1.9% до 1.0% і 0.8% відповідно.

Додатково встановлено, що включення транзакційного ризику R_{TX} підсилює транзакційний канал аналізу, для подій TX F1-міра зросла з 0.85 до 0.88, а частка хибних спрацювань зменшилася з 1.2% до 1.0%. Перехід від графово-нейромережевої конфігурації до повного інтегрованого контуру супроводжується збільшенням часу обробки 95% подій з 52 мс до 64 мс, що відповідає накладним витратам близько 23%. Водночас такі витрати залишаються керованими, оскільки забезпечують додаткову доказовість, відтворюваність рішень, контроль цілісності журналу та можливість аудитної перевірки.

Таким чином, у четвертому розділі розроблено та експериментально перевірено метод інтегрованого забезпечення довіри й цілісності у вебсистемах, що поєднує модель ІКДЦ, метод блокчейн-верифікованого журналювання критичних подій і контролю доступу, метод графово-нейромережевого виявлення вебспаму й підозрілої активності та композицію функціональних відображень критичних подій у клас рішень. Результати проведених досліджень свідчать про повне виконання завдань розділу та підтверджують практичну придатність розроблених програмних і алгоритмічних рішень для забезпечення цілісності системи, простежуваності критичних подій, відтворюваності рішень і підвищення точності їх прийняття у вебсистемах, які функціонують у динамічних середовищах.

Висновки

У результаті дисертаційного дослідження вирішено актуальне наукове завдання щодо розроблення та обґрунтування моделей і методів забезпечення довіри, цілісності та надійності вебсистем на рівні архітектури вебзастосунків. Запропонований підхід ґрунтується на поєднанні незмінного журналювання критичних подій, криптографічної верифікації, графового подання подій та методів машинного навчання для автоматизованого виявлення вебспау, підозрілої активності й аномалій вебтрафіку. Розв'язання поставленого завдання має значення для розвитку спеціальності 121 «Інженерія програмного забезпечення», оскільки стосується моделей, методів, архітектурних рішень, програмних компонентів і процесів забезпечення якості, надійності та захищеності програмного забезпечення.

1. На основі проведеного аналізу сучасного стану досліджень у сфері забезпечення безпеки вебсистем встановлено, що наявні підходи переважно розв'язують задачі контролю цілісності, аудитної перевірності, журналювання критичних подій та інтелектуального виявлення загроз ізольовано. Показано, що відсутність єдиного формального контуру, який би поєднував подання критичних подій, механізми незмінного журналювання, графове оцінювання ризику, політики реагування та аудитну перевірку, обмежує доказовість рішень і ускладнює забезпечення довіри до результатів обробки подій у вебсистемах.

2. Вперше розроблено модель інтегрованого контуру довіри й цілісності у вебсистемі, що ґрунтується на кортежно-графовому поданні критичних подій і криптографічних принципах їх верифікації. Модель формалізує подання подій вебформ, SQL-операцій, транзакційних дій, пов'язаних із ними сутностей, відношень між ними, процедур формування ознак, оцінювання ризику, політик реагування та незмінного журналу подій і рішень. За рахунок цього модель ІКДЦ забезпечує єдине інформаційне середовище для контролю цілісності даних, простежуваності подій, аудитної перевірки та відтворюваності рішень у вебсистемі.

3. Вперше розроблено метод блокчейн-верифікованого журналювання критичних подій і контролю доступу у вебсистемах, що ґрунтується на моделі ІКДЦ

та теорії криптографічно зв'язаного ланцюга подій із хешуванням, цифровим підписом і пороговим правилом прийняття рішення щодо доступу. Метод забезпечує фіксацію критичних подій і рішень доступу в незмінному журналі, дозволяє перевіряти цілісність записів, виявляти приховану модифікацію інформації, підвищувати доказовість журналів і посилювати контроль цілісності даних під час розслідування інцидентів.

4. Вперше розроблено метод графово-нейромережевого виявлення вебспаму та підозрілої активності у вебсистемах, що ґрунтується на моделі ІКДЦ та багатопредставленому графовому описі подій, поданих через систему ознак технічного, змістового, часово-поведінкового та контекстного характеру з урахуванням зв'язків між подіями й результатами аналітичного оцінювання. Метод забезпечує розрізнення легітимних, підозрілих і шкідливих звернень, підвищує точність виявлення вебспаму та зменшує частку хибних спрацювань.

5. Вперше розроблено метод інтегрованого забезпечення довіри й цілісності у вебсистемах, що ґрунтується на моделі ІКДЦ, методі блокчейн-верифікованого журналювання критичних подій і контролю доступу та методі графово-нейромережевого виявлення вебспаму й підозрілої активності. Метод також базується на теорії композиції функціональних відображень критичних подій у клас рішень. Це забезпечує узгоджене перетворення критичної події у ризикову оцінку, клас рішення, дію реагування та доказовий запис, що підвищує цілісність системи, простежуваність подій і точність прийняття рішень у вебсистемі.

6. Результати дисертаційної роботи реалізовано у вигляді програмних засобів для підтримки інтегрованого контуру довіри й цілісності, блокчейн-верифікованого журналювання критичних подій, контролю доступу та графово-нейромережевого виявлення вебспаму й підозрілої активності. За результатами впровадження розробленої системи у ТОВ «ШЛІФАРБ» узагальнений показник якості автоматичного розпізнавання повідомлень зріс з 78% до 92%, а частка хибних спрацювань зменшилася з 2,6% до 0,9%. За результатами впровадження у ТОВ «АРМА МОТОРС КИЇВ» узагальнений показник якості виявлення ризикових ситуацій зріс з 74% до 90%, частка хибних спрацювань

зменшилася з 1,9% до 0,8%, а повнота виявлення MITM-атак на канали обміну транзакційними подіями склала 98%. Результати дисертаційного дослідження також впроваджено в Інституті програмних систем Національної академії наук України при формуванні плану перспективних наукових досліджень. Контрольне оцінювання показало, що застосування методу інтегрованого забезпечення довіри й цілісності у вебсистемах дозволяє підвищити якість оцінювання критичних подій з 85% до 96%, а частку хибних спрацювань зменшити з 3% до 0,8%.

7. Експериментальна перевірка підтвердила практичну придатність запропонованого інтегрованого контуру ІКДЦ для обробки подій типу SUBMIT і TX. Для подій SUBMIT значення F1-міри зросло з 0,78 у базовій конфігурації правил до 0,90 для графово-нейромережевої конфігурації та до 0,92 для повного інтегрованого контуру ІКДЦ. Частка хибних спрацювань при цьому зменшилася з 2,6% до 1,2% і 0,9% відповідно. Для подій TX значення F1-міри зросло з 0,74 до 0,88 і 0,90, а частка хибних спрацювань зменшилася з 1,9% до 1,0% і 0,8% відповідно. Додавання інтегрального транзакційного ризику R_{TX} забезпечило додаткове підсилення транзакційного каналу аналізу, а незмінне журналювання забезпечило доказовість, відтворюваність і аудитну перевірність прийнятих рішень.

8. Достовірність одержаних результатів забезпечується використанням положень теорії графів, теорії довіри до інформаційних систем, криптографічних методів хешування та цифрового підпису, підходів блокчейн-верифікованого журналювання, методів машинного навчання, зокрема графових нейронних мереж, а також методів математичної статистики, теорії ймовірностей і планування експерименту. Достовірність також підтверджується узгодженістю теоретичних положень з результатами програмної реалізації, експериментальної перевірки та актами впровадження у практичну діяльність підприємств, наукову діяльність установи та освітній процес.

Мету дослідження щодо підвищення рівня довіри, цілісності та безпеки архітектури вебсистем за рахунок побудови моделей і методів контролю цілісності даних, фіксації критичних подій, адаптивного виявлення підозрілої активності та незмінного журналювання рішень досягнуто. Усі поставлені наукові завдання

виконано. Одержані результати є внеском у розвиток моделей, методів і технологій програмної інженерії, орієнтованих на забезпечення довіри, цілісності, функціональної стійкості, надійності, аудитної перевірності та адаптивного виявлення загроз у вебсистемах.

Перспективними напрямками подальших досліджень є розширення моделі ІКДЦ на багатодоменні та міжсервісні вебсередовища, удосконалення механізмів пояснюваності графово-нейромережових рішень, розвиток методів інкрементального та федеративного навчання для оновлення моделей без порушення вимог приватності, а також побудова масштабованих механізмів незмінного журналювання та аудитної перевірки для високоінтенсивних потоків критичних подій у розподілених вебсистемах.

Список використаної літератури

1. Tanriverdi, M., Tekerek, A. (2020). Implementation of Blockchain Based Distributed Web Attack Detection Application. Feminist Press at CUNY
2. Zhai, Z., Shen, S., Mao, Y. (2022). BPKI: A secure and scalable blockchain-based public key infrastructure system for web services. Journal of Information Security and Applications, DOI: 10.1016/j.jisa.2022.103226
3. Chondrogiannis, E., Andronikou, V., Karanastasis, E., Litke, A., Varvarigou, T. (2022). Using blockchain and semantic web technologies for the implementation of smart contracts between individuals and health insurance organizations. Blockchain: Research and Applications, DOI: 10.1016/j.bcra.2021.100049
4. Kumar, A. S. (2021). Efficient sensitivity orient blockchain encryption for improved data security in cloud. Concurrent Engineering Research and Applications, DOI: 10.1177/1063293X211008586
5. Hsiao, S.-J., Sung, W.-T. (2021). Employing Blockchain Technology to Strengthen Security of Wireless Sensor Networks. IEEE Access, DOI: 10.1109/ACCESS.2021.3079708
6. Aini, Q., Rahardja, U., Tangkaw, M. R., Santoso, N. P. L., Khoirunisa, A. (2020). Embedding aBlockchain Technology Pattern Into the QR Code for an Authentication Certificate. JOIN (Jurnal Online Informatika), DOI: 10.15575/join.v5i2.583
7. Trung, T. T. (2020). Intelligent CRM systems of transport companies. Amazonia Investiga, DOI: 10.34069/AI/2020.26.02.47
8. Besançon, L., Ferreira Da Silva, C., Ghodous, P., Gelas, J.-P. (2022). A Blockchain Ontology for DApps Development. IEEE Access, DOI: 10.1109/ACCESS.2022.3173313
9. Chan, K. C., Zhou, X., Gururajan, R., Ally, M., Gardiner, M. (2020). Integration of Blockchains with Management Information Systems. 2019 International Conference on Mechatronics, Robotics and Systems Engineering (MoRSE), DOI: 10.1109/MoRSE48060.2019.8998694

10. Kriuchenkov, O., Morozova, O., Kharchenko, V., Tetskyi, A., Storchak, K. (2023). Development of a web system for recognizing the images taken by UAV. 2022 12th International Conference on Dependable Systems, Services and Technologies (DESSERT), DOI: 10.1109/DESSERT58054.2022.10018785
11. Chen, C., Zhang, L., Li, Y., Liao, T., Zhao, S., Zheng, Z., Huang, H., Wu, J. (2022). When Digital Economy Meets Web3.0: Applications and Challenges. IEEE Open Journal of the Computer Society, DOI: 10.1109/OJCS.2022.3217565
12. Abdallah, S., Nizamuddin, N. (2023). Blockchain-based solution for Pharma Supply Chain Industry. Computers & Industrial Engineering, DOI: 10.1016/j.cie.2023.108997
13. Aydos, M., Aldan, Ç., Coşkun, E., Soydan, A. (2022). Security testing of web applications: A systematic mapping of the literature. Journal of King Saud University - Computer and Information Sciences, DOI: 10.1016/j.jksuci.2021.09.018
14. Karamchandani, A., Srivastava, S. K., Abha, Srivastava, A. (2023). A lower approximation based integrated decision analysis framework for a blockchain-based supply chain. Computers & Industrial Engineering, DOI: 10.1016/j.cie.2023.109092
15. Ryu, J., Son, S., Lee, J., Park, Y. (2022). Design of Secure Mutual Authentication Scheme for Metaverse Environments Using Blockchain. IEEE Access, DOI: 10.1109/ACCESS.2022.3206457
16. Gaur, R., Prakash, S., Kumar, S., Abhishek, K., Msahli, M. (2022). A Machine-Learning-Blockchain-Based Authentication Using Smart Contracts for an IoHT System. Sensors, DOI: 10.3390/s22239074
17. Asim, J., Khan, A. S., Saqib, R. M., Abdullah, J., Ahmad, Z., Honey, S., Afzal, S., Alqahtani, M. S., Abbas, M. (2022). Blockchain-based Multifactor Authentication for Future 6G Cellular Networks: A Systematic Review. Applied Sciences, DOI: 10.3390/app12073551
18. Prakash, R., Anoop, V.S., Asharaf, S. (2022). Blockchain technology for cybersecurity: A text mining literature analysis. International Journal of Information Management Data Insights, DOI: 10.1016/j.jjime.2022.100112

19. Aini, Q., Manongga, D., Rahardja, U., Sembiring, I., Elmanda, V., Faturahman, A., Santoso, N. P. L. (2022). Security Level Significance in DApps Blockchain-Based Document Authentication. Aptisi Transactions on Technopreneurship, DOI: 10.34306/att.v4i3.277
20. Nagabhooshanam, N., Bala sundara ganapathy, N., Ravindra Murthy, C., Al Ansari Mohammed Saleh, CosioBorda, R. F. (2023). Neural network based single index evaluation for SQL injection attack detection in health care data. Measurement: Sensors. DOI: 10.1016/j.measen.2023.100779
21. Devalla, V., Srinivasa Raghavan, S., Maste, S., Kotian, J. D., Annapurna, D. (2022). mURLi: A Tool for Detection of Malicious URLs and Injection Attacks. Procedia Computer Science, 215, 662-676. DOI: 10.1016/j.procs.2022.12.068
22. Liu, G. (2022, August 29). The Application of Data Encryption Technology in Computer Network Communication Security. Mobile Information Systems. DOI: 10.1155/2022/3632298
23. Chen, W., Chen, G., Zhao, Y., Zhang, J. (2021). Security vulnerability and encryption technology of computer information technology data under big data environment. Journal of Physics: Conference Series. DOI: 10.1088/1742-6596/1800/1/012012
24. Awadallah, R., Samsudin, A. (2021, October 4). Using Blockchain in Cloud Computing to Enhance Relational Database Security. IEEE Access. DOI: 10.1109/ACCESS.2021.3117733
25. Alghawazi, M., Alghazzawi, D., Alarifi, S. (2022, September 20). Detection of SQL Injection Attack Using Machine Learning Techniques: A Systematic Literature Review. DOI: 10.3390/jcp2040039
26. Sakharkar, S. (2023, October). Systematic Review: Analysis of Coding Vulnerabilities across Languages. Journal of Information Security. DOI: 10.4236/jis.2023.144019
27. Zaman, S., Alhazmi, K., Aseeri, M. A., Ahmed, M. R., Khan, R. T., Kaiser, M. S. (2021, June 16). Security Threats and Artificial Intelligence Based Countermeasures for

Internet of Things Networks: A Comprehensive Survey. IEEE Access. DOI: 10.1109/ACCESS.2021.3089681

28. Alouffi, B., Hasnain, M., Alharbi, A., Alosaimi, W., Alyami, H., Ayaz, M. (2021, April 14). A Systematic Literature Review on Cloud Computing Security: Threats and Mitigation Strategies. IEEE Access. DOI: 10.1109/ACCESS.2021.3073203

29. Li, J., Kassem, M. (2021, December). Applications of distributed ledger technology (DLT) and Blockchain-enabled smart contracts in construction. Automation in Construction, 103955. DOI: 10.1016/j.autcon.2021.103955

30. Queralta, J. P., Keramat, F., Salimi, S., Fu, L., Yu, X., Westerlund, T. (2023, September 29). Blockchain and Emerging Distributed Ledger Technologies for Decentralized Multi-robot Systems. Current Robotics Reports. DOI: 10.1007/s43154-023-00101-3

31. Tanwar, S., Gupta, N., Iwendi, C., Kumar, K., Alenezi, M. (2022, August 24). Next Generation IoT and Blockchain Integration. Journal of Sensors. DOI: 10.1155/2022/9077348

32. Hussein, K. M., Al-Gailani, M. F. (2023, August 30). Evaluation Performance of Bloom Filter in Blockchain Network. Iraqi Journal of Information and Communications Technology. DOI: 10.31987/ijict.6.1.204

33. Kalajdjieski, J., Raikwar, M., Arsov, N., Velinov, G., Gligoroski, D. (2023, March). Databases fit for blockchain technology: A complete overview. Blockchain: Research and Applications. DOI: 10.1016/j.bcra.2022.100116

34. Przytarski, D., Stach, C., Gritti, C., Mitschang, B. (2021, December 21). Query Processing in Blockchain Systems: Current State and Future Challenges. Security and Privacy in Blockchains and the IoT. DOI: 10.3390/fi14010001

35. Hu, Vincent C. Blockchain for Access Control Systems. Computer Security Division Information Technology Laboratory, December 2021. DOI: 10.6028/NIST.IR.8403-draft

36. Namane, Sarra; Ben Dhaou, Imed. Blockchain-Based Access Control Techniques for IoT Applications. Security and Privacy in Blockchain/IoT, 15 June 2022. DOI: 10.3390/electronics11142225

37. Awan, Samia Masood; Azad, Muhammad Ajmal; Arshad, Junaid; Waheed, Urooj; Sharif, Tahir. A Blockchain-Inspired Attribute-Based Zero-Trust Access Control Model for IoT. *Pervasive Computing in IoT*, 16 February 2023. DOI: 10.3390/info14020129
38. Aliya, Barakova; Olga, Ussatova; Yenlik, Begimbayeva; Sogukpinar, Ibrahim. Ensuring Information Security of Web Resources Based on Blockchain Technologies. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 2023. DOI: 10.14569/IJACSA.2023.0140689
39. Shi, Jinshan; Li, Ru; Hou, Wenhan. A Mechanism to Resolve the Unauthorized Access Vulnerability Caused by Permission Delegation in Blockchain-Based Access Control. *IEEE Access*, 24 August 2020. DOI: 10.1109/ACCESS.2020.3018783
40. Четверіков, І. О.; Petrenko, A. I. BLOCKCHAIN TECHNOLOGY IN THE INFORMATION SECURITY SYSTEM. ДВНЗ «Київський національний університет імені Вадима Гетьмана», 2020. DOI: 10.33111/mise.99.14
41. Mahmood, Samreen; Chadhar, Mehmood; Firmin, Selenia. Cybersecurity Challenges in Blockchain Technology: A Scoping Review. *School of Engineering, Information Technology and Physical Sciences*, 05 Apr 2022. DOI: 10.1155/2022/7384000
42. Taylor, Paul J.; Dargahi, Tooska; Dehghantanha, Ali; Parizi, Reza M.; Choo, Kim-Kwang Raymond. A systematic literature review of blockchain cyber security. *Digital Communications and Networks*, May 2020. DOI: 10.1016/j.dcan.2019.01.005
43. Wylde, Vinden; Rawindaran, Nisha; Lawrence, John; Balasubramanian, Rushil; Prakash, Edmond; Jayal, Ambikesh; Khan, Imtiaz; Hewage, Chaminda; Platts, Jon. Cybersecurity, Data Privacy and Blockchain: A Review. *SN Computer Science*, 12 January 2022. DOI: 10.1007/s42979-022-01020-4
44. Javaid, Uzair; Jameel, Furqan; Javaid, Umair; Khan, Muhammad Toaha Raza; Jäntti, Riku. Rogue Device Mitigation in the Internet of Things: A Blockchain-Based Access Control Approach. *Mobile Information Systems*, October 2020. DOI: 10.1155/2020/8831976
45. Attkan A., Ranga V. Cyber-physical security for IoT networks: a comprehensive review on traditional, blockchain and artificial intelligence based key-

security . Complex & Intelligent Systems. - 2022. - 24 лютого. DOI 10.1007/s40747-022-00667-z

46. Latif S. A., Xian Wen F. B., Iwendi C., Wang L.-l. F., Mohsin S. M., Han Z., Band S. S. AI-empowered, blockchain and SDN integrated security architecture for IoT network of cyber physical systems. Computer Communications. - 2022. Vol. 181. - С. 274- 283. - 1 січня. DOI: 10.1016/j.comcom.2021.09.029

47. Bonfanti M. E. Artificial intelligence and the offense-defense balance in cyber security. Cyber Security Politics; Socio-Technological Transformations and Political Fragmentation. - 2022. - 16 лютого. DOI: 10.4324/9781003110224-6

48. Naik B., Mehta A., Yagnik H., Shah M. The impacts of artificial intelligence techniques in augmentation of cybersecurity: a comprehensive review. Complex & Intelligent Systems. - 2021. - 24 серпня. DOI: 10.1007/s40747-021-00494-8

49. Abdullahi M., Baashar Y., Alhussian H., Alwadain A., Aziz N., Capretz L. F., Abdulkadir S. J. Detecting Cybersecurity Attacks in Internet of Things Using Artificial Intelligence Methods: A Systematic Literature Review. Electronics. - 2022. - 11(2), 198. - 10 січня. DOI: 10.3390/electronics11020198

50. Ahanger T. A., Aljumah A., Atiquzzaman M. State-of-the-art survey of artificial intelligent techniques for IoT security. Computer Networks. - 2022. Vol. 206. – 7 квітня. DOI: 10.1016/j.comnet.2022.108771

51. Ramasamy L. K., Khan F., Shah M., Prasad B. V. V. S., Iwendi C., Biamba C. Secure Smart Wearable Computing through Artificial Intelligence-Enabled Internet of Things and Cyber-Physical Systems for Health Monitoring. Smart Healthcare Systems Based on the Internet of Things and Artificial Intelligence. - 2022. - 29 січня. DOI: 10.3390/s22031076

52. Ghillani D. Deep Learning and Artificial Intelligence Framework to Improve the Cyber Security [Electronic resource]. 2022. DOI: 10.22541/au.166379475.54266021/v1

53. Pise A. A., Almuzaini K. K., Ahanger T. A., Farouk A., Pant K., Pareek P. K., Nuagah S. J. Enabling Artificial Intelligence of Things (AIoT) Healthcare Architectures and Listing Security Issues. Computational Intelligence and Neuroscience. - 2022. - 03 серпня. DOI: 10.1155/2022/8421434

54. Zhang Z., Al Hamadi H., Damiani E., Yeun C. Y., Taher F. Explainable Artificial Intelligence Applications in Cyber Security: State-of-the-Art in Research. IEEE Access. - 2022. - 05 вересня. DOI: 10.1109/ACCESS.2022.3204051
55. Gill S. S., Xu M., Ottaviani C., Patros P., Bahsoon R., Shaghaghi A., Golec M., Stankovski V., Wu H., Abraham A., Singh M., Mehta H., Ghosh S. K., Baker T., Parlikad A. K., Lutfiyya H., Kanhere S. S., Sakellariou R., Dustdar S., Rana O., Uhlig S. AI for next generation computing: Emerging trends and future directions. Internet of Things. - 2022. - Vol. 19. - 12 березня. DOI: 10.1016/j.iot.2022.100514
56. Kumar S., Lim W. M., Sivarajah U., Kaur J. Artificial Intelligence and Blockchain Integration in Business: Trends from a Bibliometric-Content Analysis. Information Systems Frontiers. - 2023. - Vol. 25. - С. 871-896. - 12 квітня 2022. DOI: 10.1007/s10796-022-10279-0
57. Yathiraju N. Investigating the use of an Artificial Intelligence Model in an ERP Cloud-Based System. International Journal of Electrical, Electronics and Computers. - 2022. - Vol. 7, Issue 2. - Mar-Apr. - 30 квітня. <http://dx.doi.org/10.22161/eec.72.1>
58. Sujith A.V.L.N., Sajja G. S., Mahalakshmi V., Nuhmani S., Prasanalakshmi B. Systematic review of smart health monitoring using deep learning and Artificial intelligence. Neuroscience Informatics. - 2022. - Vol. 2, Issue 3. - Вересень. DOI: 10.1016/j.neuri.2021.100028
59. Nasim S. F., Ali M. R., Kulsoom U. ARTIFICIAL INTELLIGENCE INCIDENTS & ETHICS: A NARRATIVE REVIEW. Computer Science and Information Technology, Ned University, Pakistan. - 2022. - Листопад. DOI: 10.54489/ijtim.v2i2.80
60. Kunduru A. R. ARTIFICIAL INTELLIGENCE ADVANTAGES IN CLOUD FINTECH APPLICATION SECURITY. CENTRAL ASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES. - 2023. Vol. 4, No. 8. - С. 48-53. - 14 серпня
61. Chang V., Bhavani V. R., Xu A. Q., Hossain M. A. An artificial intelligence model for heart disease detection using machine learning algorithms. Healthcare Analytics. - 2022. - Листопад. DOI: 10.1016/j.health.2022.100016

62. Babitha M., Sushama C., Gudivada V. K., Kazi K. S. L., Bandaru S. R. Trends of Artificial Intelligence for Online Exams in Education. International Journal of Early Childhood Special Education (INT-JECSE). - 2022. - Травень. DOI: 10.9756/INT- JECSE/V14I1.290
63. Esenogho E., Djouani K., Kurien A. M. Integrating Artificial Intelligence Internet of Things and 5G for Next-Generation Smartgrid: A Survey of Trends Challenges and Prospect. IEEE Access. - 2022. - 06 січня. DOI: 10.1109/ACCESS.2022.3140595
64. Bi S., Wang C., Zhang J., Huang W., Wu B., Gong Y., Ni W. A Survey on Artificial Intelligence Aided Internet-of-Things Technologies in Emerging Smart Libraries. AI-Aided Wireless Sensor Networks and Smart Cyber-Physical Systems. - 2022. 13 квітня. DOI: 10.3390/s22082991
65. Nallamothu P.T., Khan M.S. Machine Learning for SPAM Detection. Asian Journal of Advances in Research. 17.03.2023
66. Ahmed N., Amin R., Aldabbas H., Koundal D., Alouffi B., Shah T. Machine Learning Techniques for Spam Detection in Email and IoT Platforms: Analysis and Research Challenges. Security, Trust, and Privacy in Machine Learning and Internet of Things. 30.11.2021. DOI: 10.1155/2022/1862888
67. Ullah M.U. et al. Intelligent Intrusion Detection System for APACHE WEB SERVER Empowered with Machine Learning Approaches. International Journal of Computational and Innovative Sciences. 30.03.2022
68. Lin C. et al. VulEye: A Novel Graph Neural Network Vulnerability Detection Approach for PHP Application. Applied Sciences. 06.01.2023. DOI: 10.3390/app13020825.
69. Tubishat M. et al. An Improved Dandelion Optimizer Algorithm for Spam Detection: Next-Generation Email Filtering System. Computers. 28.09.2023. DOI: 10.3390/computers12100196
70. Venugopal I.V.S., Bhaskari D.L., Seetaramanath M.N. "Detection of Severity-based Email SPAM Messages using Adaptive Threshold Driven Clustering". International Journal of Advanced Computer Science and Applications(IJACSA). 2022. Вип. 10. DOI: 10.14569/IJACSA.2022.0131040

71. Mageshkumar N. et al. Efficient spam filtering through intelligent text modification detection using machine learning. Materials Today: Proceedings. 19.07.2022. DOI: 10.1016/j.matpr.2022.05.364
72. Ji K., Kwon Y. "New Spam Filtering Method with Hadoop Tuning-Based MapReduce Naïve Bayes". Computer Systems Science & Engineering. 16.08.2022. DOI: 10.32604/csse.2023.031270
73. Dash B. et al. Threats and Opportunities with AI-Based Cyber Security Intrusion Detection: A Review. International Journal of Software Engineering & Applications (IJSEA), Vol.13, No.5. Вересень 2022
74. Bharadiya J.P. "AI-Driven Security: How Machine Learning Will Shape the Future of Cybersecurity and Web 3.0". American Journal of Neural Networks and Applications. 10.06.2023. DOI: 10.11648/j.ajnna.20230901.11
75. Zhurakovskiy B., Averichev I., Shakhmatov I. Using the Latest Methods of Cluster Analysis to Identify Similar Profiles in Leading Social Networks. CEUR Workshop Proceedings. - 2023. - Vol. 3646. - P. 116-126
76. Шахматов І. О., Замрій І. В. Технологія blockchain як інструмент протидії неправомірному доступу до вебсайтів. Інженерія програмного забезпечення і передові інформаційні технології (SoftTech-2023) : матеріали IV міжнар. наук.-практ. конф. молодих вчених та студентів (Київ, 19-21 груд. 2023 р.). - Київ, 2023. С. 360- 364.
77. Zamrii I., Shakhmatov I., Yaskevych V. BlockchainSQLSecure: Integration of Blockchain to Strengthen Protection Against SQL Injections. Bulletin of Taras Shevchenko National University of Kyiv. Physics and Mathematics. - 2024. - Vol. 78, No. 1. P. 160- 168. DOI: 10.17721/1812-5409.2024/1.29
78. Шахматов І. О. Технологія Blockchain як інструмент протидії неправомірному використанню доступу до вебсайтів. Зв'язок. - 2024. - № 1. - С. 20- 25. DOI: 10.31673/2412-9070.2024.012025
79. Замрій І. В., Шахматов І. О. Потенціал блокчейну у покращенні безпеки вебсайтів. Сучасний захист інформації. - 2024. - № 1(57). - С. 28-38. DOI: 10.31673/2409-7292.2024.010004

80. Замрій І.В., Шахматов І.О. Підвищення безпеки вебзастосунків через інноваційні патерни інтеграції штучного інтелекту. Сучасний стан наукових досліджень та технологій в промисловості. - 2024. - № 1(27). - С. 67-80. DOI: 10.30837/ITSSI.2024.27.067.

81. Zamrii I., Shakhmatov I., Yudin O. et al. Methods for Detecting DDoS Attacks in Web Traffic Using Autoencoders with an Adaptive Three-Level Approach. 2024 IEEE 5th International Conference on Advanced Trends in Information Theory (ATIT) (Lviv, Ukraine, 2024). - 2024. - P. 1-5. DOI: 10.1109/ATIT64324.2024.11222524.

82. Замрій І.В., Шахматов І.А. Посилення безпеки вебідентифікації через технологію блокчейн. Всеукраїнська науково-технічна конференція «Застосування програмного забезпечення в інформаційно-комунікаційних технологіях», 24 квітня 2024 року, Державний університет інформаційно-комунікаційних технологій. Збірник тез. К.: ДУІКТ, 2024. С. 249-253.

83. Шахматов І. О., Замрій І. В. Використання блокчейн-технології для підвищення безпеки від SQL ін'єкцій. Безпека інформаційних технологій: ITSec-2024 : матеріали XIII міжнар. наук.-техн. конф. (Львів, 9-11 трав. 2024 р.). - Львів, 2024. - С. 98-101.

84. Shakhmatov I., Zamrii Ir. Application of Graph Neural Networks for Effective Automation of Web Spam Filtering. Математика. Інформаційні технології. Освіта : матеріали XIII міжнар. наук.-практ. конф. (Луцьк-Світязь, 31 трав. - 2 черв. 2024 р.). - 2024.

85. Шахматов І. О., Замрій І. В. Технології масштабування даних у боротьбі з DDoS-атаками. Штучний інтелект і безпека : матеріали наук.-практ. конф. (19-21 листоп. 2024 р.). - 2024. - С. 27-30

86. Шахматов І. О., Юр'єв А. Л. Система персоналізованих рекомендацій для підвищення ефективності продажів телекомунікаційного обладнання на основі штучного інтелекту. Штучний інтелект і безпека : матеріали наук.-практ. конф. (19-21 листоп. 2024 р.). - 2024. - С. 8-9.

87. Білодід Д. В., Шахматов І. О. Захист фронтенд-компонентів від XSS-атак за допомогою Content Security Policy. Виклики та рішення в програмній інженерії : матеріали всеукр. наук.-техн. конф. (26 листоп. 2024 р.). - 2024. - С. 67-71
88. Білодід Д. В., Шахматов І. О. Ефективність CSRF-токенів у запобіганні міжсайтовим запитам у фронтенд-додатках. Виклики та рішення в програмній інженерії : матеріали всеукр. наук.-техн. конф. (26 листоп. 2024 р.). - 2024. - С. 92-96
89. Шахматов І. О. Інтегрований контур довіри у вебзастосунках на основі графового оцінювання ризику та незмінного журналювання рішень. Зв'язок. - 2026. - № 2 (180). - С.72-78. DOI: 10.31673/2412-9070.2026.024909
90. Фролов Д. І. Застосування технологій машинного навчання в кібербезпеці: огляд інновацій. Д. І. Фролов, М. С. Дягілева. Радіоелектроніка та молодь у ХХІ столітті : матеріали 29-го Міжнар. молодіж. форуму, 16-19 квітня 2025 р. - Харків : ХНУРЕ, 2025. - Т. 4. - С. 97-99.
91. Шахматов І. О., Замрій І. В. Інтегрована система безпеки для захисту синхронізації платежів від MITM-атак. Problems in Programming. - 2025. - № 2. - С. 28- 39. DOI: 10.15407/pp2025.02.028
92. Шахматов І., Замрій І. Автоматизація оцінки безпеки вебзастосунків засобами Python. Зв'язок. - 2025. - № 4. - С. 58-66. DOI: 10.31673/2412- 9070.2025.045866
93. Шахматов І.О., Замрій І.В. Адаптивні нейромережі у боротьбі з вебспамом. ITSec: Безпека інформаційних технологій: матеріали XIV Міжнар. наук.-техн. конф., м. Тернопіль, 22-24 трав. 2025 р. Тернопіль-Київ: ЗУНУ-ДУІКТ, 2025. С. 211-213.
94. Замрій І., Шахматов І. Мультирепрезентаційна GNN-модель з узгодженням і адаптивним злиттям для детекції спаму. SMICS-2025 “Безпека сучасних інформаційно-комунікаційних систем” : тези доповідей (Львів, Україна, 16- 18 жовт. 2025 р.). - Львів, 2025.
95. Weisen Pan, Jian Li, Lisa Gao, Liexiang Yue, Yan Yang, Lingli Deng, Chao Deng, Semantic Graph Neural Network: A Conversion from Spam Email Classification to

Graph Classification, Scientific Programming, vol. 2022, Article ID 6737080, 8 pages, 2022. DOI:10.1155/2022/6737080

96. Chensu Zhao, Yang Xin, Xuefeng Li, Hongliang Zhu, Yixian Yang, Yuling Chen, An Attention-Based Graph Neural Network for Spam Bot Detection in Social Networks, *Applied Sciences*, 10(22) (2020) 8160. DOI:10.3390/app10228160

97. Linjie Shen, Yanbin Wang, Zhao Li, Wenrui Ma, SMS spam detection using BERT and multi-graph convolutional networks, *International Journal of Intelligent Networks*, 6 (2025) 79-88. DOI:10.1016/j.ijin.2025.06.002

98. Jiangnan Tang, Youquan Wang, Jie Cao, Haicheng Tao, Guixiang Zhu, Inter- and Intra-Graph Attention Aggregation Learning for Multi-relational GNN Spam Detection, *Procedia Computer Science*, 214 (2022) 1522-1530. DOI:10.1016/j.procs.2022.11.339

99. Cheng Wu, Chaokun Wang, Jingcao Xu, Ziyang Liu, Kai Zheng, Xiaowei Wang, Yang Song, Kun Gai, Graph Contrastive Learning with Generative Adversarial Network, in: *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23)*, ACM, 2023, pp. 2721-2730. DOI:10.1145/3580305.3599370

100. Xuxu Zheng, Chen Feng, Zhiyi Yin, Jinli Zhang, Huawei Shen, Research on Fraud Detection Method Based on Heterogeneous Graph Representation Learning, *Electronics*, 12(14) (2023) 3070. DOI:10.3390/electronics12143070

101. Jinbo Chao, Chunhui Zhao, Fuzhi Zhang, Network Embedding-Based Approach for Detecting Collusive Spamming Groups on E-Commerce Platforms, *Security and Communication Networks*, vol. 2022, Article ID 4354086, 11 pages. DOI:10.1155/2022/4354086

102. Luzhi Wang, Yizhen Zheng, Di Jin, Fuyi Li, Yongliang Qiao, Shirui Pan, Contrastive Graph Similarity Networks, *ACM Transactions on the Web*, 18(2) (2024) Article No. 17, pp. 1-20. DOI:10.1145/3580511

103. Yupeng Hou, Binbin Hu, Wayne Xin Zhao, Zhiqiang Zhang, Jun Zhou, Ji-Rong Wen, Neural Graph Matching for Pre-training Graph Neural Networks, in: *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*, SIAM, 2022, pp. 738- 747. DOI:10.1137/1.9781611977172.20

104. Ijeoma A. Chikwendu, Xiaoling Zhang, Chiagoziem C. Ukwuoma, Okechukwu C. Chikwendu, Yeong Hyeon Gu, Mugahed A. Al-antari, Spectrum-Constrained and Skip-Enhanced Graph Fraud Detection: Addressing Heterophily in Fraud Detection with Spectral and Spatial Modeling, *Symmetry*, 17(4) (2025) 476. DOI:10.3390/sym17040476
105. Saif Safaa Shakir, Leyli Mohammad Khanli, Hojjat Emami, Convolutional Graph Network-Based Feature Extraction to Detect Phishing Attacks, *Future Internet*, 17(8) (2025) 331. DOI:10.3390/fi17080331
106. Zhiwei Liu, Yingdong Dou, Philip S. Yu, Yutong Deng, Hao Peng, Alleviating the Inconsistency Problem of Applying Graph Neural Network to Fraud Detection, in: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*, ACM, 2020, pp. 1569-1572. DOI:10.1145/3397271.3401253
107. María Novo-Lourés, David Ruano-Ordás, Reyes Pavón, Rosalía Laza, Silvana Gómez-Meire, José R. Méndez, Enhancing representation in the context of multiple-channel spam filtering, *Information Processing & Management*, 59(2) (2022) 102812. DOI:10.1016/j.ipm.2021.102812
108. Sawsan Alshathtawi, Amani Shatnawi, Anas M.R. AlSobeh, Aws A. Magableh, Beyond Word-Based Model Embeddings: Contextualized Representations for Enhanced Social Media Spam Detection, *Applied Sciences*, 14(6) (2024) 2254. DOI:10.3390/app14062254
109. Venkateswarlu B, Viswanath Shenoi V, Optimized generative adversarial network with fractional calculus based feature fusion using Twitter stream for spam detection, *Information Security Journal: A Global Perspective*, (2021). DOI:10.1080/19393555.2021.1956024
110. Paul Ntim Yeboah, A. S. M. Kayes, Wenny Rahayu, Eric Pardede, Syed Mahbub, A Framework for Phishing and Web Attack Detection Using Ensemble Features of Self-supervised Pre-trained Models, *TechRxiv*, (2025). DOI:10.36227/techrxiv.173603362.21995515/v1

111. Zheng Qu, Qingyao Jia, Chen Lyu, Jia Liu, Xiaoying Liu, Kechen Zheng, Detecting Fake Reviews with Generative Adversarial Networks for Mobile Social Networks, Security and Communication Networks, (2022) 1164125. DOI:10.1155/2022/1164125
112. Yonghong Huang, Joanna Negrete, John Wagener, Celeste Fralick, Armando Rodriguez, Eric Peterson, Adam Wosotowsky, Graph neural networks and cross-protocol analysis for detecting malicious IP addresses, Complex & Intelligent Systems, 9 (2023) 3857-3869. DOI:10.1007/s40747-022-00882-2
113. Ruchi Agarwal, Anshita Dhoot, Surya Kant, Vimal Singh Bisht, Hasmat Malik, Md. Fahim Ansari, A Novel Approach for Spam Detection Using Natural Language Processing With AMALS Models, IEEE Access, 12 (2024) 124298-124313. DOI:10.1109/ACCESS.2024.3391023
114. Abdulla Al-Subaiey, Mohammed Al-Thani, Naser Abdullah Alam, Kaniz Fatema Antora, Amith Khandakar, S. M. Ashfaq Uz Zaman, Novel interpretable and robust web-based AI platform for phishing email detection, Computers and Electrical Engineering, 120 Part A (2024) 109625. DOI:10.1016/j.compeleceng.2024.109625
115. Moon I. T., Shamsuzzaman M., Mridha M. M. R., Rahaman A. S. M. Towards the advancement of cashless transaction: A security analysis of electronic payment systems. Journal of Computer and Communications. - 2022. - T. 10, № 7. DOI: 10.4236/jcc.2022.107007
116. Mishra S. Exploring the impact of AI-based cyber security financial sector management. Applied Sciences. - 2023. - T. 13, № 10. - Article 5875. DOI: 10.3390/app13105875
117. Rabbani H., Shahid M. F., Khanzada T. J. S., Siddiqui S., Jamjoom M. M., Ashari R. B., Ullah Z., Mukati M. U., Nooruddin M. Enhancing security in financial transactions: A novel blockchain-based federated learning framework for detecting counterfeit data in fintech. PeerJ Computer Science. - 2024. - T. 10. - Article e2280. DOI: 10.7717/peerj-cs.2280
118. Kawano T., Okada Y. Experimental validation of the attack-detection capability of encrypted control systems using man-in-the-middle attacks. IEEE Transactions

on Industrial Informatics. - 2023. - T. 19, № 1. - C. 123-132.
DOI: 10.1109/TII.2023.1234567

119. Obonna U. O., Opara F. K., Mbaocha C. C., Obichere J.-K. C., Akwukwaegbu I. O., Amaefule M. M., Nwakanma C. I. Detection of man-in-the-middle (MitM) cyber-attacks in oil and gas process control networks using machine learning algorithms. Future Internet. - 2023. - T. 15, № 8. - Article 280. DOI: 10.3390/fi15080280

120. Ahuja N., Singal G., Mukhopadhyay D. Ascertain the efficient machine learning approach to detect different ARP attacks. Computers & Electrical Engineering. - 2022. - T. 99. - Article 107757. DOI: 10.1016/j.compeleceng.2022.107757

121. Kampourakis V., Kambourakis G., Chatzoglou E., Zaroliagis C. Revisiting man-in-the-middle attacks against HTTPS. Network Security. - 2022. - T. 2022, № 3. - C. 8-16. DOI: 10.12968/S1353-4858(22)70028-1.

122. Muzammil M. B., Bilal M., Ajmal S., Shongwe S. C., Ghadi Y. Y. Unveiling vulnerabilities of web attacks considering man-in-the-middle attack and session hijacking. IEEE Access. - 2024. - T. 12. - C. 6365-6375. DOI: 10.1109/ACCESS.2024.3350444.

123. Alenezi M. A., Alabdulkreem E. A. Encryption algorithms modeling in detecting man-in-the-middle attacks. International Journal of Advanced Computer Science and Applications. - 2020. - T. 11, № 5. - C. 1-7.

124. Al-Abadi A. A. J., Mohamed M. B., Fakhfakh A. Enhanced random forest classifier with K-means clustering (ERF-KMC) for detecting and preventing distributed-denial-of-service and man-in-the-middle attacks in Internet-of-Medical-Things networks. Computers. - 2023. - T. 12, № 12. - Article 262. DOI: 10.3390/computers12120262.

125. Agrawal S. Harnessing quantum cryptography and artificial intelligence for next-gen payment security: A comprehensive analysis of threats and countermeasures in distributed ledger environments. International Journal of Science and Research. - 2024. - T. 13, № 3. - C. 682-687. DOI: 10.21275/SR24309103650.

126. Saranya A., Naresh R. Dual authentication for payment request verification over cloud using bilinear dual authentication payments transaction protocol. International Journal of Advanced Computer Science and Applications. - 2022. - T. 13, № 7. - C. 25-30. DOI: 10.14569/IJACSA.2022.0130737.

127. Luo B., Zhang Z., Wang Q., Ke A., Lu S., He B. AI-powered fraud detection in decentralized finance: A project life cycle perspective. *ACM Computing Surveys*. - 2024. - T. 57, № 4. - Article 96. DOI: 10.1145/3705296.
128. Omer N., Samak A. H., Taloba A. I., Abd El-Aziz R. M. A novel optimized probabilistic neural network approach for intrusion detection and categorization. *Alexandria Engineering Journal*. - 2023. - T. 72. - C. 351-361. DOI: 10.1016/j.aej.2023.03.093.
129. Ren Y., Tian H., Song W., Yang Y. Improving transaction safety via anti-fraud protection based on blockchain. *Connection Science*. - 2023. - T. 35, № 1. - Article 2163983. DOI: 10.1080/09540091.2022.2163983.
130. Ashfaq T., Khalid R., Yahaya A. S., Aslam S., Azar A. T., Alsafari S., Hameed I. A. A machine learning and blockchain based efficient fraud detection mechanism. *Sensors*. - 2022. - T. 22, № 19. - Article 7162. DOI: 10.3390/s22197162.
131. Yisroel Mirsky. Kitsune Network Attack Dataset. Nine labeled attacks with extracted features and the original network capture. Kaggle.
132. Tariq M. T. paysim dataset. M. T. Tariq. Kaggle.
133. Elliptic. Elliptic Data Set. Bitcoin Transaction Graph. Kaggle.
134. Савчук В. В. Метод захисту вебзстосунків від завантаження і виконання підозрілих файлів. Кваліфікаційна робота магістра : 125 Кібербезпека та захист інформації. В. В. Савчук ; Хмельниц. нац. ун-т. - Хмельницький, 2025. - 111 с.
135. Zamrii I., Shakhmatov I. Multi-View Graph Model with Representation Alignment and Adaptive Fusion for Better Spam Detection. *Proceedings of the Workshop on Cryptology and Data Security (WCDS 2025), co-located with SMICS 2025, Lviv, Ukraine, October 16-18, 2025. CEUR Workshop Proceedings. 2026. Vol. 4191. P. 99-106*
136. Zhebka V., Zhebka S., Bazhan T., Skladannyi P., Sokolov V. Methodology for Choosing a Consensus Algorithm for Blockchain Technology. *Digital Economy Concepts and Technologies (DECaT'2024). CEUR Workshop Proceedings. 2024. Vol. 3665. P. 106-113*
137. Popereshnyak S., Veчерkovskaya A., Zhebka V. Intrusion Detection based on an Intelligent Security System using Machine Learning Methods. *Cybersecurity Providing*

in Information and Telecommunication Systems (CPITS-2024). CEUR Workshop Proceedings. 2024. Vol. 3654. P. 163-178

138. Zhebka S., Zhebka V., Hulak H., Kyrychok R., Platonenko A. Method for Adaptive Allocation of Cryptographic Resources in Distributed Databases. Cybersecurity Providing in Information and Telecommunication Systems 2025 (CPITS 2025). CEUR Workshop Proceedings. 2025. Vol. 3991. P. 620-628

139. Hanhalo I., Chytulian V., Zhebka V., Bebeshko B., Khorolska K. Adaptive Approach to Ensuring the Functional Stability of Corporate Educational Platforms under Dynamic Cyber Threats. Cybersecurity Providing in Information and Telecommunication Systems 2025 (CPITS 2025). CEUR Workshop Proceedings. 2025. Vol. 3991. P. 481-491

140. Anakhov P., Zhebka V., Popereshnyak S., Skladannyi P., Sokolov V. Protecting Objects of Critical Information Infrastructure from Wartime Cyber Attacks by Decentralizing the Telecommunications Network. Cybersecurity Providing in Information and Telecommunication Systems II (CPITS-II 2023). CEUR Workshop Proceedings. 2023. Vol. 3550. P. 240-245

141. Бученко І. А., Лемешко А. В., Лащевська Н. О. Адаптивний підхід до оцінювання ризиків кібербезпеки в розподілених інформаційних системах на основі нейронних мереж. Зв'язок. 2026. № 1. С. 65-71. DOI: 10.31673/2412-9070.2026.017404

142. Твердохліб А. О., Лащевська Н. О. Моделі застосування блокчейн-технологій у високонавантажених комп'ютерних системах для розподіленого зберігання та обробки даних. Телекомунікаційні та інформаційні технології. 2025. № 2. С. 88-99. DOI: 10.31673/2412-4338.2025.025697

143. Сосновий В. О., Лащевська Н. О. Виявлення шкідливої діяльності з використанням нейронної мережі для безперервної роботи. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка». 2024. Т. 3, № 23. С. 213-224. DOI: 10.28925/2663-4023.2024.23.213224

144. Сосновий В. О., Лащевська Н. О. Розробка структури нейронної мережі для аналізу виявлення вторгнень. Телекомунікаційні та інформаційні технології. 2023. № 4 (81). С. 101-109. DOI: 10.31673/2412-4338.2023.040109

145. Жебка С. В., Власенко В. О., Аронов А. О., Колодюк А. В. Addressing the Selection Dilemma of Consensus Algorithm in Distributed Systems. Телекомунікаційні та інформаційні технології. 2024. № 1. DOI: 10.31673/2412-4338.2024.019904
146. Вишнівський В. В., Замарусєва І. В., Аронов А. О., Кравчук П. О. Інформаційні технології аналітичного опрацювання різномовних текстових джерел як фактор безпеки прийняття управлінських рішень. Наукові записки ДУТ. 2023. № 2(4). DOI: 10.31673/2518-7678.2023.020101
147. Imperva. 2025 Imperva Bad Bot Report: The Rapid Rise of Bots and the Unseen Risk for Business. Imperva, a Thales company, 2025. 46 p.
148. DataDome. Global Bot Security Report 2025: The state of bot attack & AI traffic readiness across 17,000 websites worldwide. DataDome, 2025. 55 p.
149. Замрій І.В., Нестеренко К.С., Задонцев Ю.В., Глушкова О.І., Шахматов І.О. Методика підвищення функціональної стійкості інформаційної системи через виявлення вторгнень та реконфігурації мережі. Матеріали XIV Міжнародної науково-практичної конференції «Математика. Інформаційні технології. Освіта», Луцьк - Світязь, 13-15 червня 2025 р. С. 101-104.
150. Бовкун І.В., Шахматов І.О. Визначення вимог до вебсистеми для управління безконтактними замовленнями у ресторані. II Всеукраїнська науково-технічна конференція «Виклики та рішення в програмній інженерії». Збірник тез. Київ: ДУІКТ, 2025. С. 432-434.
151. Wu G, Zhou J and Fu X. Improved blockchain-based ECDSA batch verification scheme. Front. Blockchain 8:1495984. 2025. DOI: 10.3389/fbloc.2025.1495984

Додаток А

Псевдокод програмної реалізації інтегрованого контуру довіри й цілісності у вебсистемах

А.1. Псевдокод методу контролю доступу та блокчейн-фіксації рішення

```

1.  def AccessAlgorithmBlockchain(PU, S, F, threshold, blockchain):
2.      access_score ← AccessAlgorithm(PU, S, F)
3.      if access_score ≥ threshold:
4.          decision ← "ALLOW"
5.      else:
6.          decision ← "DENY"
7.      event ← {
8.          "user": PU.id,
9.          "service": S.id,
10.         "feature": F.id,
11.         "access_score": access_score,
12.         "threshold": threshold,
13.         "decision": decision,
14.         "timestamp": CurrentTime()
15.     }
16.     prev_block ← GetLastBlock(blockchain)
17.     new_block ← CreateBlock(event, prev_block)
18.     blockchain ← AddBlock(blockchain, new_block)
19.     return decision, new_block.hash
20. def AccessAlgorithm(PU, S, F):
21.     service_features ← internal_services_features(S)
22.     user_level ← PU.access_level
23.     feature_level ← F.level
24.     CFLAG ← flag_for_level(service_features, user_level, feature_level)
25.     return CFLAG
26. def internal_services_features(S):
27.     features ← S.features
28.     return features
29. def flag_for_level(features, user_level, feature_level):
30.     score ← 0
31.     for feature in features:
32.         if user_level ≥ feature.level:
33.             score ← score + 1
34.     CFLAG ← score / len(features)
35.     if user_level < feature_level:
36.         CFLAG ← CFLAG * 0.5
37.     return CFLAG

```

A.2. Псевдокод формування блоку та перевірки блокчейн-журналу

```

1.  def CreateBlock(data, previous_block):
2.      block.index  $\leftarrow$  previous_block.index + 1
3.      block.previous_hash  $\leftarrow$  previous_block.hash
4.      block.timestamp  $\leftarrow$  CurrentTime()
5.      block.data  $\leftarrow$  data
6.      block.hash  $\leftarrow$  CalculateHash(block)
7.      return block
8.
9.  def CalculateHash(block):
10.     input_string  $\leftarrow$  Concatenate(
11.         block.index,
12.         block.previous_hash,
13.         block.timestamp,
14.         block.data
15.     )
16.     hash_value  $\leftarrow$  Hash(input_string)
17.     return hash_value
18.
19. def AddBlock(blockchain, block):
20.     blockchain.append(block)
21.     return blockchain
22.
23. def VerifyBlockchain(blockchain):
24.     for i  $\leftarrow$  1 to len(blockchain) - 1:
25.         current_block  $\leftarrow$  blockchain[i]
26.         previous_block  $\leftarrow$  blockchain[i - 1]
27.         if current_block.previous_hash  $\neq$  previous_block.hash:
28.             return False
29.         recalculated_hash  $\leftarrow$  CalculateHash(current_block)
30.         if recalculated_hash  $\neq$  current_block.hash:
31.             return False
32.     return True

```

A.3. Псевдокод незмінного журналювання критичних подій

```

1.  def LogCriticalEvent(user, web_resource, operation, operation_data, blockchain):
2.      transaction ← CreateTransaction(
3.          user,
4.          web_resource,
5.          operation,
6.          operation_data
7.      )
8.      blockchain.pending_transactions.append(transaction)
9.      if NeedCreateBlock(blockchain.pending_transactions, transaction):
10.         prev_block ← GetLastBlock(blockchain)
11.         block ← CreateTransactionBlock(
12.             blockchain.pending_transactions,
13.             prev_block
14.         )
15.         blockchain.chain.append(block)
16.         blockchain.pending_transactions ← []
17.     return transaction.id
18. def CreateTransaction(user, web_resource, operation, operation_data):
19.     transaction.id ← GenerateId()
20.     transaction.user_id ← user.user_id
21.     transaction.resource_url ← web_resource.url
22.     transaction.operation ← operation
23.     transaction.data ← operation_data
24.     transaction.timestamp ← CurrentTime()
25.     return transaction
26. def NeedCreateBlock(transactions, transaction):
27.     if len(transactions) ≥ BLOCK_SIZE:
28.         return True
29.     if transaction.operation in CRITICAL_OPERATIONS:
30.         return True
31.     return False
32. def CreateTransactionBlock(transactions, previous_block):
33.     block.timestamp ← CurrentTime()
34.     block.transactions ← transactions
35.     block.previous_hash ← previous_block.hash
36.     block.hash ← Hash(block.timestamp, block.transactions, block.previous_hash)
37.     return block

```

A.4. Псевдокод аналітичної обробки блокчейн-журналу

```

1.  def DataAnalytics(blockchain):
2.      report ← InitReport()
3.      for block in blockchain.chain:
4.          block_valid ← VerifyBlock(block)
5.          for transaction in block.transactions:
6.              user_id ← transaction.user_id
7.              resource_url ← transaction.resource_url
8.              operation ← transaction.operation
9.              timestamp ← transaction.timestamp
10.         report.total_transactions ← report.total_transactions + 1
11.         report.by_user[user_id] ← report.by_user[user_id] + 1
12.         report.by_resource[resource_url] ← report.by_resource[resource_url] + 1
13.         report.by_operation[operation] ← report.by_operation[operation] + 1
14.         if block_valid = False:
15.             report.invalid_blocks ← report.invalid_blocks + 1
16.     report.chain_valid ← VerifyBlockchain(blockchain.chain)
17.     return report
18.
19. def VerifyBlock(block):
20.     recalculated_hash ← Hash(
21.         block.timestamp,
22.         block.transactions,
23.         block.previous_hash
24.     )
25.     if recalculated_hash = block.hash:
26.         return True
27.     return False

```

A.5. Псевдокод модуля підготовки та аналізу даних

```

1.  def DataProcessingPipeline(file_path, target_column, test_size, random_state):
2.      data ← DataLoader.load_data(file_path)
3.      df ← DataPreprocessor.convert_to_dataframe(data)
4.      df ← DataPreprocessor.parse_datetime(df, "datetime")
5.      df ← DataPreprocessor.log_transform(df, "amount", 1)
6.      df ← DataPreprocessor.apply_country_risk(df, "country", risk_weights)
7.      df ← DataPreprocessor.calculate_hour_risk(df, "hour",  $\mu$ ,  $\sigma$ )
8.      df ← DataPreprocessor.drop_columns(df, drop_columns)
9.      df ← FeatureEncoder.fit_transform(df, categorical_columns)
10.  X_train, X_test, y_train, y_test ← DataSplitter.split_data(
11.      df,
12.      target_column,
13.      test_size,
14.      random_state
15.  )
16.  X_train ← DataScaler.fit_transform(X_train)
17.  X_test ← DataScaler.transform(X_test)
18.  model ← ModelTrainer.train(X_train, y_train)
19.  y_pred ← ModelTrainer.predict(model, X_test)
20.  report ← ModelTrainer.evaluate(y_test, y_pred)
21.  return model, report
22. A.6. Псевдокод розрахунку ризику транзакції
23. def RiskCalculator(df):
24.     for row in df:
25.         amount_risk ←  $a * \log(\text{row.amount} + 1)$ 
26.         time_risk ←  $b * \exp($ 
27.              $-((\text{row.hour} - \mu)^2) / (2 * \sigma^2)$ 
28.         )
29.         country_risk ← country_weights[row.country]
30.         card_risk ← card_weights[row.card_type]
31.         total_risk ← amount_risk + time_risk + country_risk + card_risk
32.         row.risk_score ← Normalize(total_risk)
33.     return df
34. def Normalize(value):
35.     if value < 0:
36.         return 0
37.     if value > 1:
38.         return 1
39.     return value

```

A.7. Псевдокод навчання та використання моделі аналізу транзакцій

```

1. def ModelTrainer(model, X_train, y_train, X_test, y_test):
2.   model.fit(X_train, y_train)
3.   y_pred ← model.predict(X_test)
4.   accuracy ← CalculateAccuracy(y_test, y_pred)
5.   precision ← CalculatePrecision(y_test, y_pred)
6.   recall ← CalculateRecall(y_test, y_pred)
7.   f1 ← CalculateF1(y_test, y_pred)
8.   report ← {
9.     "accuracy": accuracy,
10.    "precision": precision,
11.    "recall": recall,
12.    "f1": f1
13.  }
14.  return model, report
15.
16. def TransactionAnalyzer(transaction_data, model, scaler, encoder):
17.  transaction ← prepare_new_transaction(transaction_data)
18.  transaction ← encoder.transform(transaction)
19.  transaction ← scaler.transform(transaction)
20.  probability ← model.predict_proba(transaction)
21.  if probability <  $\tau_1$ :
22.    decision ← "SAFE"
23.  else if probability <  $\tau_2$ :
24.    decision ← "SUSPECT"
25.  else:
26.    decision ← "THREAT"
27.  return decision, probability

```

A.8. Псевдокод побудови графа подій вебсистеми

```

1.  def BuildGraph(events, features_by_id):
2.      G.nodes ← []
3.      G.edges ← []
4.      for event in events:
5.          node.id ← event.id
6.          node.features ← features_by_id[event.id]
7.          G.nodes.append(node)
8.      for i ← 0 to len(events) - 1:
9.          for j ← i + 1 to len(events) - 1:
10.             e_i ← events[i]
11.             e_j ← events[j]
12.             if IsLinked(e_i, e_j):
13.                 edge_ij.source ← e_i.id
14.                 edge_ij.target ← e_j.id
15.                 edge_ij.weight ← CalculateWeight(e_i, e_j)
16.                 edge_ji.source ← e_j.id
17.                 edge_ji.target ← e_i.id
18.                 edge_ji.weight ← edge_ij.weight
19.                 G.edges.append(edge_ij)
20.                 G.edges.append(edge_ji)
21.      return G
22.
23. def IsLinked(e_i, e_j):
24.     same_actor ← e_i.actor = e_j.actor
25.     same_ip ← e_i.ip_prefix = e_j.ip_prefix
26.     same_device ← e_i.device_id = e_j.device_id
27.     close_time ← Abs(e_i.timestamp - e_j.timestamp) ≤ Δt
28.     if close_time and (same_actor or same_ip or same_device):
29.         return True
30.     return False
31.
32. def CalculateWeight(e_i, e_j):
33.     similarity ← FeatureSimilarity(e_i.features, e_j.features)
34.     time_factor ← 1 / (1 + Abs(e_i.timestamp - e_j.timestamp))
35.     weight ← similarity + time_factor
36.     return weight

```

A.9. Псевдокод графово-нейромережевої моделі SpamFilterGNN

```

1. def SpamFilterGNN( $X$ ,  $edge\_index$ ,  $\theta$ ):
2.    $H\_1 \leftarrow GraphConvolution(X, edge\_index, \theta\_1)$ 
3.    $H\_1 \leftarrow ReLU(H\_1)$ 
4.    $H\_2 \leftarrow GraphConvolution(H\_1, edge\_index, \theta\_2)$ 
5.    $Y \leftarrow Softmax(H\_2)$ 
6.   return  $Y$ 
7. def GraphConvolution( $X$ ,  $edge\_index$ ,  $\theta$ ):
8.    $A \leftarrow BuildAdjacencyMatrix(edge\_index)$ 
9.    $A\_hat \leftarrow A + I$ 
10.   $D\_hat \leftarrow DegreeMatrix(A\_hat)$ 
11.   $Z \leftarrow D\_hat^{(-1/2)} * A\_hat * D\_hat^{(-1/2)} * X * \theta$ 
12.  return  $Z$ 

```


A.10. Псевдокод навчання графово-нейромережевої моделі

```

1. def TrainSpamFilterGNN(model, graph_data, labels, optimizer, epochs):
2.   for epoch  $\leftarrow 1$  to epochs:
3.     model.train()
4.     output  $\leftarrow$  model.forward(
5.       graph_data.features,
6.       graph_data.edge_index
7.     )
8.     loss  $\leftarrow$  Loss(output, labels)
9.     optimizer.zero_grad()
10.    loss.backward()
11.    optimizer.step()
12.  return model
13. def EvaluateSpamFilterGNN(model, test_graph, test_labels):
14.  model.eval()
15.  output  $\leftarrow$  model.forward(
16.    test_graph.features,
17.    test_graph.edge_index
18.  )
19.  loss  $\leftarrow$  Loss(output, test_labels)
20.  y_pred  $\leftarrow$  ArgMax(output)
21.  accuracy  $\leftarrow$  CalculateAccuracy(test_labels, y_pred)
22.  f1  $\leftarrow$  CalculateF1(test_labels, y_pred)
23.  return loss, accuracy, f1

```

A.11. Псевдокод класифікації повідомлення вебформи

```

1. def ClassifyMessage(message_event, graph_context, trained_gnn):
2.   features  $\leftarrow$  ExtractMessageFeatures(message_event)
3.   graph_context  $\leftarrow$  AddNode(graph_context, message_event.id, features)
4.   graph_context  $\leftarrow$  AddContextEdges(graph_context, message_event)
5.   output  $\leftarrow$  trained_gnn.forward(
6.     graph_context.features,
7.     graph_context.edge_index
8.   )
9.   probability  $\leftarrow$  output[message_event.id]
10.  if probability  $<$   $\tau_1$ :
11.    class_label  $\leftarrow$  "LEGITIMATE"
12.  else if probability  $<$   $\tau_2$ :
13.    class_label  $\leftarrow$  "SUSPICIOUS"
14.  else:
15.    class_label  $\leftarrow$  "SPAM"
16.  return class_label, probability

```

A.12. Псевдокод зведеного конвеєра ІКДЦ

```

1.  def SecurityPipeline(e):
2.      canonical_event ← Canonicalizer.canonical_event(e)
3.      event_hash ← Hash(canonical_event)
4.      event_signature ← Sign(event_hash)
5.      if e.type = "SUBMIT":
6.          raw_features ← ExtractSubmitFeatures(e)
7.      else if e.type = "TX":
8.          raw_features ← ExtractTransactionFeatures(e)
9.      features ← FeatureNormalizer.transform(raw_features)
10.     feature_hash ← Hash(features)
11.     SlidingWindowStore.add(e)
12.     events ← SlidingWindowStore.events(e.timestamp)
13.     features_by_id ← ExtractFeaturesForWindow(events)
14.     graph ← GraphBuilder.build(events, features_by_id)
15.     probability, contribution ← RiskModel.predict(graph, e.id)
16.     decision, action ← DecisionPolicy.decide(probability)
17.     ledger_record ← CreateLedgerRecord(
18.         event_hash,
19.         event_signature,
20.         feature_hash,
21.         RiskModel.model_id,
22.         RiskModel.model_hash,
23.         DecisionPolicy.policy_id,
24.         DecisionPolicy.policy_hash,
25.         probability,
26.         decision,
27.         action,
28.         contribution,
29.         e.timestamp
30.     )
31.     LedgerClient.append(ledger_record)
32.     return decision, action, probability

```

A.13. Псевдокод аудитної перевірки рішення

```
1. def AuditVerifier(original_event, ledger_record, ledger):
2.     canonical_event ← Canonicalizer.canonical_event(original_event)
3.     recalculated_event_hash ← Hash(canonical_event)
4.     if recalculated_event_hash ≠ ledger_record.event_hash:
5.         return "EVENT_MODIFIED"
6.     signature_valid ← VerifySignature(
7.         ledger_record.event_hash,
8.         ledger_record.event_signature
9.     )
10.    if signature_valid = False:
11.        return "INVALID_SIGNATURE"
12.    chain_valid ← VerifyBlockchain(ledger)
13.    if chain_valid = False:
14.        return "LEDGER_DAMAGED"
15.    return "VALID"
```

Додаток Б**АКТ**

**впровадження результатів дисертаційного дослідження
Шахматова Івана Олександровича
на тему: «Моделі та методи забезпечення довіри й цілісності у
вебсистемах»**

В межах функціонування підприємства ТОВ «ШЛІФАРБ» було впроваджено метод виявлення вебспаму та підозрілої активності у вебсистемах на основі нейронної мережі, зокрема, до автоматичного виявлення спаму та підозрілих повідомлень у формах сайту, розроблений Шахматовим Іваном Олександровичем під час виконання дисертаційного дослідження на тему «Моделі та методи забезпечення довіри й цілісності у вебсистемах».. За результатами впровадження методу встановлено, що узагальнений показник якості автоматичного розпізнавання повідомлень зріс з 78% у базовому варіанті до 90% після впровадження графової моделі, а після її подальшого навчання — до 92%. Частка помилкових спрацьовувань, коли система помилково позначала нормальні звернення як підозрілі, зменшилася з 2.6% до 1.2%, а згодом — до 0.9%. Таким чином, у порівнянні з початковим варіантом кількість помилкових спрацьовувань зменшилася на 65.4%. Це дозволило зменшити кількість спам-повідомлень, що потрапляють до обробки, скоротити навантаження на адміністрування заявок та покращити опрацювання реальних звернень від клієнтів. Окремо встановлено, що після накопичення нових прикладів і донавчання моделі якість її роботи додатково покращилася, що підтверджує доцільність використання такого підходу в реальних умовах експлуатації сайту.

Результати впровадження підтверджують практичну придатність розроблених у дисертаційній роботі рішень для використання у вебсистемах підприємства та можуть бути рекомендовані для подальшого застосування у задачах автоматичної фільтрації спаму, підтримки обробки звернень користувачів і підвищення надійності роботи форм сайту.

Директор ТОВ «ШЛІФАРБ»





ЗАТВЕРДЖУЮ

Директор ТОВ «АРМА МОТОРС КИЇВ»

Д.С. Погуляєв

(підпис, ініціали, прізвище)

«27» листопада 2025 року

АКТ

**про реалізацію результатів наукових досліджень
Шахматова Івана Олександровича
на тему:**

«Моделі та методи забезпечення довіри й цілісності у вебсистемах»

Комісія у складі: голови комісії – Менеджера (управителя) інформаційних технологій Д.С. Іващенко та членів комісії: Інженера програміста С.Ю. Бродського та Інженер програміста С.В. Сопіна, що в ТОВ «АРМА МОТОРС КИЇВ» були використані наукові результати наукових досліджень Шахматова Івана Олександровича:

- 1) метод забезпечення довіри й цілісності критичних подій і даних на основі блокчейн-орієнтованого контролю цілісності критичних подій і журналювання транзакційних операцій, що забезпечує безпечну API-взаємодію між сервісами платіжної інфраструктури, підвищує прозорість обробки платіжних подій і зменшує ризик прихованої зміни транзакційних даних;
- 2) програмні засоби, що реалізують блокчейн-орієнтовані методи контролю цілісності критичних подій і журналювання транзакційних операцій у платіжних терміналах компанії, у межах використання рішення для транзакційних подій встановлено, що узагальнений показник якості виявлення ризикових ситуацій зріс з 74% у базовій конфігурації до 88% у графовій конфігурації та до 90% після переходу до повного інтегрованого контуру. Частка помилкових спрацювань при цьому зменшилася з 1.9% до 1.0% і далі до 0.8%. Додатково встановлено, що врахування інтегральної оцінки транзакційного ризику підвищує якість виявлення ризикових транзакцій з 85% до 88% при одночасному зниженні частки помилкових спрацювань з 1.2% до 1.0%.

У процесі впровадження та контрольного оцінювання на матеріалах платіжної інфраструктури компанії встановлено, що повнота виявлення атак типу MITM на канали обміну транзакційними подіями склала 98%. Це підтвердило доцільність застосування блокчейн-журналювання, криптографічного зв'язування записів, часових міток і додаткової перевірки подій у критичних точках API-взаємодії між сервісами, що дозволяє

своєчасно виявляти спроби підміни, повторного надсилання або несанкціонованої зміни транзакційних повідомлень у процесі їх передавання та обробки. У процесі контрольного оцінювання швидкодії встановлено, що після впровадження повного інтегрованого контуру час обробки 95% контрольованих подій збільшився з 52 мс до 64 мс. Таке збільшення є прийнятним для платіжної інфраструктури, оскільки компенсується підвищенням контролю цілісності транзакцій, покращенням простежуваності подій та розширенням можливостей аудиту операцій між сервісами.

Даний акт не є підставою для фінансових взаєморозрахунків.

Голова комісії:

Менеджер (управитель)
інформаційних технологій

Д.С. Іващенко

Члени комісії:

Інженер програміст

С.Ю. Бродський

Інженер програміст

С.В. Сопін

ЗАТВЕРДЖУЮ

Заступник директора з наукової роботи

Інституту програмних систем

Національної академії наук України



Віктор ШЕВЧЕНКО

року

АКТ

**впровадження результатів дисертаційного дослідження
Шахматова Івана Олександровича на тему:
«Моделі та методи забезпечення довіри й цілісності у вебсистемах»**

Комісія у складі:

Голова: завідувач відділу організації та супроводження наукових досліджень Чистяков В.А., к.т.н.

члени комісії:

завідувач відділу Федоренко Р.М., к.е.н., ст.досл.

заступник завідувача відділу Ігнатенко П.П., к.т.н., с.н.с.

розглянула матеріали дисертаційного дослідження Шахматова Івана Олександровича та встановила, що у межах дисертаційної роботи впроваджено результати, що стосуються моделі інтегрованого контуру довіри й цілісності (ІКДЦ) у вебсистемі, методу блокчейн-верифікованого журналювання критичних подій і контролю доступу у вебсистемах, методу графо-нейромережевого виявлення вебспау та підозрілої активності у вебсистемах та методу інтегрованого забезпечення довіри й цілісності у вебсистемах.

Розроблена модель інтегрованого контуру довіри й цілісності (ІКДЦ) у вебсистемі, яка за рахунок поєднання незмінного журналювання критичних подій, формалізованого подання зв'язків між вебформами, SQL-операціями, рішеннями аналітичного модуля та політиками реагування забезпечує єдине інформаційне середовище для контролю цілісності даних, відстежуваності подій, придатності до аудиту та відтворюваності рішень у вебсистемі, забезпечує узгоджене представлення критичних подій у межах єдиного контуру обробки, контроль змін даних і можливість ретроспективного аналізу дій у розподілених програмних системах.

Запропонований метод інтегрованого забезпечення довіри й цілісності у вебсистемах, що ґрунтується на розробленій моделі ІКДЦ, методі блокчейн-

верифікованого журналювання критичних подій і контролю доступу у вебсистемах та методі графо-нейромережевого виявлення вебспау та підозрілої активності у вебсистемах, забезпечує єдиний підхід до контролю цілісності подій, аналізу ризикових взаємодій між сервісами та підтримки прийняття рішень у вебсистемах. Контрольне оцінювання показало, що застосування цього методу дозволяє підвищити якість оцінювання подій з 85% до 96%, а частку помилкових спрацювань зменшити з 3% до 0,8%, що підтверджує ефективність запропонованого підходу для задач забезпечення довіри й цілісності у вебсистемах. Розробка має модульну структуру та може бути адаптована до задач аналізу критичних подій у різних типах вебсистем і сервісно-орієнтованих програмних середовищах.

Комісія встановила та цим актом засвідчує, що матеріали дисертаційного дослідження Шахматова Івана Олександровича впроваджені в Інституті програмних систем Національної академії наук України при формуванні плану перспективних наукових досліджень.

Даний акт не є підставою для фінансових зобов'язань.

Голова комісії

Члени комісії



Валерій ЧИСТЯКОВ

Руслан ФЕДОРЕНКО

Петро ІГНАТЕНКО

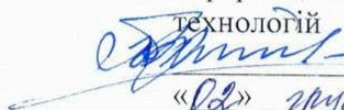
ЗАТВЕРДЖУЮ

Перший проректор

Державного університету

інформаційно-комунікаційних

технологій

 Олександр КОРЧЕНКО

«22» грудня 2025 р.

АКТ

впровадження в освітній процес Державного університету інформаційно-комунікаційних технологій наукових результатів Шахматова Івана Олександровича, одержаних під час проведення дисертаційного дослідження на здобуття наукового ступеня доктора філософії за спеціальністю 121 Інженерія програмного забезпечення, галузі знань 12 Інформаційні технології на тему: «Моделі та методи забезпечення довіри й цілісності у вебсистемах»

Комісія у складі:

голови комісії – директора Навчально-наукового інституту інформаційних технологій, д.т.н., професора Катерини НЕСТЕРЕНКО;

членів комісії:

завідувача кафедри Інженерії програмного забезпечення Навчально-наукового інституту інформаційних технологій, д.т.н., професора Ірини ЗАМРІЙ;

завідувача кафедри Інформаційних систем і технологій Навчально-наукового інституту інформаційних технологій, д.т.н., професора Каміли СТОРЧАК;

доцента кафедри інженерії програмного забезпечення Навчально-наукового інституту інформаційних технологій, к.т.н. Юрія ЗАДОНЦЕВА;

доцента кафедри інженерії програмного забезпечення Навчально-наукового інституту інформаційних технологій, доктора філософії Віталія ЗАЛИВИ;

провела роботу щодо визначення фактичного впровадження результатів наукового дослідження здобувача наукового ступеня доктора філософії Шахматова І.О. в освітній процес Державного університету інформаційно-комунікаційних технологій.

У результаті проведеної роботи комісія встановила:

1. Наукові результати, одержані здобувачем, використовуються при підготовці здобувачів освітнього рівня бакалавр за спеціальністю 121 Інженерія програмного забезпечення:

1.1. Наукові результати дисертаційної роботи, що стосуються побудови підсистеми довіри й цілісності у вебсистемах, методів незмінного журналювання критичних подій, використання хешування, цифрових підписів та механізмів аудиторної перевірки, використовуються у межах дисципліни «Безпека програм та даних» при вивченні питань забезпечення інформаційної безпеки і цілісності даних у програмних системах.

1.2. Наукові результати дисертаційної роботи, що стосуються застосування графових нейронних мереж для виявлення вебспаму та підозрілої активності, використовуються у межах дисципліни «Безпека програм та даних» як приклад сучасних засобів інтелектуального аналізу подій у задачах захисту вебзастосунків.

2. Вказані наукові результати Шахматова І.О. представлені у формі окремих навчальних тем, прикладів програмного коду та кейсів у методичних матеріалах до лекцій і практичних занять згаданих дисциплін, які проводяться в Державному університеті інформаційно-комунікаційних технологій на кафедрі Інженерії програмного забезпечення.

Голова комісії:

Директор Навчально-наукового інституту
інформаційних технологій,
д-р техн. наук, професор



Катерина НЕСТЕРЕНКО

Члени комісії:

Завідувач кафедри
Інженерії програмного забезпечення
д-р техн. наук, професор



Ірина ЗАМРІЙ

Завідувач кафедри
Інформаційних систем і технологій
д-р техн. наук, професор



Каміла СТОРЧАК

Доцент кафедри
Інженерії програмного забезпечення
канд. техн. наук



Юрій ЗАДОНЦЕВ

Доцент кафедри
Інженерії програмного забезпечення
доктор філософії



Віталій ЗАЛИВА