НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Кваліфікаційна наукова праця
на правах рукопису

ДИКА АНАСТАСІЯ ІВАНІВНА

УДК 004.891.2:004.056

ДИСЕРТАЦІЯ
МЕТОДИ МАШИННОГО НАВЧАННЯ В ТЕСТУВАННІ БЕЗПЕКИ ВЕБОРІЄНТОВАНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

122 – Комп’ютерні науки
12 – Інформаційні технології

Подається на здобуття ступеня доктора філософії.
Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ А. І. Дика

Науковий керівник:
Трофименко Олена Григорівна,
кандидат технічних наук, доцент

Одеса – 2026

АНОТАЦІЯ

Дика А. І. Методи машинного навчання в тестуванні безпеки веборієнтованого програмного забезпечення. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття ступеня доктора філософії за спеціальністю 122 – Комп'ютерні науки (галузь знань 12 – Інформаційні технології). – Національний університет «Одеська юридична академія», Одеса, 2026.

Дисертаційна робота присвячена розробці високоефективних методів тестування безпеки вебзастосунків на основі штучного інтелекту (ШІ) і машинного навчання, орієнтованих на виявлення складних та актуальних кіберзагроз, стійких до традиційних методів захисту. Дослідження сфокусоване на двох ключових напрямах: розробці наукового базису та методів детектування прихованих стеганографічних каналів у медіафайлах вебзастосунків з використанням низьковитратного математичного апарату перетворення Уолша-Адамара, а також створенні адаптивної системи протидії фішингу шляхом глибинного аналізу серверного контенту та поведінкових метрик у реальному часі. Обидва напрями узагальнені в концептуальну модель тестування вебзастосунків, яка пропонує нові підходи до інтеграції механізмів безпеки в життєвий цикл розробки веборієнтованого програмного забезпечення (ПЗ).

У першому розділі дисертаційної роботи проведено комплексний аналіз уразливостей та проблем тестування безпеки вебзастосунків. Розглянуто життєвий цикл розробки програмного забезпечення в контексті сучасних методологій (Waterfall, Agile), фреймворків (Scrum, Kanban) та практик DevOps (CI/CD) з акцентом на інтеграцію безпеки на всіх етапах. Показано важливість підходу DevSecOps, який дозволяє вбудовувати механізми захисту ще на етапі проєктування і кодування, а також забезпечує системний моніторинг у продуктивному середовищі. Запропоновано дихотомічну класифікацію безпеки вебзастосунків: безпека розробки (DevSec) та безпека

експлуатації (Runtime Security), що дозволяє чітко структурувати місце захисних заходів у життєвому циклі. Окрему увагу приділено фішинговим атакам та організації прихованих каналів передачі інформації на основі мультимедіа, які становлять найбільш актуальні та складні загрози. Показано, що наявні підходи мають обмежену ефективність і вимагають розробки нових, адаптивних методів на основі штучного інтелекту та стеганоаналізу, здатних забезпечувати надійний захист у реальному часі.

У другому розділі дисертації сформовано теоретичний базис виявлення прихованих каналів у вебзастосунках. Показано, що методи з кодовим управлінням та підходи SVD UV вирізняються високою стійкістю до класичних алгоритмів стеганоаналізу та становлять значну загрозу для безпеки інформаційних систем. Для подолання цих викликів розроблено математичний апарат, що ґрунтується на перетворенні Уолша-Адамара, яке характеризується низькою обчислювальною складністю та зберігає інформативність трансформант. Запропоновані підходи дозволяють виявляти характерні статистичні та структурні ознаки, спричинені вбудовуванням додаткової інформації, як у випадку методів з кодовим управлінням, так і для SVD UV-методів. Проведені експериментальні дослідження підтвердили можливість ефективного застосування розробленого базису як у несліпих, так і в сліпих сценаріях аналізу. Отримані результати створюють основу для розробки масштабованих стеганоаналітичних рішень, здатних працювати у вебсередовищі в режимі реального часу.

У третьому розділі дисертації розроблено три нові методи виявлення прихованих каналів у вебзастосунках на основі технологій штучного інтелекту. На відміну від традиційних статистичних підходів, запропоновані методи враховують високу варіативність даних та здатність зловмисників адаптувати стеганографічні алгоритми. Два із запропонованих методів призначено для стеганоаналізу повідомлень на основі кодового управління: із застосуванням обчислення огинаючої гістограми та ансамблю згорткових нейронних мереж. Перший метод забезпечує ефективне виділення ключових пікових ознак і

досягає точності понад 93.3%, тоді як другий демонструє більш високі результати (точність 94.3%) завдяки повній автоматизації процесу аналізу. Також запропоновано метод виявлення стеганоповідомлень, створених із використанням SVD UV-методів, з класифікацією на основі SVM, що дозволяє підвищити точність і стійкість до складних модифікацій даних. Проведене з наявними аналогами порівняння підтвердило переваги розроблених методів у швидкодії, надійності та практичній придатності для інтеграції у системи кіберзахисту вебзастосунків.

У четвертому розділі дисертації зосереджено увагу на проблематиці захисту вебзастосунків від фішингових атак, які є однією з найбільш актуальних загроз сучасного цифрового середовища. Проведено класифікацію різновидів фішингу, включно з атаками через електронну пошту, соціальні мережі, SMS, а також новітніми техніками з використанням генеративних моделей штучного інтелекту. На основі аналізу сучасних підходів, рекомендованих міжнародними організаціями (OWASP, NIST, ENISA), окреслено їхні сильні сторони та обмеження. Запропоновано новий метод автоматизованого виявлення шкідливого контенту, який поєднує байтове подання HTML-документів із візуалізацією у вигляді зображень та подальшою обробкою за допомогою алгоритмів машинного навчання. Розроблена модель на основі SVM досягла точності класифікації 92.45%, продемонструвавши збалансованість precision та recall. Метод відзначається універсальністю, незалежністю від текстового наповнення сторінок і можливістю інтеграції у браузери чи серверні системи. Це забезпечує його практичну цінність у реальних умовах роботи систем кіберзахисту вебзастосунків. У розділі проведено систематизацію сучасних підходів і засобів тестування безпеки вебзастосунків, а також визначено можливості інтеграції розроблених автором методів у практику кіберзахисту. Запропоновані ШІ-методи стеганоаналізу та виявлення фішингу органічно доповнюють традиційні практики та виконують роль активних runtime-механізмів для протидії прихованим каналам і

фішинговим атакам. Це створює єдиний контур безпеки, здатний адаптивно реагувати на сучасні виклики.

Робота розвиває підходи до підвищення ефективності тестування безпеки вебзастосунків за рахунок поєднання математичних методів аналізу сигналів і сучасних алгоритмів машинного навчання. Особливу увагу приділено забезпеченню високої продуктивності та придатності методів до використання в умовах обмежених ресурсів і необхідності обробки даних у реальному часі.

Аналіз показав, що використання трансформант перетворення Уолша-Адамара дозволяє суттєво знизити обчислювальну складність без втрати інформативності ознак. На підставі дослідження встановлено можливість впровадження стеганоаналізу безпосередньо у середовища виконання вебзастосунків, що забезпечує ефективну обробку потокових даних у реальному часі. Додатково доводиться ефективність підходів як у несліпих, так і у сліпих сценаріях аналізу.

Встановлено, що запропоновані методи забезпечують високу швидкодію: час обробки окремих об’єктів становить частки секунди, що дозволяє масштабувати рішення для аналізу великих обсягів мультимедійного контенту. Це є критично важливим для сучасних вебсервісів із великими потоками даних.

Обґрунтовано, що розроблені підходи забезпечують оптимальний баланс між точністю та обчислювальними витратами порівняно з ресурсоємними моделями глибинного навчання. Це робить їх придатними для використання у вбудованих системах захисту, де ключовими є швидкість реагування та ефективне використання ресурсів.

Запропоновано підходи до створення адаптивних систем безпеки, здатних не лише виявляти відомі загрози, а й реагувати на нові типи атак. На підставі аналізу визначено, що такі рішення можуть бути інтегровані у системи моніторингу, аналізу та автоматичного реагування, підвищуючи загальний рівень захищеності.

Результати дослідження доповнюють існуючі підходи до забезпечення кібербезпеки, акцентуючи увагу на практичній реалізації, масштабованості та ефективності роботи в реальному часі, що є критично важливим для сучасних вебзастосунків.

Основні наукові результати дисертації опубліковано в 28 працях, зокрема: 11 статей – у наукових фахових періодичних виданнях України, 1 монографія, 2 навчальних посібники, 1 авторське свідоцтво, 13 публікацій – у матеріалах міжнародних та всеукраїнських науково-практичних конференцій.

Ключові слова: вебзастосунки, штучний інтелект, машинне навчання, нейронні мережі, тестування безпеки, життєвий цикл ПЗ, DevSecOps, CI/CD, фішингові атаки, приховані канали передачі інформації, стеганографія, стеганоаналіз, перетворення Уолша-Адамара, SVD (сингулярний розклад), кодове управління.

ABSTRACT

Dyka A. I. Machine learning methods in web-based software security testing. – Qualification scientific paper in the form of a manuscript.

Dissertation for the degree of Doctor of Philosophy in specialty 122 – Computer Science (field of knowledge 12 – Information technologies). – National University "Odesa Law Academy", Odesa, 2026.

The dissertation is devoted to the development of highly effective methods for testing the security of web applications based on machine learning, focused on identifying complex and relevant cyber threats that are resistant to traditional protection methods. The research is focused on two key areas: the development of a scientific basis and practical algorithms for detecting covert steganographic channels in media files using the low-cost mathematical apparatus of the Walsh-Hadamard transform, as well as the creation of an adaptive system for combating phishing through in-depth analysis of server content and behavioral metrics in real time. Both areas are generalized into a conceptual testing model that offers new approaches to integrating security mechanisms into the software development life cycle.

The first section of the dissertation provides a comprehensive analysis of vulnerabilities and security testing issues for web applications. The software development life cycle is considered in the context of modern methodologies (Waterfall, Agile), frameworks (Scrum, Kanban), and DevOps practices (CI/CD), with an emphasis on integrating security at all stages. The importance of the DevSecOps approach is shown, which allows building in security mechanisms at the design and coding stages, and also provides system monitoring in a productive environment. A dichotomous classification of web application security is proposed: development security (DevSec) and operation security (Runtime Security), which allows for a clear structure of the place of protective measures in the life cycle. Special attention is paid to phishing attacks and the organization of covert channels of information transmission based on multimedia, which pose the most relevant and complex threats. It is shown that existing approaches have limited effectiveness and

require the development of new, adaptive methods based on artificial intelligence and steganography, capable of providing reliable protection in real time.

In the second section of the dissertation, a theoretical basis for detecting covert channels in web applications created using modern steganographic methods is formed. It is shown that code-controlled methods and SVD UV approaches are highly resistant to classical steganalysis algorithms and pose a significant threat to the security of information systems. To overcome these challenges, a mathematical apparatus based on the Walsh-Hadamard transform has been developed, characterized by low computational complexity and preserving the informativeness of the transformants. The proposed approaches allow detecting characteristic statistical and structural features caused by additional information embedding, both in the case of code-controlled methods and for SVD UV algorithms. The performed experimental research confirmed the possibility of effective application of the developed basis in both non-blind and blind analysis scenarios. The obtained results create the basis for the development of scalable steganalysis solutions capable of operating in the web environment in real time.

The third section of the dissertation presents the development of new methods for detecting covert channels in web applications based on artificial intelligence technologies. Unlike traditional statistical approaches, the proposed methods take into account the high variability of data and the ability of attackers to adapt steganographic algorithms. Two approaches for steganalysis of the code-controlled method have been developed: based on the calculation of the envelope of histograms and based on the ensemble of convolutional neural networks. The first approach provides effective extraction of key peak features and achieves an accuracy of over 93.3%, while the second demonstrates higher results (accuracy equal to 94.3%) due to the full automation of the analysis process. Additionally, a method for detecting steganographic messages created using SVD UV methods with SVM-based classification is proposed, which allows for increased accuracy and resistance to complex data modifications. A comparison with existing analogues confirmed the

advantages of the developed solutions in terms of performance, reliability, and practical suitability for integration into cyber defense systems.

The fourth section of the dissertation focuses on the issue of protecting web applications from phishing attacks, which remain one of the most relevant threats in the modern digital environment. A classification of phishing types is performed, including attacks via email, social networks, SMS, as well as the latest techniques using generative artificial intelligence models. Based on the analysis of modern approaches recommended by international organizations (OWASP, NIST, ENISA), their strengths and limitations are outlined. A new method for automated detection of malicious content is proposed, which combines byte representation of HTML documents with visualization in the form of images and further processing using machine learning algorithms. The developed model based on SVM achieved a classification accuracy of 92.45%, demonstrating a balance of precision and recall. The method is characterized by versatility, independence from the text content of pages, and the ability to integrate into browsers or server systems. This provides the practical value of the approach for real-world cybersecurity conditions. Modern methods and tools for testing web application security were systematized, and the possibilities of integrating the approaches developed by the author into cybersecurity practice were identified. The proposed steganoanalysis and AI-phishing detection methods organically complement traditional practices, playing the role of active runtime mechanisms to counter covert channels and phishing attacks. This creates a single security loop that can respond adaptively to modern challenges.

The work advances approaches to improving the efficiency of web application security testing by combining mathematical signal analysis methods with modern machine learning algorithms. Particular attention is paid to ensuring high performance and the applicability of the methods under resource-constrained conditions and real-time data processing requirements.

The analysis has shown that the use of Walsh–Hadamard transform coefficients makes it possible to significantly reduce computational complexity without loss of feature informativeness. Based on the study, the feasibility of

integrating steganalysis directly into web application runtime environments has been established, enabling efficient real-time processing of streaming data. Additionally, the effectiveness of the proposed approaches is demonstrated in both non-blind and blind analysis scenarios.

It has been established that the proposed methods provide high processing speed: the time required to process individual objects amounts to fractions of a second, which allows the solution to be scaled for analyzing large volumes of multimedia content. This is critically important for modern web services with high data throughput.

It is substantiated that the developed approaches ensure an optimal balance between accuracy and computational cost compared to resource-intensive deep learning models. This makes them suitable for use in embedded security systems, where response speed and efficient resource utilization are key requirements.

Approaches to the development of adaptive security systems are proposed, capable not only of detecting known threats but also of responding to new types of attacks. Based on the analysis, it is determined that such solutions can be integrated into monitoring, analysis, and automated response systems, thereby enhancing the overall level of security.

The results of the study complement existing approaches to cybersecurity by emphasizing practical implementation, scalability, and real-time performance, which are critically important for modern web applications.

The main scientific results of the dissertation were published in 28 papers, including 11 publications in scientific professional periodicals of Ukraine, 1 monograph, 2 textbooks, 1 author's certificate, 13 publications in the materials of international and all-Ukrainian scientific and practical conferences.

Keywords: web applications, artificial intelligence, machine learning, neural networks, security testing, software life cycle, DevSecOps, CI/CD, phishing attacks, covert information transmission channels, steganography, steganalysis, Walsh-Hadamard transform, SVD (singular value decomposition), code control.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Наукові праці, в яких опубліковані основні наукові результати дисертації:

1. Dyka A. I. Detection of SVD-based hidden data in web applications using Walsh-Hadamard transformant analysis. *Scientific notes of V. I. Vernadsky Taurida National University*. 2025. Vol. 36 (75), № 4. P. 110-118. DOI: https://doi.org/10.32782/2663-5941/2025.4.2/15

2. Dyka A. I. Detection of covert channels in web applications based on unimodality violation in the Walsh-Hadamard spectrum. *Informatics and Mathematical Methods in Simulation*. 2025. Vol.15, № 1. P. 5-14. DOI: https://doi.org/10.15276/imms.v15.no1.5

3. Dyka A. I. Intelligent methods for steganalysis of code-controlled covert messages in web applications. *Scientific notes of V. I. Vernadsky Taurida National University*. 2025. Vol. 36(75), № 2. P. 324-333. DOI: https://doi.org/10.32782/2663-5941/2025.2.2/44

4. Дика А. І., Трофименко О. Г., Ковальжі А. С. Інтелектуальне виявлення фішингових сторінок у середовищі вебзастосунків із застосуванням візуального подання. *Вчені записки ТНУ імені В.І. Вернадського*. Серія: Технічні науки. 2025. Т. 36(75), № 1. С. 298-308. DOI: https://doi.org/10.32782/2663-5941/2025.1.2/43

5. Трофименко О. Г., Дика А. І., Задерейко О. В., Шевченко О. В. Сфери застосування штучного інтелекту в розробці та тестуванні ігор. *Кібербезпека: освіта, наука, техніка*. 2025. № 1(29). С. 41-59. DOI: https://doi.org/10.28925/2663-4023.2025.29.832

6. Трофименко О. Г., Дика А. І., Лобода Ю. В., Толокнов А. А., Бондаренко М. Є. Штучний інтелект у розробці ігор. *Кібербезпека: освіта, наука, техніка*. 2025. № 3(27). С. 109-119. DOI: https://doi.org/10.28925/2663-4023.2025.27.701

7. Трофименко О., Дика А., Логінова Н., Задерейко О., Струк Н. Штучний інтелект у військових навчальних симуляторах. *Інформаційні технології та суспільство*. 2024. № 2(13). 2024. С. 89-95. DOI: https://doi.org/10.32689/maup.it.2024.2.13

8. Трофименко О. Г., Лобода Ю. Г., Гура В. І., Дика А. І., Стрілець М. І. Інструменти штучного інтелекту для системного аналізу. *Вісник Херсонського національного технічного університету.* 2024. № 4. С. 349-357. DOI: https://doi.org/10.35546/kntu2078-4481.2024.4.46

9. Трофименко О. Г., Дика А. І., Лобода Ю. Г. Аналіз уразливостей та проблем безпеки вебзастосунків. *Системні технології*. 2023. № 3(146). С. 25-37. DOI: https://doi.org/10.34185/1562-9945-3-146-2023-03

10. Трофименко О. Г., Дика А. І., Лобода Ю. Г. Аналіз інструментів тестування вебзастосунків. *Кібербезпека: освіта, наука, техніка.* 2023. № 4(20). С. 62-71. DOI: https://doi.org/10.28925/2663-4023.2023.20.6271

11. Трофименко О. Г., Савєльєва О. С., Прокоп Ю. В., Логінова Н. І., Дика А. І. Soft skills для розробників програмного забезпечення. *Кібербезпека: освіта, наука, техніка.* 2023. № 3(19). С. 6-19. DOI: https://doi.org/10.28925/2663-4023.2023.19.619

Наукові праці, які засвідчують апробацію матеріалів дисертації:

12. Дика А. І. Застосування контекстуальних ембеддінгів для виявлення ознак шкідливих введень. *Інформаційне суспільство: проблеми та перспективи:* матер. X Всеукр. наук.-практ. конф. (Одеса, 23 травня 2025 р.). Одеса. 2025. С. 158-160. DOI: https://doi.org/10.32837/11300.29842

13. Дика А. І. Застосування NLP-методів для виявлення XSS-атак // Європейські орієнтири розвитку України: нові виміри у воєнний час : матеріали міжнар. наук.-практ. конф. у 2 т. (Одеса, 16 травня 2025 р.). Одеса : Фенікс, 2025. Т. 2. С. 520-522. URL: https://hdl.handle.net/11300/30820

14. Дика А. І. Сучасні інтерактивні інструменти тестування безпеки застосунків // Інформаційні технології: моделі, алгоритми, системи (ITMAS-2024) : матеріали V Всеукр. наук.-практ. інтернет-конф. (м. Миколаїв, 30-31 жовт. 2024 р.). Миколаїв : НУК імені адмірала Макарова, 2024. С. 188-190. URL: https://itconf.nuos.edu.ua/2024/publications/analysis-of-project-management-software/

15. Дика А. І. Роль штучного інтелекту у тестуванні безпеки вебзастосунків // Інформаційне суспільство: проблеми та перспективи : матер. IX всеукр. наук.-практ. конф. (м. Одеса, 24 трав. 2024 р.). Одеса : НУ «ОЮА», 2024. С. 153-156. DOI: https://doi.org/10.32837/11300.27842

16. Дика А. І. Тестування безпеки вебзастосунків за допомогою ChatGPT // Актуальні тенденції розвитку освіти, науки та технологій : матеріали VII міжнар. наук.-практ. конф. (м. Харків-Бахмут, 15 трав. 2024 р.) Т. 2. С. 28-30. URL: https://www.nnppi.in.ua/index.php/abit/2-uncategorised/270-naukovi-konferentsiyi

17. Дика А. І., Трофименко О. Г. Аналіз потенційних загроз засобами ChatGPT для тестування безпеки вебзастосунків // Європейські орієнтири розвитку України: науково-практичний вимір в умовах воєнних викликів : матеріали міжнар. наук.-практ. конф. (м. Одеса, 26 квіт. 2024 р.) Одеса : НУ «ОЮА», 2024. С. 876-878. URL: https://hdl.handle.net/11300/28232

18. Дика А. І., Максименко А. Ж. Штучний інтелект у веброзробці: революція персоналізації користувача // Кіберпростір в умовах війни та глобальних викликів XXI століття: теорія та практика : матеріали міжнар. наук.-практ. конф. (м. Одеса, 24 листоп. 2023 р.). Одеса : НУ «ОЮА», 2023. С. 174-176. DOI: https://doi.org/10.32837/11300.27179

19. Дика А. І. Тестування штучного інтелекту: ключові виклики, стратегії вдосконалення // Електронні інформаційні ресурси: створення, використання, доступ : матеріали міжнар. наук.-практ. інтернет-конф. (м. Вінниця, 20-21 листоп. 2023 р.). Вінниця : КЗВО «Вінницька академія безперервної освіти». С. 93-95.

20. Манаков С. Ю., Дика А. І. Вибір сучасних засобів розробки мобільних застосунків // Інформаційне суспільство: проблеми та перспективи : матеріали VIII всеукр. наук.-практ. конф. (Одеса, 2 червня 2023 р.). Одеса: НУ «ОЮА», 2023. С. 86-89. DOI: https://doi.org/10.32837/11300.25263

21. Дика А. І., Лобода Ю. Г. Розвиток застосування WebAssembly у веброзробці // Європейські орієнтири розвитку України в умовах війни та глобальних викликів XXI століття: синергія наукових, освітніх та технологічних рішень: у 2 т. : матеріали міжнар. наук.-практ. конф. (м. Одеса, 19 трав. 2023 р.). Одеса : Юридика, 2023. Т. 1. С. 594-596. URL: https://hdl.handle.net/11300/27641

22. Трофименко О. Г., Дика А. І. Проблеми тестування вебсайтів е-комерції // Інформаційні технології та менеджмент у вищій освіті та науці : матеріали міжнар. наук. конф. (м. Фергана, 28 листоп. 2022 р.). Izdevnieciba «Baltija Publishing», 2022. С. 195-199.

23. Дика А. І. Регресійне тестування в Agile // Кібербезпека в сучасному світі: актуальні виклики : матеріали IV всеукр. наук.-практ. конф. (м. Одеса, 25 листоп. 2022 р.). Одеса : НУ «ОЮА», 2022. С. 44-48. URL: https://hdl.handle.net/11300/28986 ; DOI: 10.32837/11300.28986.

24. Логінова Н. І., Дика А. І., Янковський О. Г. Аналіз вразливостей систем керування базами даних // Європейський вибір України, розвиток науки та національна безпека в реаліях масштабної військової агресії та глобальних викликів XXI століття (до 25-річчя Національного університету «Одеська юридична академія» та 175-річчя Одеської школи права) : матеріали міжнар. наук.-практ. конф. (м. Одеса, 17 черв. 2022 р.). Одеса: Видавничий дім «Гельветика», 2022. Т. 1. С. 682-685. URL: https://hdl.handle.net/11300/19760

Наукові праці, які додатково відображають наукові результати дисертації:

25. Концептуальні основи захисту інформаційного суверенітету України : монографія / Задерейко О. В., Троянський О. В., Чанишев Р. І., Дика А. І. Одеса : Фенікс, 2022. 220 с. DOI: https://doi.org/10.32837/11300.22733

26. Тестування та забезпечення якості програмних систем : навч.-метод. посіб. для підгот. здобув. вищої освіти галузі знань 12 «Інформаційні технології» / Нац. ун-т «Одес. юрид. академія»; уклад.: О. Г. Трофименко, А. І. Дика ; Одеса : Фенікс, 2024. 140 с. URL: https://hdl.handle.net/11300/27246; DOI: https://doi.org/10.32837/11300.27246

27. Тестування та забезпечення якості програмних систем : навч. посіб. для підгот. здобув. вищої освіти галузі знань 12 «Інформаційні технології» / О. Г. Трофименко, А. І. Дика; Нац. ун-т «Одес. юрид. академія». Одеса: Фенікс, 2024. 195 с. URL: https://hdl.handle.net/11300/27717. ISBN 978-617-8395-88-9.

28. Комп'ютерна програма «Black Sea Hanter»: авторське свідоцтво № 129198 від 21.08.2024 / Струк Н. О., Дика А. І., Задерейко О. В., Логінова Н. І., Трофименко О. Г. https://sis.nipo.gov.ua/uk/search/detail/1821799/

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API	–	application programming interface
CI/CD	–	continuous integration / continuous delivery або continuous deployment
CMS	–	content management system
CNN	–	convolutional neural networks
CSPM	–	cloud security posture management
CSRF	–	cross-site request forgery
CZT	–	chirp z-transform
DAST	–	dynamic application security testing
DCT, ДКП	–	discrete cosines transform, дискретне косинусне перетворення
DevOps	–	development operations
DevSec	–	development security
DevSecOps	–	development, security, operations
DFT	–	discrete fourier transform
DOM	–	document object model
EXIF	–	exchangeable image file format
FN	–	false negative
FP	–	false positive
GAN	–	generative adversarial networks
GPU	–	graphics processing unit
IAST	–	interactive application security testing
IDN	–	internationalized domain names
IDS/IPS	–	intrusion detection / prevention systems
IoT	–	internet of things
JPEG	–	joint photographic experts group
LSB	–	least significant bit
MLP	–	multilayer perceptron
MPEG	–	moving picture experts group

NLP	–	natural language processing
NRCS	–	natural resources conservation service
PNG	–	portable network graphics
RAD	–	rapid application development
RASP	–	runtime application self-protection
RBF	–	radial basis function
ReLU	–	rectified linear unit
RNN	–	recurrent neural networks
SAST	–	static application security testing
SCA	–	software composition analysis
SDLC	–	software development life cycle
SSL	–	secure sockets layer
SQL	–	structured query language
SVD	–	singular value decomposition
SVM	–	support vector machine
TP	–	true positive
t-SNE	–	t-distributed stochastic neighbor embedding
URL	–	uniform resource locator
XSS	–	cross-site scripting
WAF	–	web application firewalls
WIP	–	work in progress
WHT	–	Walsh-Hadamard transform
БПЛА	–	безпілотний літальний апарат
ДІ	–	додаткова інформація
ЗПАІС	–	загальний підхід до аналізу стану й технології функціонування інформаційних систем
ПЗ	–	програмне забезпечення
ШІ	–	штучний інтелект
ШНМ	–	штучна нейронна мережа

ВСТУП

Обґрунтування вибору теми дослідження. У сучасному цифровому середовищі вебзастосунки перетворилися на критичну інфраструктуру для бізнесу, державного управління, фінансових операцій та соціальної взаємодії [1, 2]. Вони забезпечують обробку та зберігання конфіденційних даних мільйонів користувачів, що робить їх одними з головних цілей кіберзлочинців [3, 4]. Стрімкий розвиток DevOps-практик, зокрема CI/CD, прискорив життєвий цикл розробки програмного забезпечення [5], але водночас створив нові виклики у сфері забезпечення безпеки [6]. Традиційна парадигма, за якої безпека розглядалася як завершальний етап тестування, виявилася неефективною. Натомість поширення набули принцип «security as code» та підхід DevSecOps, який інтегрує механізми захисту на всіх етапах – від планування та кодування до розгортання і експлуатації [7].

Масштаб загроз підтверджується сучасними статистичними даними. За результатами міжнародних аналітичних звітів, у 2024 році середня кількість кібератак на одну організацію перевищувала 1 800 інцидентів на тиждень, що становить зростання понад 70% порівняно з попереднім роком. Загальні глобальні збитки від кіберзлочинності оцінюються у понад 10 трлн доларів США щорічно, а частка атак, пов’язаних із вебзастосунками, стабільно зростає. Крім того, обсяг світового інтернет-трафіку та кількість переданих мультимедійних даних щороку демонструють двозначні темпи приросту, що створює додаткові можливості для організації прихованих каналів передачі інформації.

Водночас, попри значний прогрес у сфері автоматизованого тестування вразливостей (SAST, DAST, IAST, SCA) [8], низка критично важливих векторів атак залишається недостатньо дослідженою і захищеною [9, 10]. Особливо це стосується вебзастосунків, що працюють із мультимедійним контентом і схильні до двох типів загроз: фішингових атак та використання прихованих каналів передавання інформації за допомогою стеганографії [11].

Фішингові атаки залишаються одним із найпоширеніших методів компрометації користувачів. Вони еволюціонували від підроблених клієнтських інтерфейсів до використання серверних вразливостей вебзастосунків для розміщення шкідливих ресурсів. Це дозволяє обходити традиційні клієнтські засоби захисту та робить проблему протидії фішингу особливо гострою у контексті серверної логіки вебзастосунків [12]. Сучасні дослідження підтверджують, що інтеграція механізмів виявлення та блокування фішингової активності безпосередньо у серверну частину систем є більш ефективною, ніж використання зовнішніх чи клієнтських засобів контролю.

Іншим не менш небезпечним вектором загроз є використання прихованих каналів у вебзастосунках, які обробляють зображення, аудіо та відео [13]. Завантаження мультимедійних файлів створює можливість для зловмисників приховувати дані за допомогою стеганографії [14]. Класичні методи виявлення, засновані на просторовому аналізі або базових статистичних відхиленнях, не здатні протистояти сучасним алгоритмам.

Класичні методи стеганоаналізу, які ґрунтуються на аналізі просторових характеристик або базових статистичних ознак, уже не відповідають рівню складності сучасних стеганографічних алгоритмів. Нові методи, що використовують адаптивне вбудовування, сингулярний розклад чи багатокомпонентні частотні області, здатні зберігати статистичні характеристики контейнера практично незмінними. Це суттєво ускладнює процес детектування та знижує ефективність наявних підходів. У результаті виникає суперечність: швидкість розвитку методів приховування інформації значно перевищує темпи вдосконалення методів виявлення. З практичної точки зору проблема має критичне значення. Приховані канали у вебзастосунках можуть використовуватися для витоку конфіденційної інформації з корпоративних і державних систем, координації шкідливих дій у розподілених бот-мережах, обходу систем контролю та цензури. Особливу небезпеку вони становлять у сфері кіберзахисту критичної інфраструктури,

військових комунікацій, фінансових сервісів та хмарних платформ, де навіть незначний витік може призвести до масштабних наслідків [1, 15]. Таким чином, підвищення ефективності систем стеганоаналізу є нагальною потребою як для наукової спільноти, так і для практики інформаційної безпеки.

Окремої уваги потребує питання швидкодії методів. Вебзастосунки функціонують у середовищі реального часу, обробляючи великі обсяги мультимедійних даних [16, 17]. Методи на основі складних перетворень, зокрема дискретного косинусного перетворення, сингулярного розкладу чи глибоких нейронних мереж, хоча й показують високу точність, є надто ресурсоємними. Це обмежує їхнє практичне застосування у масштабованих вебсервісах і знижує ефективність інтеграції у CI/CD-конвеєри [18]. Тому актуальною є розробка нових підходів, які поєднують низьку обчислювальну складність із високою точністю, здатних забезпечити виявлення прихованих каналів у реальному часі без шкоди для продуктивності системи.

У цьому контексті перспективним напрямом дослідження є використання перетворень із низькою обчислювальною складністю (наприклад, перетворення Уолша-Адамара) [19, 20] у поєднанні з методами машинного навчання та штучного інтелекту [21]. Такий підхід дозволяє створювати системи, які одночасно зберігають високу точність і адаптивність [23, 24], а також відповідають вимогам масштабованих вебплатформ [25, 26]. Крім того, особливу актуальність має виявлення сучасних стеганографічних методів з кодовим управлінням та SVD-методів [14, 16], оскільки вони демонструють підвищену стійкість до класичних атак стеганоаналізу та здатність функціонувати навіть на платформах із обмеженими обчислювальними ресурсами [27].

Таким чином, актуальність цього дослідження зумовлена суперечністю між швидкою еволюцією методів атак на вебзастосунки (зокрема організації прихованих каналів за допомогою сучасних стеганографічних алгоритмів та фішингових технік, що використовують як клієнтські, так і серверні вразливості) та недостатнім розвитком комплексних і швидкодіючих засобів

їхнього виявлення у процесі тестування безпеки програмного забезпечення. Це визначає об’єктивну потребу створення нових науково обґрунтованих методів тестування вебзастосунків, здатних ефективно виявляти і блокувати як фішингові атаки, так і стеганографічні загрози в умовах реального часу. Розв’язання цього завдання має подвійне значення: теоретичне – для розвитку наукової бази кібербезпеки у сфері тестування ПЗ та практичне – для підвищення надійності цифрових сервісів, захисту конфіденційних даних користувачів і забезпечення стійкості критично важливої інформаційної інфраструктури.

Значний внесок у вивчення життєвого циклу безпечної розробки та впровадження практик безпеки у процеси DevOps зробили такі дослідники, як Kuhrmann M. (Адаптація гнучких методологій), Prates L., Pereira R. (практики та інструменти DevSecOps), а також Sinan M., Shahin M., Gondal I. (інтеграція контролю безпеки). Найважливішим етапом життєвого циклу є тестування безпеки. Проблематиці автоматизованого тестування вебзастосунків, включно з методами статичного (SAST) та динамічного (DAST) аналізу, присвятили свої праці дослідники Li K., Croft R., Albahar M. Питання аналізу компонентів на наявність уразливостей (SCA) всебічно вивчені Imtiaz N. та Zhao L.

Особливу роль у контексті даного дослідження відіграє аналіз стеганографічних методів приховування даних, які можуть бути використані для створення прихованих каналів передачі у вебзастосунках. Основні роботи в галузі стеганоаналізу, зокрема в просторовій та частотній областях (DCT, DWT, SVD), належать Holub V., Fridrich J., Setiadi D.R.I.M., Shehab D.A., Alhaddad M.J., а також вітчизняним вченим Кобозевій А.А., Шелесту М.Є., Соколову А.В., Костирка О.В.

У галузі машинного навчання для завдань класифікації та виявлення загроз, включно з фішингом та стегоаналізом, фундаментальні дослідження проведені Jain A.K., Gupta B.B., Catal C., Kavya S., Sumathi D., de La Croix NJ, а також Eid W.M. Попри значний обсяг проведених досліджень, залишається недостатньо вивченою проблема комплексного виявлення прихованих

стеганографічних каналів у вебзастосунках на основі аналізу їхніх цифрових відбитків в області перетворення Уолша-Адамара.

Отже, попри значний доробок вітчизняних і зарубіжних дослідників у суміжних галузях, проблема комплексного виявлення фішингових атак і стеганографічних прихованих каналів у вебзастосунках залишається відкритою. Існуючі підходи або розглядають ці загрози ізольовано, або не забезпечують необхідної швидкодії для інтеграції у середовища CI/CD та DevSecOps. Це підтверджує потребу у створенні науково обґрунтованих методів, які поєднують теоретичну строгість із практичною ефективністю та придатністю до впровадження у реальні системи захисту вебзастосунків.

Зв'язок роботи з науковими програмами, планами, темами. По-перше, дослідження узгоджується зі Стратегією національної безпеки України (Указ Президента України від 14.09.2020 р. № 392/2020, з урахуванням змін, внесених Указом Президента від 06.01.2025 р. № 16/2025), де у п. 52 Розділу III визначено ключове завдання розвитку системи кібербезпеки – «гарантування кіберстійкості та кібербезпеки національної інформаційної інфраструктури, зокрема в умовах цифрової трансформації». По-друге, дисертація відповідає «Концепції розвитку штучного інтелекту в Україні» (схвалена Розпорядженням Кабінету Міністрів України від 02.12.2020 р. № 1556-р), Розпорядженню Кабінету Міністрів України «Про затвердження плану заходів з реалізації Концепції розвитку штучного інтелекту в Україні на 2021-2024 роки» від 12.05.2021 р. № 438-р, Постанові Верховної Ради України «Про затвердження завдань Національної програми інформатизації на 2022-2024 роки» від 08.07.2022 р. № 2360-IX, Постанові Кабінету Міністрів України «Про затвердження Національної економічної стратегії на період до 2030 року» від 03.03.2021 р. № 179. По-третє, результати дисертаційної роботи пов'язані з положеннями Стратегії кібербезпеки України, затвердженої Указом Президента України від 26.08.2021 р. № 447/2021 (на підставі рішення РНБО від 14.05.2021 р.). У документі визначено оновлені стратегічні цілі та завдання держави у сфері кібербезпеки, серед яких – розвиток національної

спроможності з виявлення та протидії кібератакам, захист критичної інформаційної інфраструктури, інтеграція інноваційних підходів і технологій, зокрема на основі штучного інтелекту. Дисертаційне дослідження виконано в межах плану науково-дослідної роботи кафедри інформаційних технологій Національного університету «Одеська юридична академія» «Інформаційні технології в сучасному суспільстві» на 2021-2025 роки як складової плану науково-дослідної роботи Національного університету «Одеська юридична академія» «Правове і соціальне життя сучасної України у людиноцентристському вимірі в цифрову еру» (державний реєстраційний номер 0122U000266), а також плану науково-дослідної роботи кафедри програмної інженерії «Інтелектуалізація процесів розроблення та експлуатації інформаційних систем на основі інтеграції методів штучного інтелекту та сучасних архітектурних підходів» у межах виконання теми науково-дослідної роботи Національного університету «Одеська юридична академія» «Соціально-політична стабільність в період воєнних та післявоєнних трансформацій: правові, економічні, гуманітарні, інноваційно-технічні можливості в цифрову еру» на 2026-2030 роки.

Мета і завдання дослідження. Метою роботи є підвищення ефективності виявлення фішингових атак та прихованих стеганографічних каналів у вебзастосунках шляхом розробки теоретичного базису та методів тестування безпеки на основі перетворення Уолша-Адамара і методів машинного навчання, придатних для інтеграції у CI/CD-середовище.

Для досягнення поставленої мети в роботі були визначені такі завдання:

1. Аналіз сучасного стану проблеми щодо вразливостей вебзастосунків, зокрема загроз фішингових атак та створення прихованих каналів передачі інформації засобами стеганографії в контексті життєвого циклу розробки ПЗ (CI/CD) та процесів тестування безпеки.

2. Розробка теоретичного базису для виявлення прихованих каналів, побудованих на основі концепції кодового управління, шляхом аналізу статистичних властивостей трансформант перетворення Уолша-Адамара.

3. Узагальнення розробленого теоретичного базису виявлення прихованих каналів у просторі перетворення Уолша-Адамара на випадок каналів, створених за допомогою SVD UV-методів, через аналіз характерних зсувів у їхньому спектрі.

4. Розробка методів стеганоаналізу на основі машинного навчання для виявлення стеганографічних повідомлень, сформованих сучасними методами (зокрема з кодовим управлінням та SVD UV), орієнтованих на роботу у реальному часі та інтеграцію у Runtime Security вебзастосунків для тестування мультимедійної інформації, що надходить від користувачів.

5. Формування ансамблевих підходів до стеганоаналізу сучасних стеганоалгоритмів шляхом інтеграції розроблених теоретичних засад та сучасних алгоритмів машинного навчання з подальшим дослідженням їхньої точності, стійкості та швидкодії при виявленні прихованих каналів у вебзастосунках.

6. Інтеграція алгоритмів штучного інтелекту та машинного навчання у процедури автоматизованого тестування безпеки вебзастосунків з метою створення адаптивних методів виявлення фішингу з високою точністю детектування.

7. Проведення порівняльного аналізу ефективності запропонованих методів із відомими аналогами за базовими критеріями у контексті тестування безпеки вебзастосунків.

8. Розробка підходів до інтеграції запропонованих стеганоаналітичних методів і методу захисту від фішингу у життєвий цикл вебзастосунків у межах концепції DevSecOps.

Об’єктом дослідження є процеси забезпечення інформаційної безпеки вебзастосунків у життєвому циклі розробки програмного забезпечення та його експлуатації.

Предметом дослідження є теоретичні основи та методи виявлення прихованих каналів і фішингових атак у вебзастосунках.

Методи дослідження. Для розробки теоретичної бази захисту від прихованих каналів у вебзастосунках використовувався загальний підхід до аналізу стану й технології функціонування інформаційних систем (ЗПАІС), методи матричного аналізу, спектрального та сингулярного розкладу, теорії інформації та кодування, теорії досконалих алгебраїчних конструкцій. Для створення стеганоаналітичних методів виявлення прихованих каналів, зокрема на основі трансформант дискретного косинусного перетворення, сингулярного розкладу та перетворення Уолша-Адамара, застосовувалися методи теорії сигналів, статистичного аналізу та машинного навчання. Для побудови моделей виявлення фішингових атак і адаптивних методів безпеки вебзастосунків використовувалися методи машинного навчання та штучного інтелекту. Для оцінки якісних і кількісних характеристик запропонованих методів застосовувалися методи математичного моделювання, теорія алгоритмів і методи статистичної оцінки стохастичної якості.

Наукова новизна одержаних результатів.

1. *Вперше* на основі ЗПАІС встановлено та строго доведено твердження щодо модальностей і зсувів статистичних характеристик коефіцієнтів перетворення Уолша-Адамара у контейнерах мультимедійних даних, що, на відміну від наявних підходів, які ґрунтуються переважно на евристичному аналізі просторових або DCT/SVD-представлень, дало можливість обґрунтувати ефективність використання цього перетворення як математичного апарату для побудови теоретичного базису виявлення прихованих каналів у вебзастосунках.

2. *Вперше* на основі доведених тверджень щодо статистичних характеристик трансформант перетворення Уолша-Адамара запропоновано теоретичний базис для виявлення прихованих каналів, побудованих на основі концепції кодового управління та SVD UV-методів, що, на відміну від наявних рішень, які розглядають зазначені методи ізольовано або без формалізованого спектрального узагальнення, дозволило закласти основу створення

ефективних методів стеганоаналізу для виявлення прихованих каналів у вебзастосунках.

3. *Подальшого розвитку* за рахунок застосування розробленого теоретичного базису набула технологія виявлення прихованих каналів у вебзастосунках, у межах якої запропоновано нові методи стеганоаналізу стеганоповідомлень на основі кодового управління та SVD UV-методів, які забезпечують підвищену ефективність і швидкодію порівняно з наявними аналогами.

4. *Подальшого розвитку* отримала технологія автоматизованого виявлення фішингових сторінок у вебзастосунках за рахунок використання методів обробки контенту та машинного навчання, що дозволило, на відміну від наявних аналогів, враховувати структурні й семантичні ознаки у реальному часі.

5. *Удосконалено* підхід до інтеграції методів тестування та забезпечення безпеки вебзастосунків у життєвий цикл розробки ПЗ, що дозволило поєднати традиційні засоби SAST/DAST/IAST із розробленими у дисертації інтелектуальними методами стеганоаналізу й виявлення фішингових атак, на відміну від наявних сучасних аналогів, які не забезпечують комплексної інтеграції у процеси CI/CD та DevSecOps.

Практичне значення одержаних результатів. Практична цінність роботи полягає у тому, що здобуті наукові результати були реалізовані у вигляді конкретних методів та алгоритмів, придатних до використання в прикладних системах захисту інформації. Запропоновані методи відзначаються високою обчислювальною ефективністю та відносною простотою алгоритмічної реалізації, що забезпечує їхню придатність для інтеграції у серверні або клієнтські частини вебзастосунків. Зокрема:

1. Розроблено методи виявлення стеганографічних повідомлень з кодовим управлінням на основі запропонованого математичного апарату аналізу характеристик модальностей розподілу трансформант перетворення Уолша-Адамара, реалізовані у вигляді мережі Multilayer Perceptron та

ансамблю згорткових нейронних мереж. Обидва підходи забезпечують високий рівень точності класифікації (Precision = 0.95…0.98, Recall = 0.96…0.98, F1-score ≈ 0.96…0.97), що перевищує відомі аналоги на 5.6–6.6%. При цьому метод на основі MLP характеризується високою швидкодією (виділення ознак із зображення великого розміру займає близько 0.45 с, класифікація одного зображення – 0.0044 с), що дозволяє обробляти понад 200 зображень за секунду, а метод на основі ансамблю CNN забезпечує підвищену надійність і стійкість до випадкових похибок, демонструючи загальну достовірність виявлення понад 94.3%.

2. На основі запропонованого математичного апарату дослідження змін, які вносить стеганографічний SVD UV-метод в області перетворень Уолша-Адамара, запропоновано метод виявлення прихованих каналів передавання інформації у вебзастосунках. Запропонований підхід ґрунтується на аналізі зсувів у гістограмах трансформант перетворення Уолша-Адамара та класифікації на основі ансамблю методів опорних векторів (SVM). Це дозволило досягти високих показників точності (Precision = 0.96, Recall = 0.94, F1-score ≈ 0.95) та забезпечити загальну правильність класифікації понад 90.57%.

3. Метод виявлення фішингового контенту, який поєднує побайтову візуалізацію HTML-коду та машинне навчання. Запропонований підхід забезпечує коректне відображення структури вебсторінок у вигляді числових векторів ознак і дозволяє ефективно відокремлювати легітимні ресурси від шкідливих. Точність класифікації, досягнута на тестових вибірках (92.45% при збалансованих значеннях precision, recall та F1-score), свідчить про надійність методу і його здатність з мінімальною кількістю помилок виявляти потенційно небезпечні вебресурси. Висока швидкодія та невисокі вимоги до ресурсів дозволяють реалізувати метод як модуль безпеки браузера або як компонент серверної частини вебзастосунків для оперативного аналізу посилань у режимі, наближеному до реального часу.

4. Результати дисертаційної роботи використовуються в IT-компанії EPAM та впроваджені у навчальний процес на факультеті кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія», що підтверджено відповідними актами (додаток А).

Особистий внесок здобувача. Усі наукові результати, представлені в дисертаційній роботі, отримані здобувачем самостійно. Роботи [6, 8, 12, 14, 16, 19, 20, 22, 23, 26] – одноосібні. У співавторстві опубліковано праці [1-5, 7, 9-11, 13, 15, 17, 18, 21, 24, 25, 27], де здобувач відповідала за розробку методології досліджень та практичне втілення результатів. Зокрема, внесок здобувача у зазначені роботи такий: [11] – здобувачем запропоновано інтелектуальні методи виявлення фішингових сторінок у вебзастосунках із використанням візуального подання; проведено порівняльний аналіз алгоритмів машинного навчання; побудовано прототип системи. У [3, 7, 18] – виконано системний аналіз уразливостей вебзастосунків та оцінку ефективності сучасних інструментів тестування безпеки; розроблено рекомендації щодо вдосконалення процедур тестування. У [15, 17, 21, 25] – розроблено концепції застосування методів штучного інтелекту в експериментальних програмних прототипах. У [2, 9, 10] – проведено аналіз проблем тестування вебсайтів електронної комерції; розроблено методологічні підходи для оцінки ефективності тестування та забезпечення безпеки користувацьких даних. У [4, 5] – досліджено сучасні засоби розробки мобільних застосунків; виконано аналіз вразливостей систем керування базами даних; здійснено порівняльний аналіз платформ та інструментів; запропоновано рекомендації щодо оптимізації розробки. У [1] – розглянуто сучасні умови та технології інформаційного впливу на громадян держави, які цілеспрямовано впроваджуються у суспільство з метою зламу інформаційного суверенітету держави. У [13] – досліджено застосування WebAssembly у веброзробці; розроблено методику інтеграції WebAssembly у сучасні вебсервіси; сформовано рекомендації щодо підвищення продуктивності та безпеки. У [24] – проведено аналіз потенційних загроз вебзастосунків;

запропоновано практичні підходи до оцінки безпеки вебсервісів із використанням штучного інтелекту. У [27] – сформовано методологію оцінки soft skills розробників програмного забезпечення та інтегровано її в навчальні та практичні модулі для ІТ-фахівців.

Апробація результатів дисертації. Положення, висновки та рекомендації, викладені у дисертації, доповідалися на: міжнародній науково-практичній конференції «Європейський вибір України, розвиток науки та національна безпека в реаліях масштабної військової агресії та глобальних викликів XXI століття» (м. Одеса, 17 червня 2022 р.); всеукраїнській науково-практичній конференції «Кібербезпека в сучасному світі: актуальні виклики» (м. Одеса, 25 листопада 2022 р.); міжнародній науковій конференції «Інформаційні технології та менеджмент у вищій освіті та науці» (м. Фергана, Республіка Узбекистан, 28 листопада 2022 р.); міжнародній науково-практичній конференції «Європейські орієнтири розвитку України в умовах війни та глобальних викликів XXI століття: синергія наукових, освітніх та технологічних рішень» (м. Одеса, 19 травня 2023 р.); всеукраїнській науково-практичній конференції «Інформаційне суспільство: проблеми та перспективи» (м. Одеса, 2 червня 2023 р.); міжнародній науково-практичній конференції «Електронні інформаційні ресурси: створення, використання, доступ» (м. Вінниця, 20-21 листопада 2023 р.); міжнародній науково-практичній конференції «Кіберпростір в умовах війни та глобальних викликів XXI століття: теорія та практика» (м. Одеса, 24 листопада 2023 р.); міжнародній науково-практичній конференції «Європейські орієнтири розвитку України: науково-практичний вимір в умовах воєнних викликів» (м. Одеса, 26 квітня 2024 р.); міжнародній науково-практичній конференції «Актуальні тенденції розвитку освіти, науки та технологій» (м. Харків – Бахмут, 15 травня 2024 р.); всеукраїнській науково-практичній конференції «Інформаційне суспільство: проблеми та перспективи» (м. Одеса, 24 травня 2024 р.); всеукраїнській науково-практичній інтернет-конференції «Інформаційні технології: моделі, алгоритми, системи» (м. Миколаїв, 30-31

жовтня 2024 р.); міжнародній науково-практичній конференції «Європейські орієнтири розвитку України: нові виміри у воєнний час» (м. Одеса, 16 травня 2025 р.); всеукраїнській науково-практичній конференції «Інформаційне суспільство: проблеми та перспективи» (м. Одеса, 23 травня 2025 р.).

Публікації. Матеріали дисертації повною мірою викладені у 28 публікаціях, з яких 24 наукові публікації розкривають основний зміст дисертації, зокрема, опубліковано 11 статей у наукових виданнях, включених до Переліку наукових фахових видань України з технічних наук. Опубліковано: 13 тез доповідей у матеріалах всеукраїнських та міжнародних наукових конференцій, які засвідчують апробацію матеріалів дисертації; 4 наукові праці, які додатково відображають наукові результати дисертації, зокрема авторське свідоцтво України на комп'ютерну програму.

Структура і обсяг дисертації. Дисертація складається зі переліку умовних позначень, вступу, 4 розділів, загальних висновків, списку використаних джерел (237 найменувань) та 2 додатків. Загальний обсяг дисертації складає 218 сторінок, з них основний текст – 177 сторінок, в тому числі 29 рисунків, 5 таблиць. Додатки розміщені на 13 сторінках.

РОЗДІЛ 1. АНАЛІЗ УРАЗЛИВОСТЕЙ ТА ПРОБЛЕМ ТЕСТУВАННЯ БЕЗПЕКИ ВЕБЗАСТОСУНКІВ

1.1. Цифрова трансформація та безпечна розробка вебзастосунків

У сучасному інформаційному суспільстві вебзастосунки відіграють ключову роль, оскільки стали невіддільною частиною майже всіх сфер діяльності людини [28]. Вони забезпечують зручний доступ до інформаційних ресурсів, електронних послуг, комунікаційних платформ та інструментів для бізнесу, науки й освіти. Завдяки розвитку вебтехнологій відбувається цифровізація державного управління, фінансових операцій, медичних сервісів, електронної комерції та соціальних мереж. Таким чином, вебзастосунки формують основу сучасної цифрової інфраструктури та впливають на ефективність роботи організацій і комфорт повсякденного життя користувачів.

Сучасна розробка програмного забезпечення спирається на різні підходи до організації процесу створення продукту, кожен із яких виконує певну роль у забезпеченні якості та ефективності розробки. Методології управління розробкою, як-от Waterfall (каскадна модель) [29] та Agile [30], визначають загальні принципи організації процесу. Waterfall має чітку послідовність етапів – аналіз вимог, проєктування, реалізація, тестування та впровадження, що забезпечує структурованість, проте обмежує гнучкість у разі змін. Agile, навпаки, є ітеративною, гнучкою парадигмою, орієнтованою на швидке реагування на зміни вимог і неперервне вдосконалення продукту.

У межах підходу Agile застосовуються відповідні фреймворки та методи організації розробки, зокрема Scrum і Kanban. Scrum є структурованим фреймворком для ітеративної розробки ПЗ через спринти, визначення ролей (таких як Product Owner, Scrum Master, команда розробки), а також регулярні зустрічі для планування, щоденного моніторингу та оцінки прогресу. Kanban, зі свого боку, є методом управління потоком робіт, який забезпечує

візуалізацію завдань, обмеження кількості незавершених завдань (Work In Progress, WIP) та поступову оптимізацію процесів без фіксованих ітерацій.

На відміну від методологій та фреймворків, CI/CD (Continuous Integration / Continuous Delivery або Continuous Deployment) [31, 31] є процесною практикою, яка концентрується на технологічній автоматизації: неперервній інтеграції коду, автоматизованому тестуванні та швидкій доставці змін у продуктивне середовище. CI/CD не визначає, як організувати команду чи планувати роботу, а ефективно підтримує реалізацію вибраної методології, забезпечуючи стабільність, якість і безпеку продукту на всіх етапах життєвого циклу.

Таким чином, сучасний підхід до розробки вебзастосунків базується на поєднанні класичних та гнучких методологій управління життєвим циклом програмного забезпечення (зокрема Waterfall і Agile), реалізованих через відповідні фреймворки (наприклад, Scrum) і методи управління потоком робіт (зокрема Kanban), а також на впровадженні процесних практик DevOps, включно з неперервною інтеграцією та доставкою (CI/CD) [33]. Така інтеграція дозволяє одночасно організувати ефективний робочий процес, зберегти гнучкість у змінах та забезпечити автоматизовану підтримку якості, стабільності та безпеки програмного продукту.

У цьому контексті особливе значення має саме процесна практика CI/CD, яка забезпечує неперервну інтеграцію, тестування та доставку змін у коді [34]. Вона дозволяє поєднати гнучкість Agile-підходів із високою стабільністю та автоматизацією процесів, що робить її ключовою практикою у сучасній розробці вебзастосунків.

Життєвий цикл [35, 36] розробки вебзастосунків (Software Development Life Cycle, SDLC) – це послідовність етапів (рис. 1.1), які проходить програмний продукт від моменту виникнення ідеї до його виведення з експлуатації.

Рисунок 1.1 – Основні етапи розробки програмного забезпечення

Життєвий цикл розробки вебзастосунку в контексті CI/CD має такі етапи:

1. Планування (Planning). На цьому етапі визначаються вимоги до функціоналу, безпеки та продуктивності. Формуються беклог завдань і план спринтів або релізів, що задають основу для подальшої розробки [33].

2. Написання коду (Coding). Розробники реалізують функціонал згідно з вимогами. Кожна зміна коду регулярно інтегрується в спільний репозиторій, що є початковим кроком для безперервної інтеграції [35].

3. Безперервна інтеграція (Continuous Integration). Код автоматично збирається, проходить юніт-тести та статичний аналіз. Це дозволяє швидко виявляти помилки і підтримувати стабільність основної гілки коду [36].

4. Автоматизоване тестування (Automated Testing). Проводиться тестування функціональності, інтеграційних взаємодій, безпеки та продуктивності. Автоматизація дозволяє забезпечити високу якість коду та швидке виявлення дефектів [37].

5. Збірка артефактів (Build). Програмний продукт пакується у вигляді готових артефактів (наприклад, контейнерів, виконуваних файлів або бібліотек), готових для доставки у тестове чи продуктивне середовище.

6. Безперервна доставка / розгортання (Continuous Delivery / Deployment). Артефакти автоматично доставляються на тестові або продуктивні середовища. Continuous Delivery передбачає ручне підтвердження перед релізом, а Continuous Deployment – повністю автоматизоване розгортання.

7. Моніторинг та спостереження (Monitoring). Після розгортання здійснюється постійний моніторинг працездатності, продуктивності та безпеки системи, що дозволяє виявляти аномалії та реагувати на проблеми в реальному часі.

8. Зворотний зв'язок і вдосконалення (Feedback & Improvement). На основі даних моніторингу та відгуків користувачів команда аналізує результати, вносить зміни до вимог та коду, що запускає новий цикл CI/CD.

Окремо варто наголосит, що сучасні вебзастосунки розробляються та експлуатуються за умов підвищених вимог до безпеки. Інтеграція підходів DevSecOps [38, 39] забезпечує долучення механізмів захисту на всіх етапах життєвого циклу ПЗ – від проєктування і написання коду до тестування, розгортання та підтримки. Безпека більше не розглядається як завершальний етап, а стає невіддільною частиною процесів розробки та експлуатації. Це дозволяє своєчасно виявляти вразливості, підвищувати стійкість вебзастосунків до кіберзагроз та зменшувати ризики, пов'язані з компрометацією даних і сервісів. Таким чином, поєднання принципів DevOps із системним підходом до безпеки формує основу сучасної практики створення надійних вебзастосунків [40, 41].

При цьому четвертий етап за процесною практикою CI/CD, а саме тестування безпеки вебзастосунків, є критично важливим процесом, який визначає захищеність даних користувачів, стійкість бізнес-процесів та довіру до програмного продукту. Систематичне тестування безпеки дозволяє своєчасно виявляти вразливості, які можуть бути використані зловмисниками, мінімізувати ризики відмови системи та запобігати потенційним кібератакам [42, 43]. На відміну від класичного функціонального тестування, воно

зосереджується на перевірці конфіденційності, цілісності та доступності даних, а також на здатності системи протистояти сучасним загрозам.

Тестування безпеки у CI/CD забезпечує [44, 45]:

– виявлення вразливостей на ранніх стадіях розробки, що знижує вартість та складність їх усунення;

– перевірку стійкості вебзастосунку при внесенні змін у код та інтеграції нових компонентів;

– оцінку рівня захисту даних і бізнес-логіки від потенційних атак;

– контроль продуктивності та надійності системи за умов шкідливого впливу;

– гарантію відповідності сучасним стандартам безпеки та нормативним вимогам.

1.2. Тестування безпеки вебзастосунків

У сучасній науковій літературі часто підкреслюється, що гарантування безпеки повинно бути вбудоване в усі етапи життєвого циклу вебзастосунків: від прототипування та написання перших рядків коду до їх експлуатації у продуктивному середовищі [10, 45]. Водночас, хоча ця ідея широко цитується, її практична реалізація залишається складною через необхідність координації численних інструментів та методів, інтеграції автоматизованих перевірок у процеси CI/CD і застосування підходів DevSecOps, які передбачають спільну відповідальність усіх учасників розробки за безпеку.

З нашого погляду, питання безпеки вебзастосунків потребують максимальної простоти та прозорості у визначеннях і підходах. Абстрактні або надмірно загальні формулювання часто ускладнюють практичну інтеграцію безпеки в реальні процеси розробки та експлуатації. Спираючись на ґрунтовний огляд численних досліджень у сфері безпеки програмного забезпечення, доцільно представити безпеку вебзастосунків у дихотомічному вигляді, розділивши її на два ключові напрями:

1. Безпека розробки (DevSec). Цей аспект охоплює впровадження принципів безпечного програмування, використання статичного аналізу коду, автоматизоване тестування на вразливості, а також інтеграцію політик безпеки у конвеєри CI/CD. Такий підхід дозволяє зменшувати кількість потенційних дефектів і вразливостей ще на ранніх етапах життєвого циклу ПЗ.

2. Безпека експлуатації (Runtime Security). Вона стосується забезпечення захисту системи вже після її розгортання у продуктивному середовищі. Сюди належать моніторинг подій безпеки, виявлення та реагування на аномалії, контроль доступу, захист від актуальних атак (наприклад, SQL-ін’єкцій, XSS чи DDoS), а також підтримка неперервної відповідності нормативним вимогам.

Таке дихотомічне подання дозволяє не лише чітко структурувати місце безпеки у життєвому циклі вебзастосунків, а й узгодити її із сучасними практиками DevOps та DevSecOps [46, 47, 48]. Як наслідок, безпека перестає бути окремим завершальним етапом, а інтегрується у всі процеси розробки та експлуатації, що відповідає сучасним тенденціям в академічних дослідженнях і практичній індустрії.

Завдяки такому підходу можна класифікувати основні методи тестування безпеки ПЗ [48, 60]: SAST (Static Application Security Testing) з аналізом вихідного коду без його виконання; DAST (Dynamic Application Security Testing), який ґрунтується на моделюванні атак під час виконання програми; IAST (Interactive Application Security Testing), що поєднує статичний та динамічний аналіз у реальному середовищі; та RASP (Runtime Application Self-Protection), який інтегрується безпосередньо в застосунок і забезпечує захист під час його роботи (рис. 1.2). Окрім цих напрямів, є й інші методи тестування, як-от [50, 51]: SCA (Software Composition Analysis) для виявлення вразливостей у бібліотеках та залежностях; Fuzzing для перевірки застосунків випадковими або некоректними вхідними даними; а також penetration testing, що імітує дії реального зловмисника з метою комплексної оцінки стійкості системи тощо.

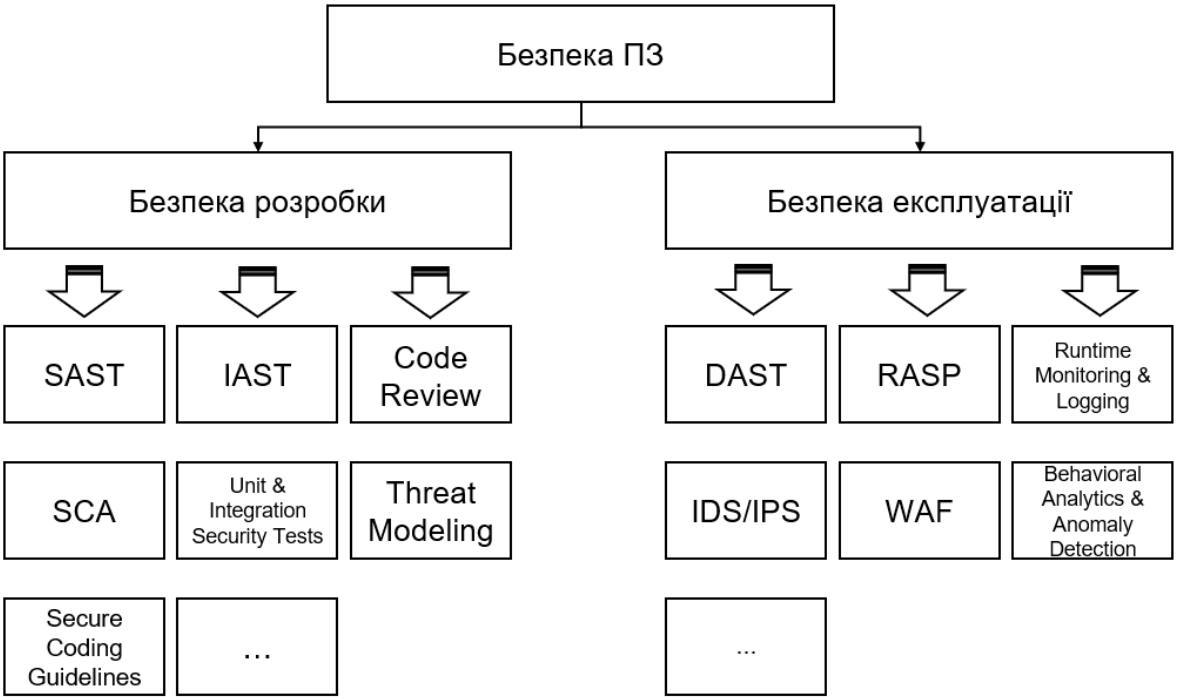


Рисунок 1.2 – Дихотомічне подання методів тестування програмного забезпечення

У межах дихотомічного представлення безпеки вебзастосунків безпека розробки (DevSec) охоплює заходи, спрямовані на попередження та мінімізацію вразливостей ще на етапі створення програмного забезпечення. Вона інтегрує принципи безпечного програмування, автоматизованого тестування коду та дотримання стандартів безпеки, забезпечуючи високу якість і надійність продукту до його розгортання у продуктивному середовищі.

Основні методи, що застосовуються у межах безпеки розробки, містять:

1. SAST (Static Application Security Testing) [7, 52] – статичний аналіз вихідного або байт-коду без його виконання. Метод дозволяє виявляти синтаксичні та логічні помилки, потенційні вразливості та недоліки безпеки на ранніх стадіях розробки. SAST забезпечує глибоку перевірку структури коду, контроль за дотриманням безпечних практик програмування та відповідність стандартам (наприклад, OWASP, CERT).

2. IAST (Interactive Application Security Testing) [52, 53] – інтерактивне тестування, яке поєднує переваги статичного та динамічного аналізу, проводиться під час виконання тестових сценаріїв і дає змогу більш точно виявляти вразливості у конкретних потоках виконання коду. IAST дозволяє виявляти складні логічні помилки та інтеграційні вразливості, що не завжди видно при традиційному SAST [54].

3. Code Review / Peer Review [55, 56] – ручний або автоматизований аналіз коду колегами для виявлення логічних помилок, антипатернів та потенційних вразливостей. Цей метод дозволяє поєднати людський досвід з автоматичними інструментами контролю якості коду.

4. Software Composition Analysis (SCA) [57, 58] – аналіз сторонніх бібліотек і компонентів на наявність відомих уразливостей та ліцензійних обмежень, що є критично важливим у сучасних вебзастосунках, які активно використовують open-source компоненти.

5. Unit та Integration Security Tests [59] – юніт-тести та інтеграційні тести, орієнтовані на безпеку, які перевіряють критичні ділянки коду та взаємодію між модулями на наявність потенційних уразливостей.

6. Threat Modeling [60] – формалізований підхід до прогнозування та аналізу потенційних загроз на етапі проєктування, що дозволяє визначити критичні точки системи та спланувати превентивні заходи.

7. Secure Coding Guidelines / Static Analysis Rules [61] – впровадження політик безпечного програмування, включно зі стандартами та правилами для інструментів статичного аналізу, що формалізують процес розробки безпечного коду.

Таким чином, безпека розробки є фундаментальною складовою дихотомічної моделі безпеки вебзастосунків, яка забезпечує превентивний рівень захисту та значно знижує ризик виникнення критичних вразливостей на подальших етапах життєвого циклу.

У межах дихотомічного подання безпеки вебзастосунків безпека експлуатації (Runtime Security) охоплює заходи, спрямовані на захист системи

та даних користувачів під час фактичного функціонування застосунку у продуктивному середовищі. На відміну від безпеки розробки, яка є превентивною, безпека експлуатації зосереджується на виявленні та нейтралізації атак у режимі реального часу, забезпеченні стійкості до зовнішніх загроз та підтримці безперервності бізнес-процесів [62].

Основні методи та підходи до забезпечення безпеки експлуатації:

1. DAST (Dynamic Application Security Testing) [63] – динамічне тестування, що імітує поведінку потенційного зловмисника під час роботи застосунку. DAST дозволяє виявляти вразливості, які проявляються лише під час виконання коду, такі як ін’єкції, помилки аутентифікації та авторизації, вразливості у взаємодії з базами даних і сервісами.

2. RASP (Runtime Application Self-Protection) [54, 64] – інтегровані механізми захисту, які працюють всередині застосунка під час його виконання. RASP відстежує поведінку запитів і реакції системи, виявляє підозрілі дії та автоматично блокує атаки без потреби в зовнішньому втручанні.

3. Runtime Monitoring та Logging [65] – постійний моніторинг подій безпеки, журналювання активності користувачів та системних подій з метою виявлення аномалій, підозрілих патернів і потенційних атак.

4. Intrusion Detection / Prevention Systems (IDS/IPS) [66] – системи виявлення та запобігання вторгнень, які аналізують трафік і поведінку застосунку у реальному часі для блокування атак та запобігання витоку даних.

5. Web Application Firewalls (WAF) [67] – мережеві або програмні фільтри, що контролюють вхідний і вихідний трафік вебзастосунку та захищають його від типових атак, таких як SQL-ін’єкції, XSS, CSRF тощо.

6. Behavioral Analytics та Anomaly Detection – методи, що використовують статистичні та машинні моделі для виявлення нетипової поведінки користувачів або системи, що може свідчити про спробу компрометації.

Попри значний розвиток методів безпеки експлуатації, ця область залишається менш дослідженою порівняно з безпекою розробки, особливо у частині захисту вебзастосунків від сучасних, складних загроз. Серед таких актуальних проблем вирізняється захист від фішингових атак, що імітують легітимні сервіси для викрадення облікових даних, а також виявлення та запобігання організації прихованих каналів комунікації (covert channels) із застосуванням стеганографічних методів, що дозволяють зловмисникам обходити традиційні механізми контролю і втручання. Ці напрямки є відкритими для досліджень та розвитку інструментів захисту вебзастосунків у продуктивному середовищі.

1.3. Різновиди інструментів тестування веббезпеки

Аналізу засобів тестування веббезпеки присвячено не так багато досліджень. Переважно дослідників цікавлять проблеми тестування та аналіз можливих уразливостей ПЗ. У статтях [68, 69] проаналізовано методологію та критерії, які використовуються для кількісної оцінки якості програмних сканерів безпеки вебзастосунків, зазначено, що недоліки й обмеження таких сканерів пов’язані з низьким охопленням тестування і неточними результатами. Автори роботи [70] дійшли висновку, що наявні рішення безпеки мають певні обмеження, в них відсутній індивідуальний підхід, який здатен комплексно інтегрувати методи тестування безпеки в усі етапи SDLC та охопити усі можливі ризики, характерні для якогось прикладного ПЗ. Адже різне прикладне ПЗ стикається з унікальними ризиками, які виникають через їх взаємозв’язки, залежність від пристроїв Інтернету речей і вплив різноманітних джерел даних. У дослідженні [71] порівнюються результати тестування різних інструментів автоматизованого і ручного тестування, зазначається, що ручне тестування залишається важливим, оскільки є такі типи вразливостей, які можна виявити лише засобами ручного тестування і досвіду тестувальника. Проведений аналіз публікацій виявив актуальність

пошуку ефективних інструментів щодо мінімізації ризиків та вразливостей безпеки сайтів.

Поширеність порушень безпеки вебзастосунків та важливість їх запобігання зробило тестування безпеки невіддільною складовою життєвого циклу розробки відповідного ПЗ, яка має виявляти вразливості, пов'язані із забезпеченням цілісного підходу до захисту програми від хакерських атак, вірусів, несанкціонованого доступу до конфіденційних даних. Тестування веббезпеки перевіряє бізнес-логіку, проблеми автентифікації та авторизації, кодування вхідних і вихідних даних, а також інші можливі ризики, щоб виявити уразливі місця в безпеці й переконатися, що всі функції вебзастосунків безпечні [23].

Для тестування безпеки важливо враховувати різницю між ризиком та ефективністю розроблених тестів безпеки щодо пом’якшення відповідного ризику безпеки для аналізу рентабельності інвестицій у безпеку [72].

Незалежно від використовуваної методології та практик розробки ПЗ (Agile, DevOps / DevSecOps, швидкої розробки застосунків (RAD) тощо), варто інтегрувати тести безпеки у робочі процеси CI/CD, щоб підтримувати безпеку на належному рівні й виявляти наявні уразливості. Методологія Agile та практики DevOps змінили способи розробки та доставки програмного забезпечення і таким чином прискорили життєвий цикл розробки від написання коду до релізу. Зараз групи розробників випускають ПЗ неперервно і значно швидше, ніж раніше, адже вони автономно приймають рішення про використовувані технології та їх впровадження. DevSecOps надає різноманітні інструменти і технології для усунення недоліків та забезпечення швидкого автоматизованого оцінювання безпеки в рамках конвеєра CI/CD. Проте і традиційні, і новітні наскрізні підходи до забезпечення безпеки вебзастосунків, включно з DevSecOps, матимуть проблеми, якщо інструменти та процеси тестування безпеки залишати незмінними без оновлення та вдосконалення [21].

Для виявлення уразливостей безпеки є різні інструменти, підходи та практики тестування безпеки щодо їх застосування [6, 7, 73] (рис. 1.3).

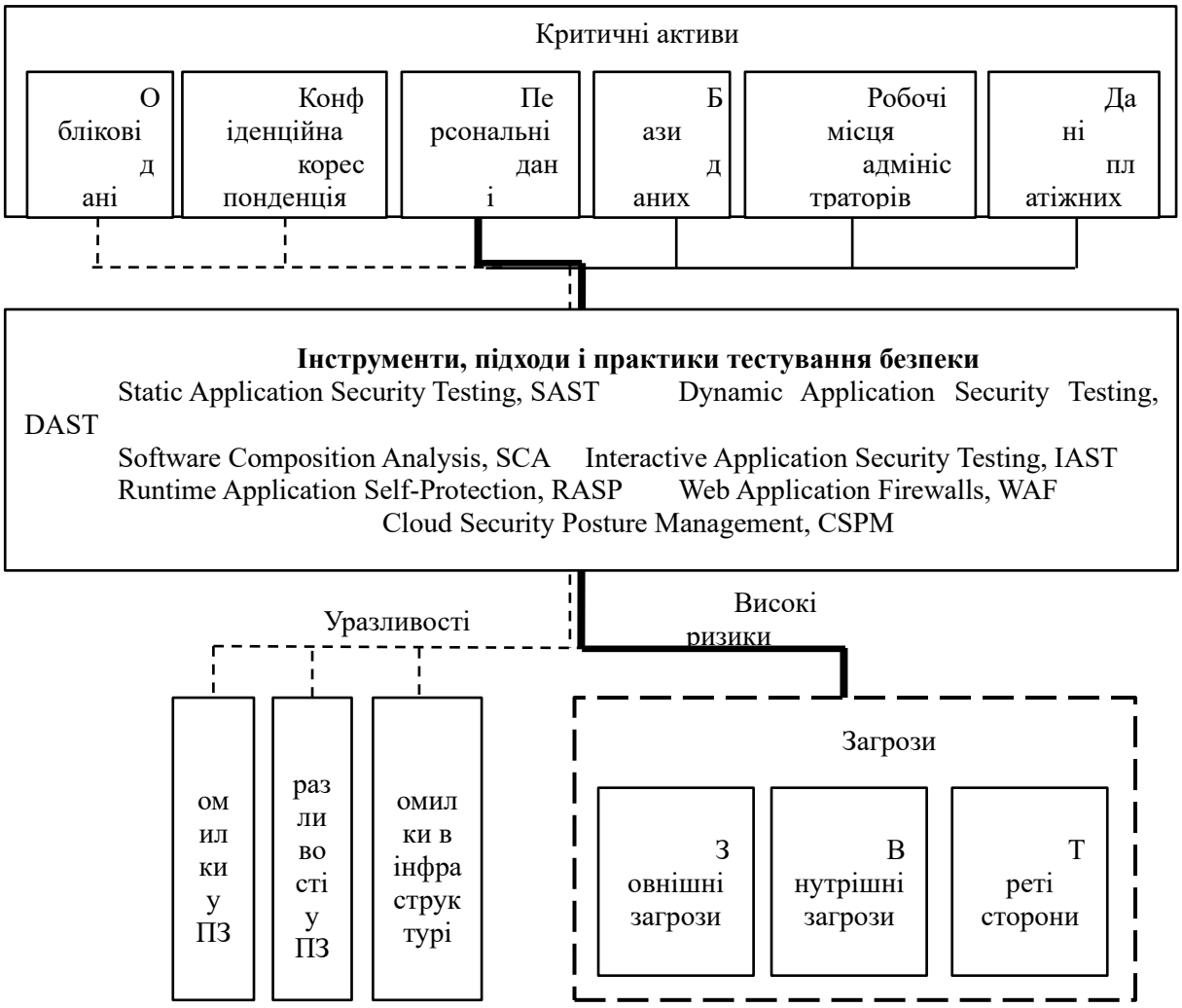


Рисунок 1.3 – Виявлення уразливостей безпеки сайтів та вебзастосунків

Всі інструменти тестування безпеки не є рівнозначними й взаємозамінними, оскільки вони мають різну специфіку оцінки, виявлення та реагування на ті чи інші різновиди вразливостей [3, 74, 75]:

1) *інструмент безпеки DAST* є, по суті, методом тестування чорної скриньки, який можна інтегрувати на ранньому етапі життєвого циклу розробки ПЗ. Він проводить оцінку вразливостей ззовні, оскільки не має доступу до вихідного коду програми і спрямований на те, щоб зменшити ризики та виявити вразливості, якими може скористатися зловмисник [76].

DAST слід виконувати до запущеної програми перед її розгортанням в середовищі, подібному до робочого. Це дозволяє використовувати підхід «ззовні всередину» для тестування застосунків на наявність умов експлуатації, які неможливо виявити у статичному стані;

2) *SAST*, також відоме як тестування білої скриньки, сканує та аналізує програму ще до компіляції коду. Цей підхід використовують для перевірки внутрішньої структури ПЗ та коректності її інтеграції із зовнішніми системами. SAST досліджує код задля виявлення можливих SQL-ін'єкцій та інших уразливостей безпеки ПЗ. У DevSecOps це тестування зазвичай інтегрується в середовища розробки для оперативного зворотного зв’язку про ризики безпеки.

Методи SAST і DAST використовуються по-різному та доповнюють один одного. І те, й інше тестування є доцільними й корисними для комплексного тестування безпеки. Вони знаходять різні типи уразливостей та ефективні на різних фазах життєвого циклу розробки ПЗ;

3) *інструменти SCA* зазвичай аналізують програми в процесі розробки й потрібні для ідентифікації вбудованих open source компонентів і бібліотек та виявлення в них уразливостей [77]. SCA доповнює SAST, адже знаходить уразливості, які неможливо виявити шляхом перевірки вихідного коду;

4) *IAST* поєднує підходи SAST і DAST, що дозволяє проявити переваги їх обох. Наприклад, щодо покриття коду, то і статичне, і динамічне тестування пропускають великі частини програмного коду: SAST не перевіряє бібліотеки та фреймворки, що значно обмежує аналіз уразливостей, а DAST може перевіряти лише відкриту поверхню програми. IAST же перевіряє всю програму зсередини, включно з бібліотеками, фреймворками, потоками даних, конфігураціями, HTTP-запитами та відповідями на них, інформацією про внутрішні підключення тощо. Тож в IAST покриття всієї кодової бази значно краще [78]. Крім того, статичні й динамічні інструменти погано масштабуються і зазвичай потребують експертів для налаштування, запуску та інтерпретації результатів. При застосуванні IAST розмір і складність програми

ніяк не впливають на продуктивність тестування, тобто можна безпроблемно тестувати великі за розміром програми. Оскільки інструменти інтерактивного тестування працюють всередині програми, то можна у режимі реального часу виявляти й усувати уразливості практично без помилкових спрацьовувань і з миттєвим зворотним зв'язком. І при цьому програма тестується не періодично, як при SAST і DAST, а безперервно й автоматично [55]. Важливо, що IAST легко інтегрується з наявними тестуваннями безпеки й дозволяє використовувати власні правила для персоналізації покриття загроз для конкретних підприємств;

5) *інструмент самозахисту застосунку під час виконання – RASP* – дозволяє на розгорнутих програмах безперервно відстежувати та аналізувати поведінку і структуру застосунку, повідомляти про атаки та блокувати неавторизовані дії зловмисника. Інструмент RASP інтегрується в програму та використовує підхід, подібний до штучного інтелекту, щоб постійно перехоплювати будь-які виклики до програми та перевіряти їхню безпеку. Він перевіряє та позначає трафік зловмисного програмного забезпечення, що надходить до програми, ще до того, як таке зловмисне ПЗ буде запущено всередині програми. Попри те, що RASP створює додаткове інфраструктурне навантаження на продуктивність програмного середовища, такі засоби можна використовувати проти атак на вебзастосунок без втручання людини. RASP зосереджено на захисті окремої програми, а тому його не використовують для захисту всієї мережі. Це добре з позиції пріоритету безпеки, оскільки йому потрібно лише контролювати кожен вхід, вихід і внутрішній процес у програмі, яку він захищає [75]. RASP працює всередині програми під час її виконання і є, по суті, інструментом безпеки, а не тестування [76]. Він інтегрується з програмою і може контролювати виконання цієї програми, захищаючи її, навіть якщо захист периметра мережі порушено, а програма містить уразливості безпеки;

6) *брандмауери вебпрограм (WAF)* відстежують трафік на рівні програм, виявляють потенційні атаки та спроби використання уразливостей

шляхом зіставлення шаблонів за попередньо визначеними правилами. WAF пом’якшує атаки DDoS і добре справляється з блокуванням атак грубої сили, SQLi та XSS [16]. Він захищає програми, фільтруючи, відстежуючи та блокуючи зловмисні запити або трафік, що надходять у програму. При цьому, як і для RASP, усі дії фіксуються у журналі подій безпеки для можливості подальшого аудиту та виявлення відхилень у поведінці програмної системи. WAF є гнучким інструментом і підтримує гібридне розгортання. Зокрема, WAF може виявляти атаки з використанням автоматизованих засобів. Проте за всіх переваг WAF не покриває автоматизовану перевірку всіх вразливостей застосунку. Є багато способів обходу WAF, які дозволяють зловмиснику продовжувати атаки на систему. Застосування брандмауера недостатньо, якщо мережа захищена неналежним чином. Для перевірки забезпечення постійного захисту та безпеки корпоративної мережі доцільно проводити тестування на проникнення через міжмережевий екран. Тести на проникнення в брандмауер є критично важливими і можуть допомогти групам безпеки виявити уразливі місця в архітектурі мережі [7];

7) *CSPM* автоматизує керування ресурсами і сервісами хмарних середовищ, включно з візуалізацією та оцінкою стану захисту, виявленням помилкових конфігурацій для забезпечення безпеки корпоративних хмарних застосунків. Сучасні компанії та організації мають складні інформаційні системи, якими може бути важко керувати через численні ресурси та обов’язки, а тому важко відстежити втрату та нецільове використання основних активів. Рішення Cloud Security Posture Management можуть рекомендувати або автоматично застосовувати методи безпеки на основі внутрішньої політики організації. До функціоналу CSPM входить: візуалізація та оцінка стану безпеки; виявлення помилкових конфігурацій у хмарній інфраструктурі, які можуть залишити неконтрольованими потенційні ризики та вектори атак; моделювання та активне застосування еталонних політик; захист від атак та внутрішніх загроз; розвідка загроз безпеки хмарного середовища для виявлення вторгнень у хмару; відповідність нормативним

вимогам і практичним рекомендаціям. CSPM забезпечує для організацій швидкі й ефективні заходи хмарного захисту та автоматизовані передові методи DevSecOps. Серед популярних рішень CSPM, якими можна усунути уразливості хмарної служби, є такі, як-от: CrowdStrike, Prisma Cloud, Continuity Software, CloudGuard Management, BMC Cloud Security, Lacework, Fugue тощо [77]. Оскільки інструменти CSPM працюють із хмарними ресурсами, то і самі вони є хмарними. Ці системи безпеки повинні мати можливість перетинати платформи та перевіряти роботу численних ресурсів, щоб усунути уразливі місця і забезпечити безпеку даних компаній та організацій у разі складних і гібридних хмарних систем [7].

1) Інструменти SAST

Інструменти статичного тестування безпеки застосунків (Static Application Security Testing, SAST) аналізують вихідний код і допомагають розробникам виявляти недоліки ще під час кодування. Процес SAST ще відомий як тестування білої скриньки. Цей підхід тестувальники використовують, щоб перевірити внутрішню структуру програмного забезпечення та побачити, як воно інтегрується із зовнішніми системами. Інструменти статичного аналізу коду використовують такі методи: аналіз синтаксису, аналіз потоку даних і аналіз безпеки. Синтаксичний аналіз перевіряє вихідний код на наявність синтаксичних помилок подібно до того, як редактори документів, наприклад Microsoft Word, виділяють граматичні помилки. Аналіз потоку даних програмно аналізує потік даних на виявлення проблем зациклення. Нарешті, аналіз безпеки перевіряє вихідний код на наявність вразливостей у безпеці, якими можуть скористатися під час атаки.

Інструменти статичного аналізу коду допомагають у створенні якісного коду для будь-якого програмного проєкту. Низька якість коду може спричинити зниження ефективності, проблеми з масштабованістю, вразливості безпеки тощо. Якщо ці проблеми не вирішити на етапі розробки, це може спричинити величезні проблеми надалі.

З іншого боку, на практиці, хоча інструменти SAST корисні для раннього виявлення вразливостей, вони можуть генерувати помилкові спрацьовування та пропускати вразливості [79].

Наразі на ринку є десятки різних інструментів безпеки, які пропонують різні набори послуг безпеки програмних продуктів. Більшість наявних на ринку інструментів безпеки за різну абонплату намагаються надавати комплексні набори засобів для розробників і команд безпеки та не обмежуються наданням інструментів SAST, DAST чи то інших. Для розуміння можливої специфіки та відмінностей розглянемо п’ять популярних на сьогодні інструментів безпеки, які дозволяють безоплатно аналізувати відкритий код на виявлення можливих проблем або надають безоплатну тріал-версію, а саме: PMD, Code Climate Quality, Snyk, Codacy та Veracode Static Analysis.

PMD (https://pmd.github.io/) – розширюваний багатомовний статичний аналізатор відкритого коду, який підтримує загалом 16 мов, але переважно стосується Java та Apex. PMD доступний для Windows, macOS і Linux. На відміну від більшості інших аналізаторів коду, які вимагають платної ліцензії або пропонують обмежену функціональність у своїх безплатних планах, PMD є програмним забезпеченням з відкритим вихідним кодом, що робить його економічно ефективною, безоплатною альтернативою платним варіантам. PMD виявляє типові недоліки програмування, як-от: невикористані змінні, порожні блоки catch, створення непотрібних об’єктів тощо. Він має 400+ вбудованих правил, які можна розширювати. Вбудовані перевірки дозволяють налаштовувати правила для різних мов, щоб забезпечити дотримання стандартів кодування. Інструмент також містить детектор копіювання/вставлення (CPD), який допомагає ідентифікувати повторюваний код у кодовій базі. Інтеграція через плагіни доступна з такими популярними IDE, як Eclipse, JDeveloper і Gradle. Щодо недоліків PMD, то налаштування правил аналізу може бути складним, а зворотний зв'язок щодо кодування не надається в режимі реального часу.

Code Climate Quality (https://codeclimate.com/quality) – є інструментом аналізу коду, що здатен забезпечити статичний аналіз для таких мов: PHP, Java, JavaScript, Python і Ruby. Code Climate Quality має вбудовану інтеграцію з GitHub і GitLab. Інструмент також інтегрується з такими системами обміну заявками та повідомленнями, як Asana, Trello та Slack. Особливістю, яка відрізняє Code Climate Quality, є наявна 10-бальна оцінка (від A до F) коду на основі його ремонтопридатності та покриття тестами. Він також оцінює, скільки часу може знадобитися для розв'язання проблеми. Ці показники та візуальні звіти про прогрес допомагають краще визначити пріоритети проблем з обслуговуванням або недостатнім покриттям, Таким чином допомагають командам розробників створювати кращий код. Пропонується безоплатна версія для проєктів з відкритим кодом.

Snyk (https://snyk.io/) – платформа безпеки для розробників, яка пропонує сканування та аналіз коду в реальному часі. Snyk підтримує такі мови для аналізу коду: Python, Java, JavaScript, TypeScript, C++, Kotlin, Ruby, Go, Apex, .NET, PHP, Scala, Swift і VB .NET. Він пропонує інтеграцію зі сховищем Git, що дозволяє визначати пріоритетність проблем у проєктах. Доступна інтеграція для таких інструментів CI/CD: Jenkins, Azure Pipelines і Bitbucket Pipelines. Є також плагіни для інструментів IDE: Eclipse, JetBrains Visual Studio і PhpStorm. Зручною функцією Snyk є надання кожній проблемі оцінки ризику, тож можна визначити пріоритетність проблем і зробити свій код більш безпечним. Інструменти аналізу коду Snyk включають сканування контейнерів, яке перевіряє наявність вразливостей у зображеннях контейнерів, і відстеження коду в реальному часі, яке перевіряє код під час роботи. Його інструмент DeepCode AI відкриває список швидких виправлень, які можна переглянути та застосувати у своєму інтегрованому середовищі розробки (IDE). Хоча повний функціонал Snyk є платним (25$ на місяць), доступний безоплатний план з обмеженням у 100 тестів на місяць.

Codacy (https://www.codacy.com/) – інструмент перевірки коду, який автоматизовано аналізує вихідний програмний код і висвітлює виявлені

проблеми, дозволяючи розробляти більш ефективне програмне забезпечення. Платформа підтримує понад 40 мов програмування та фреймворків, добре інтегрується з робочими процесами CI і DevOps. Проста інтеграція Codacy з GitHub, GitLab, Bitbucket, Jira та Slack дозволяє отримувати миттєві відгуки про код, стандартизувати якість коду, автоматично блокуючи невідповідні запити, тобто вирішувати будь-які виявлені негаразди з кодом. В Codacy є зручна можливість, крім наявних сотень доступних правил, встановлювати ще й власні набори правил. Передбачена 14-денна безоплатна пробна версія. До мінусів можна віднести відсутність інтеграції з Lombok, бібліотекою Java, яка скорочує шаблонний код, та неможливість експортувати шаблони коду.

Veracode Static Analysis (https://www.veracode.com/) – платформа статичного тестування безпеки, яка підтримує понад 100 мов програмування і фреймворків, забезпечуючи широке покриття. Можлива інтеграція з понад 40 платформами, серед яких: Azure DevOps, Bitbucket, Eclipse, Jenkins, Visual Studio тощо. Veracode забезпечує зворотний зв’язок у режимі реального часу, сканує та визначає вразливості протягом усього циклу розробки, пропонує конвеєрні інтеграції CI/CD. Рекомендації щодо виправлення в режимі реального часу допомагають визначити пріоритетність виправлень, які становлять найбільшу загрозу. Особливостями Veracode Static Analysis є його швидка продуктивність сканування та низький рівень хибнопозитивних результатів (<1.1%) [80]. Доступна 14-денна безоплатна пробна версія Veracode. Щодо можливих недоліків, то в деяких областях бракує документації, крім того, характерною є крута крива навчання.

2) Інструменти DAST

На відміну від статичного тестування, інструменти динамічного тестування безпеки застосунків (Dynamic Application Security Testing, DAST) шукають уразливості в застосунках, які вже працюють, тобто в розгорнутій робочій версії. DAST виконує тестування на проникнення, тобто імітує атаки для виявлення вразливостей у вебзастосунку під час його роботи. Інструменти

DAST призначені для автоматизації безпеки програм. Вони надсилають сповіщення групі безпеки для негайного виправлення.

DAST використовує методологію тестування чорної скриньки (Black Box), тобто проводить оцінку вразливості ззовні, оскільки не має доступу до вихідного коду програми. Інструменти DAST не мають уявлення про внутрішні процеси програми.

Проаналізуємо функціонал деяких популярних з десятків доступних на ринку інструментів тестування безпеки, які надають послуги DAST, а саме: ZAP (Zed Attack Proxy), Aikido та Intruder [81, 82].

OWASP ZAP (Zed Attack Proxy) (https://www.zaproxy.org/) є інструментом DAST із відкритим вихідним кодом, який активно підтримується спільнотою. Він надає доступ до великої кількості функцій і плагінів, створених великою спільнотою експертів і розробників з кібербезпеки. Це робить його ідеальним вибором для ентузіастів, які використовують технологію з відкритим кодом і віддають перевагу спільному аспекту розробки рішень. ZAP надає автоматичні сканери, утиліти ручного тестування та традиційні функції проксі. Його можливості налаштування, створення сценаріїв дозволяють задовольнити різноманітні потреби користувачів. ZAP добре інтегрується з іншими інструментами розробки програмного забезпечення та безпеки, забезпечуючи сумісність з інструментами Jenkins для CI/CD та різними системами відстеження помилок. ZAP має інсталятори для Windows, Linux і macOS. ZAP пропонує гнучкість і адаптивність продукту, який постійно розвивається разом із вхідними даними та потребами завдяки активній участі та співпраці користувачів. ZAP є безоплатним інструментом із відкритим вихідним кодом, але новим користувачам може знадобитися тривале навчання та додаткова підтримка з налаштування. Щодо недоліків, то продуктивність ZAP може бути не настільки оптимізованою, як у комерційних інструментах.

Aikido (https://www.aikido.dev/) – платформа поверхневого моніторингу, яка динамічно перевіряє загальні вразливості у інтерфейсі вебзастосунку, не

знижуючи продуктивності та не порушуючи жодних функцій інтерфейсу. Цікаво, що вона виконує сканування вразливостей у тимчасових середовищах, які видаляються після завершення сканування. Крім того, Aikido вимагає доступу лише для читання даних, тому не може редагувати вихідний код. Автентифіковані DAST-перевірки оцінюють, які можливості отримує користувач після входу в систему, зокрема чи має він доступ до конфіденційних даних вебзастосунку. Aikido автоматично сканує програму один раз на день, видаляє хибні спрацьовування, видаляє дублікати та визначає пріоритетність сповіщень на основі серйозності й контексту та сповіщає адміністраторів про будь-які вразливості відповідно до спеціальних правил попередження. Aikido сумісна з усіма основними постачальниками засобів контролю версій (GitHub, GitLab, Bitbucket, Asure DevOps), мовами (JS, Python, Java, Go, Ruby, Rust, PHP, .NET) та хмарними постачальниками (AWS, GCP, Asure) з плавним розгортанням у наявних режимах безпеки. Aikido є комплексною платформою безпеки та, окрім DAST, пропонує сканування залежностей із відкритим вихідним кодом, SAST, керування секретами, інфраструктуру як сканування коду, керування безпекою в хмарі, моніторинг часу виконання наприкінці терміну служби, сканування контейнерів і сканування ліцензій. Щодо ціноутворення, то є безоплатна версія з певним обмеженням функціоналу в один хмарний обліковий запис і 10 репозиторіїв.

Intruder (https://www.intruder.io/) – проактивна DAST-платформа моніторингу безпеки вебзастосунків. Вона забезпечує сканування вразливостей і керування ними, моніторинг поверхонь атак, тестування на проникнення та полегшене виправлення виявлених вразливостей. Intruder інтегрується у процес DevOps і безперервно сканує цифрові активи на виявлення вразливостей і загроз, забезпечуючи чітку видимість поверхні онлайн-атаки. Сканування вразливостей охоплює мережеву інфраструктуру, вебпрограми та API, не вимагаючи жодних змін в інфраструктурі. Надійна система сповіщень відфільтровує нерелевантні сповіщення. Є 14-денна безоплатна версія.

Загалом інструменти DAST є цінними для розробників, груп безпеки та системних адміністраторів, які прагнуть забезпечити надійність своїх програм і захистити їх від потенційних загроз кібербезпеки.

3) Інструменти IAST

В інтерактивних інструментах безпеки (Interactive Application Security Testing, IAST) усунуто всі обмеження, які є в SAST і DAST. Інструменти IAST виявляють проблеми та повідомляють про них в режимі реального часу під час виконання програми. Агенти IAST зазвичай розгортаються на серверах застосунків.

IAST використовує методологію тестування сірої скриньки, може визначити рядок коду, який спричиняє проблеми з безпекою, та швидко повідомити розробника про це для негайного усунення. Наприклад, якщо дані надходять від користувача і він хоче виконати SQL-ін’єкцію в програмі, додавши SQL-ін’єкцію до запиту, запит буде позначений як небезпечний.

Інструменти IAST легко інтегруються у фреймворки для аналізу вихідного коду і контролю виконання програми. Їх можна легко інтегрувати в конвеєр CI/CD команди DevOps. Проводячи IAST, команди DevOps можуть виявити та виправити будь-які вразливості до того, як вебзастосунок вийде на ринок. Це означає, що виправити такі вразливості набагато легше та дешевше. Це також гарантує безпеку програми до того, як хтось її фактично розгорне, допомагаючи запобігти тому, щоб майбутні користувачі програми стали жертвами потенційного витоку даних.

З іншого боку, IAST залежить від мови програмування, тому, якщо організація використовує не дуже популярну технологію розробки, вона може бути несумісною з інструментом IAST. Суттєво, що інструменти IAST сканують код під час його виконання. Тому рекомендовано розгортати інструмент IAST у середовищі контролю якості, яке запускає автоматизовані функціональні тести. Це може допомогти уникнути людської помилки та переконатися, що увесь функціонал програми перевірено [83].

Розглянемо декілька популярних інструментів IAST: Synopsys Seeker, Invicti, Checkmarx та Acunetix.

Synopsys Seeker (https://www.synopsys.com/) – інструмент IAST з активною перевіркою та відстеженням конфіденційних даних для вебзастосунків. Synopsys допомагає командам розробників, QA, DevOps і безпеки автоматизувати тестування безпеки вебзастосунків і служб. Він визначає вразливі місця застосунків шляхом активного моніторингу їх взаємодії. Цей динамічний підхід до безпеки робить його особливо корисним для організацій, яким потрібні ефективні та ґрунтовні рішення IAST. Synopsys Seeker здатен виявляти вразливості в режимі реального часу під час роботи програми. Він інтегрується з інструментами Jira для легкого відстеження помилок і платформами CI/CD для більш плавного робочого процесу розробки. З іншого боку, ціни Synopsys Seeker не є загальнодоступними, їх можна отримати за запитом.

Invicti (https://www.invicti.com/) – масштабована платформа, яка пропонує свій підхід до інтерактивного та динамічного (DAST + IAST) автоматизованого сканування безпеки. Invicti (раніше Netsparker) автоматизовано виявляє та оцінює вразливості, що допомагає розставляти пріоритети у роботі з усунення проблем. Панелі моніторингу Invicti надають інформацію у зрозумілій та стислій формі, навіть якщо у користувача є велика кількість вебресурсів. Invicti використовує масштабовані агенти сканування, які звітують перед основним застосунком та ефективно використовують кілька IT-ресурсів для скорочення часу сканування. Звіти Invicti можуть бути налаштовані відповідно до вимог користувача. Invicti можна інтегрувати з наявними системами Azure DevOps, GitLab, GitHub, Jira, а також системами керування вразливістю Metasploit або Kenna. Також можна інтегрувати Invicti з платформами CI/CD: Jenkins, TeamCity або Bamboo. Цінова політика на офіційному сайті не публікується і надається за запитом, тріал-версії не передбачено.

Acunetix (https://www.acunetix.com/) – платформа сканування вразливостей у програмах, написаних на Node.js, PHP, Java (включаючи Spring framework) і ASP.NET. Acunetix поєднує сканування DAST + IAST і забезпечує перевірку широкого спектра вразливостей: топ-10 OWASP і топ-10 API, SQL-ін'єкції, XSS, неправильні конфігурації, викриті бази даних, позадіапазонні вразливості тощо. Acunetix IAST з AcuSensor сканує кожен файл, зокрема приховані та незв’язані, забезпечуючи користувачам кращу видимість серверної частини їхніх вебзастосунків. Acunetix також може імпортувати файли визначення API та посилання для тестування API за допомогою архітектури REST, SOAP або GraphQL.

Checkmarx (https://checkmarx.com/) – корпоративна хмарна платформа безпеки вебзастосунків, яка допомагає організаціям виявляти, керувати та пом’якшувати вразливі місця веббезпеки. Підходить для статичного та інтерактивного тестування безпеки застосунків. Checkmarx IAST сумісний із застосунками на основі мікросервісів і забезпечує зворотний зв’язок у реальному часі та комплексний аналіз спеціального коду, бібліотек, фреймворків і потоку даних під час виконання. Checkmarx приділяє значну увагу безпеці API, забезпечуючи охоплення топ-10 уразливостей API OWASP. Він виявляє, класифікує та документує API, а також відстежує їх використання та авторизацію. Розроблений з урахуванням комфорту розробників, Checkmarx пропонує детальний аналіз вихідного коду для швидкого усунення потенційних вразливостей. Checkmarx має автоматизоване сканування SCA та відображення вразливостей сторонніх розробників під час сканування IAST. Checkmarx IAST можна розгорнути як локально в приватному центрі обробки даних, так і розмістити в приватному клієнті в AWS, пропонуючи гнучкі варіанти розгортання. Платформа має широкі можливості налаштування, що дозволяє створювати та налаштовувати користувацькі запити для оптимізації результатів, а її повна інтеграція в наявні робочі процеси забезпечує безпечний процес розробки без збоїв.

4) Інструменти RASP та WAF

RASP (Runtime Application Self-Protection) є комплексним рішенням самозахисту програми під час виконання, яке відстежує поведінку програми на наявність уразливостей і активних загроз, а потім автоматично усуває ці загрози в режимі реального часу. Оскільки інструменти RASP вбудовані в середовище виконання програми або підключені до нього, вони мають доступ до архітектури програми та процесу виконання під час виконання, а також можуть отримати доступ до всіх внутрішніх даних програми, включно з логікою, конфігураціями та потоками даних подій. Інструменти RASP використовують ці дані в поєднанні з моніторингом поведінки для виявлення вразливостей і аномальної поведінки, які можуть вказувати на активну загрозу. Якщо інструмент RASP виявляє зловмисну активність, він сповіщає групу безпеки та автоматично блокує доступ, запобігаючи виконанню. Це те, що дає назву інструментам RASP: вони дозволяють програмам «самозахищатися» без втручання людини [84]. З іншого боку, технології RASP нерідко стикаються з проблемами інтеграції та продуктивності [79].

Оскільки RASP використовується після випуску продукту, це робить його більш орієнтованим на безпеку інструментом, як порівняти з іншими, відомими для тестування [75]. Традиційно організації покладалися на SAST, DAST та інструменти аналізу складу ПЗ для захисту своїх програм. Однак завдяки неперервному моніторингу в режимі реального часу в поєднанні з низьким рівнем помилкових спрацьовувань і автоматизованим виправленням, навіть загроз нульового дня, інструменти RASP стають дедалі важливішим гравцем у просторі AppSec. Розробники все частіше використовують програмне забезпечення RASP для виявлення вразливостей у своїх програмах під час виробництва, а організації використовують їх для блокування спроб використання наявних уразливостей у розгорнутих програмах.

Більшість наявних на ринку інструментів безпеки, які надають послуги RASP, також мають інструменти WAF. Тому в цьому підрозділі приклади таких рішень розглянуто разом. RASP та WAF – це два взаємодоповнюючі

рішення для безпеки програм. У той час як WAF забезпечує першу лінію захисту, фільтруючи загрози ще до того, як вони досягнуть вебзастосунків, RASP використовує контекст, наданий глибокою видимістю цих програм, щоб ідентифікувати та блокувати атаки, які прослизають через брандмауер до вебзастосунків. Ця комбінація мінімізує вплив атак, які легко виявити, а також забезпечує захист від складних загроз.

Ключова відмінність RASP від WAF (Web-Application Firewall) полягає в тому, що RASP працює всередині програми та разом з нею. WAF можна обійти або спробувати закодувати запит так, щоб WAF пропустив його до програми. Якщо зловмисник внесе зміни у код програми, WAF не зможе визначити це, оскільки не зрозуміє контекст виконання, на відміну від RASP, що має доступ до даних, отриманих від програми, аналіз яких дозволяє виявляти аномалії та блокувати атаки. Додаткові модулі можуть робити RASP-рішення корисними не тільки для команд безпеки, але й для розробників, адміністраторів, аналітиків та інших підрозділів.

Розглянемо декілька сучасних комплексних рішень RASP і WAF: Aikido Firewall, Imperva та Dynatrace.

Aikido Firewall (https://www.webfirewall.dev/) – повністю вбудований агент RASP з відкритим вихідним кодом. Aikido забезпечує надійну безпеку під час виконання програм Node.js. Цей брандмауер автоматично блокує критичні ін’єкційні атаки, запроваджує обмеження швидкості для API та відстежує вихідний трафік. Він легко інтегрується у наявний стек технологій і підтримує широкий спектр баз даних: MySQL, MongoDB та PostgreSQL. Як вже зазначалося, Aikido є комплексною платформою безпеки, яка надає інструменти SAST, DAST, CSPM, SCA та багато інших. Щодо ціноутворення саме брандмауера Aikido, то є безоплатна 30-денна версія з певним обмеженням функціоналу для одного вебзастосунку.

Imperva RASP (https://www.imperva.com/) – рішення безпеки вебпрограм та хмарних застосунків, зосереджене на захисті програм зсередини. Завдяки інтеграції в середовище виконання, Imperva RASP

ефективно виявляє та запобігає кібератакам у режимі реального часу, включаючи атаки нульового дня та топ-10 уразливостей OWASP. Використовуючи запатентовані методи та проактивний підхід, Imperva RASP дозволяє підприємствам уникати операційних витрат, пов’язаних із виправленням нульового дня. Іншою важливою особливістю Imperva RASP є підтримка хмарних програм. Забезпечуючи захист зсередини, Imperva RASP гарантує, що він завжди синхронізований із програмою, незалежно від будь-яких трансформацій чи змін робочого навантаження. Imperva підтримує: Java, .NET, Nodejs, Oracle, PostgreSQL, MySQL, SQL Server, IBM DB2, IBM Radar, Elastic тощо, і працює для всіх типів програм, включно з API, застарілими, контейнерними. **Imperva Web Application Firewall** – WAF-інструмент, який виявляє та запобігає кіберзагрозам, забезпечуючи безперебійну та стабільну роботу програмного продукту. Imperva WAF є ключовим компонентом комплексного стека Web Application and API Protection (WAAP). Запатентована компанією Imperva технологія динамічного профілювання (Dynamic Profiling) дозволяє автоматизувати процес навчання поведінки користувачів та застосунків шляхом створення профілів застосунків і побудови «базового рівня» або «білого списку» допустимої поведінки користувачів. Також система автоматично навчається тому, що час від часу програма змінюється. Динамічний профіль дозволяє виключити ручне конфігурування та оновлення численних атрибутів програми: URL, параметрів, куків і методів. Брандмауер Imperva ідентифікує і реагує на загрози в трафіку: SQL-ін'єкції, міжсайтовий скриптинг та віддалену вставку файлів, атаки на бізнес-логіку (збір даних із сайту та спам-коментарі), ботнети та атаки DDoS, а також спроби заволодіти обліковим записом до того моменту, коли шахрайство може бути реалізовано.

Dynatrace Application Security (https://www.dynatrace.com/) є комплексним рішенням для захисту хмарних застосунків під час роботи, а також інтелектуальної автоматизації. Дозволяючи клієнтам захищати свої програми у виробництві, Dynatrace Application Security допомагає

оптимізувати процеси DevSecOps, дозволяючи їм швидше впроваджувати інновації, одночасно знижуючи ризики. Dynatrace використовує визначення пріоритетів за допомогою штучного інтелекту через свого радника з безпеки Davis, який надає командам точну інформацію для першочергового усунення критичних уразливостей. Платформа виявляє та блокує поширені атаки на вразливості прикладного рівня, SQLi, атаки JNDI. Dynatrace отримує метрики для кількох попередньо вибраних просторів імен, включно з AWS WAF. Функція аналітики безпеки Dynatrace дозволяє користувачам зменшити витрати на дослідження журналів, пов'язаних з інцидентами безпеки, і вживати профілактичних заходів для посилення захисту. Платформа сприяє швидкому розслідуванню інцидентів безпеки застосунків, пропонуючи причинно-наслідкові дані для контекстного аналізу великих обсягів даних безпеки застосунків.

Не існує ідеального універсального інструмента тестування безпеки. Усі інструменти тестування безпеки мають свої переваги й недоліки через специфіку їх організації. Комбінування та використання переваг кожного чи то більшості з них може забезпечити високий рівень безпеки програмного вебпродукту [82]. Загалом 51% організацій планують збільшити інвестиції в безпеку внаслідок злому, включно з плануванням та тестуванням реагування на інциденти, навчанням співробітників, а також засобами виявлення загроз і реагування [85].

Розробка автоматизованих інструментів завжди була проблемою тестування вебзастосунків на безпеку. Створити автоматизовані інструменти для тестування безпеки важче, ніж для тестування функціональності вебзастосунку [8, 86]. Традиційні інструменти не встигають за темпами екосистеми програмного забезпечення. Генерація тестів безпеки займає багато часу, є схильною до помилок і трудомісткою, тому бажано повністю або частково автоматизувати процеси тестування безпеки [87, 88]. Проблеми, з якими стикаються автоматизовані інструменти для тестування веббезпеки, полягають у тому, що вони повинні йти в ногу з технологіями, які стрімко

розвиваються та змінюються (наприклад, RIA), і належним чином інтегруватися в наявні робочі процеси розробки.

Перевагу автоматизованого тестування на проникнення вебзастосунків важко переоцінити, оскільки сканер безпеки вебзастосунків не тільки скорочує час, вартість і ресурси, необхідні для тестування безпеки вебзастосунків, а й вивільняє час тестувальників безпеки. Проте сканери мають свої недоліки та обмеження, які пов’язані з низьким охопленням тестування і неточними результатами. Сканери безпеки не сканують вебпрограми повністю, а інколи пропускають уразливості й генерують неправильні результати тестування. Дослідження кількісного підрахунку результатів сканерів безпеки вебзастосунків [71] виявили відсутність чітко визначеного методу чи критеріїв для оцінки якості та ефективності їх результатів. Нині популярними автоматизованими інструментами тестування на проникнення є: OpenVAS, Wireshark, Burp Suite, Invicti (раніше Netsparker), Astra Pentest, Acunetix, Intruder, Indusface, AppKnox, Veracode, Detectify, ZAP, Wfuzz, Wapiti тощо [89]. Вибір сканера для автоматизованого тестування на проникнення залежить від багатьох чинників: бюджету, який може виділити організація на тестування, вартості та функцій сканера, вимог та потреб у такому тестуванні вебзастосунку, відповідності різноманітним важливим стандартам (PCI-DSS, HIPAA, GDPR, ISO 27001), наявності підтримки обслуговування, деталізації звітів, наявності допомоги у виправленні й відновленні після виявлення уразливостей тощо. До речі, є безоплатні інструменти тестування безпеки застосунків, наприклад: ZAP, Wfuzz і Wapiti. Практики використовують різноманітні набори методологій, тестових стендів, сканерів безпеки вебзастосунків і вимірювальних показників для кількісної оцінки сканерів безпеки вебзастосунків. Проте показники вимірювання поверхневого покриття та кількості посилань надто неоднозначні, щоб через них визначати тестове покриття того чи іншого сканера безпеки, оскільки сучасні вебзастосунки складаються з численних вебелементів, які є критичними для оцінки уразливості [7].

Поруч із застосуванням різних автоматизованих інструментів не втрачає актуальність і ручне тестування безпеки вебзастосунків. Сама по собі автоматизація не в змозі гарантувати ретельне тестування програми з позиції безпеки. Так, деякі типи вразливостей автоматизовані тести можуть пропустити, наприклад, помилки авторизації, атаки сліпих SQL-ін'єкцій, логічні недоліки, міжсайтові сценарії, вразливості контролю доступу [71]. Їх можна виявити лише за допомогою ручного тестування та спостережень, інтуїції, досвіду й інтелекту тестувальника безпеки. І хоча ручне тестування потребує багато часу у порівнянні з автоматизованим і не завжди є точним через людські помилки, а отже, і менш надійним, саме ручне тестування є нині найпоширенішим методом тестування безпеки. Так, жоден автоматичний сканер не може перевірити всі вразливості OWASP – тут на допомогу приходить ручне тестування. Під час такого тестування Pentester керується не лише власним досвідом, а й використовує відповідні інструменти, серед яких популярними нині є Nmap, Burp Suite, Metasploit, Nessus, Astra Pentest, WireShark, Nikto, Intruder, W3AF, SQLmap, Zed Attack Proxy, Acunetix, Aircrack тощо [197, 81, 90, 91]. Наприклад, безоплатний Nmap використовується для сканування великих мереж і допомагає перевіряти хости, служби та виявляти вторгнення. Проксі-сервер Burp Suite допомагає перехоплювати та змінювати запити, надіслані на сервер, що дозволяє імітувати атаки й збирати інформацію про їхню ціль. Платформу Metasploit тестувальники безпеки використовують для перевірки безпеки мережі або злому віддаленого комп'ютера шляхом розробки та виконання коду експлойту на віддаленій цільовій машині.

Не існує універсального засобу, методу чи інструменту тестування безпеки, який би підходив усім вебзастосункам. Ручне тестування безпеки є трудомістким процесом і вимагає розуміння програми для виконання тесту. Pentester використовує свої особисті навички та досвід, щоб виявити вразливості програми [10, 27]. Автоматизовані інструменти можуть бути менш дорогими, і їх можна використовувати для тестування більшої кількості цілей. Важливо використовувати різні ефективні інструменти автоматизованого

тестування безпеки, а також проводити ретельне ручне тестування, щоб максимально забезпечити виявлення та усунення уразливостей, експлойтів і слабких місць у роботі вебзастосунку. Сканери уразливостей, аналізатори коду та аналізатори компонування ПЗ є автоматизованими інструментами, тоді як фреймворки атак і засоби зламування паролів є ручними інструментами тестувальника безпеки. При виборі методів і засобів тестування безпеки варто враховувати різні чинники: вартість інструмента, можливості автоматизації, функції звітування інструмента і надання доказів уразливостей, які допоможуть вжити правильних заходів і зведуть до мінімуму помилкові спрацьовування. Одні інструменти добре знаходять недоліки безпеки, інші мають гарні можливості звітування, є прості у використанні, а деякі пропонують багатий набір функцій [7]. Не просто знайти доречний інструмент тестування безпеки програм для того чи іншого середовища і того чи іншого вебзастосунку. Кожне ПЗ має свої унікальні функції. Отже, при такому пошуку і виборі варто дотримуватись правильного балансу між швидкістю, точністю, покриттям і вартістю.

1.4. Фішингові атаки у вебзастосунках

Фішингові атаки є однією з найпоширеніших та водночас небезпечних загроз сучасних вебзастосунків. Вони спрямовані на введення користувачів в оману з метою викрадення конфіденційних даних – облікових записів, паролів, фінансової інформації або іншої критичної інформації [94…102]. З точки зору кібербезпеки вебзастосунків, фішинг може реалізовуватись як через клієнтську частину (наприклад, підроблені форми або посилання), так і через серверну інфраструктуру – шляхом підробки вебсторінок, модифікації контенту або організації редиректів на зловмисні ресурси [94, 95].

У науковій та практичній літературі виділяють кілька ключових аспектів фішингових атак у контексті вебзастосунків [96]:

1. Технології підробки сторінок і доменів – використання схожих URL, доменів другого рівня, підстановок символів Unicode (IDN homograph attacks), що дозволяє зловмисникам створювати сторінки, практично ідентичні легітимним ресурсам.

2. Використання серверної частини для атак – наприклад, через компрометацію вебсервера або CMS, ін’єкції шкідливого коду, редиректи на фішингові ресурси, а також підробку SSL-сертифікатів.

3. Соціальна інженерія та цільові атаки – фішинг часто поєднується з методами соціальної інженерії, що підвищує ефективність атак, оскільки користувачі можуть не підозрювати, що взаємодіють із зловмисним ресурсом.

Сучасні системи захисту від фішингу в вебзастосунках забезпечують автоматизоване виявлення підроблених сторінок, перевірку URL та SSL-сертифікатів, а також аналітику поведінки користувачів [97, 98]. Водночас традиційні підходи часто зосереджені на клієнтській частині або на зовнішніх фільтрах, тоді як інтеграція методів захисту безпосередньо у серверну частину вебзастосунку дозволяє більш ефективно блокувати атаки ще до того, як користувач отримає доступ до шкідливого контенту [7, 99].

Таким чином, фішингові загрози демонструють високу актуальність розробки цілеспрямованих методів захисту вебзастосунків від підроблених сторінок, оскільки традиційні засоби безпеки, орієнтовані на клієнтську частину або зовнішні фільтри, не завжди здатні своєчасно виявляти та блокувати шкідливий контент. Для вебзастосунків проблема ускладнюється тим, що фішингові сторінки можуть бути створені всередині серверної інфраструктури, використовувати компрометацію компонентів, редиректи або підроблені сертифікати, що робить атаки малопомітними для кінцевого користувача [23, 100].

Це підкреслює потребу розробки методів автоматичного визначення легітимності вебсторінки, які можуть інтегруватися безпосередньо у серверну частину застосунку. Такі методи повинні не лише аналізувати URL та структурні характеристики сторінок, але й враховувати поведінку

користувачів, вміст контенту, взаємодію з бекендом та інші динамічні ознаки, що дозволяє виявляти тонкі сигнали фішингової активності.

Сучасні підходи до захисту вебзастосунків від фішингу можна класифікувати за кількома напрямами [103…109]:

1. Статичні методи аналізу – перевірка URL, доменних імен, SSL-сертифікатів, метаданих сторінки. Дозволяють відокремлювати відомі фішингові ресурси від легітимних, проте мають обмежену ефективність щодо нових або модифікованих атак [52].

2. Динамічні методи моніторингу та аналізу поведінки – аналіз взаємодії користувача зі сторінкою, скриптів, форм та редиректів; відстеження нетипової поведінки, наприклад, автоматизованих запитів або підозрілих послідовностей переходів [54].

3. Системи на основі правил та сигнатур – використовують заздалегідь визначені шаблони фішингових сторінок, аналітику мережевого трафіку та сигнатури відомих атак. Ефективні для повторюваних атак, але не здатні адаптуватися до нових схем.

4. Методи на основі машинного навчання та штучного інтелекту – нейромережі, класифікація DOM-структури, NLP-аналіз текстового контенту, детекція аномалій у поведінці користувачів [100, 101]. Ці підходи дозволяють виявляти нові, ще не зафіксовані фішингові сторінки та інтегруватися безпосередньо у серверну частину вебзастосунку, забезпечуючи адаптивний та швидкий захист у реальному часі.

5. Комбіновані гібридні системи – поєднують кілька підходів: сигнатури, динамічний аналіз та алгоритми ШІ [101, 102], що підвищує точність і знижує кількість хибних спрацьовувань.

Ідеальним інструментом для створення таких систем є штучний інтелект (ШІ) та методи машинного навчання [14, 21], оскільки вони здатні:

– автоматично аналізувати великі обсяги даних та виявляти складні патерни, які неможливо формалізувати вручну;

– інтегрувати різнорідні джерела інформації (структура сторінки, поведінка користувачів, історія запитів, контентні характеристики);

– адаптуватися до нових, ще не зафіксованих методів фішингу завдяки здатності до навчання на прикладах;

– забезпечувати швидке прийняття рішень у реальному часі для блокування підозрілих сторінок без затримки обслуговування користувачів.

Таким чином, поєднання інтегрованих методів захисту на серверному рівні з алгоритмами ШІ дозволяє створити ефективну, адаптивну та масштабовану систему захисту вебзастосунків від фішингових атак, що є критично важливим для підвищення безпеки користувачів і стабільності бізнес-процесів.

1.5. Захист вебзастосунків від прихованих каналів передавання інформації

Сучасні вебзастосунки часто дозволяють користувачам завантажувати мультимедійні файли, такі як зображення, аудіо або відео, для зберігання або обробки на сервері [3, 7]. Однак ця функціональність може бути використана зловмисниками для організації прихованих каналів передачі інформації (covert channels) [110…122], які дозволяють обходити традиційні механізми контролю та захисту.

Приховані канали характеризуються тим, що інформація передається неявним способом, тобто без явних змін у стандартному функціонуванні вебзастосунку або мережевого протоколу [12, 105]. У контексті мультимедійних файлів це може реалізовуватися через:

1. стеганографію у зображеннях – приховане вбудовування бітових патернів у пікселі, які несуть додаткову інформацію без помітної зміни візуального зображення [19, 105];

2. стеганографію в аудіо та відео – змінення частотних компонентів, амплітуди чи фазових характеристик сигналу, яке дозволяє передавати дані непомітно для користувачів і базових систем перевірки [106];

3. метадані та додаткові поля файлів – використання тегів, EXIF-даних, заголовків файлів або структурованих полів для прихованого кодування інформації [107].

Такі приховані канали можуть бути організовані без порушення функціональності вебзастосунку і без видимих ознак для адміністратора або систем безпеки, що робить їх особливо небезпечними. При цьому традиційні методи контролю доступу або антивірусний захист не здатні ефективно виявляти такі канали, оскільки вони маскуються у легітимному контенті [20, 108].

Таким чином, вебзастосунки, що дозволяють завантаження мультимедійних файлів, потребують спеціалізованих методів виявлення та запобігання прихованим каналам, які можуть реалізовуватися як на серверному рівні, так і з використанням сучасних методів стеганоаналізу та алгоритмів штучного інтелекту [16, 109]. Водночас сучасні підходи до стеганоаналізу вебзастосунків не відповідають масштабам та складності сучасних загроз: вони часто орієнтовані на окремі типи файлів або прості стеганографічні методи, не здатні ефективно виявляти складні та адаптивні схеми передачі прихованої інформації, що зловмисники реалізують у реальних системах [110]. Це підкреслює критичну необхідність розробки нових, масштабованих та адаптивних методів детектування, здатних інтегруватися у життєвий цикл вебзастосунків [111] та забезпечувати постійний моніторинг і захист від прихованих каналів передачі даних [112].

Інструментом захисту від створення таких прихованих каналів передавання інформації є стеганоаналіз [113]. Стеганоаналіз спрямований на виявлення фактів наявності прихованих повідомлень у мультимедійних контейнерах (зображеннях, аудіо, відео) та оцінку їх стійкості до атак [114]. Основними завданнями стеганоаналізу є:

1. Визначення факту наявності прихованого повідомлення.

2. Оцінка можливості прочитання або модифікації прихованого повідомлення.

3. Виявлення слабких місць у стеганосистемі, що дозволяє підвищити її безпеку.

На рис. 1.4 наведено класифікацію технік стеганоаналізу, що застосовуються для атак на стеганосистеми [123…126].

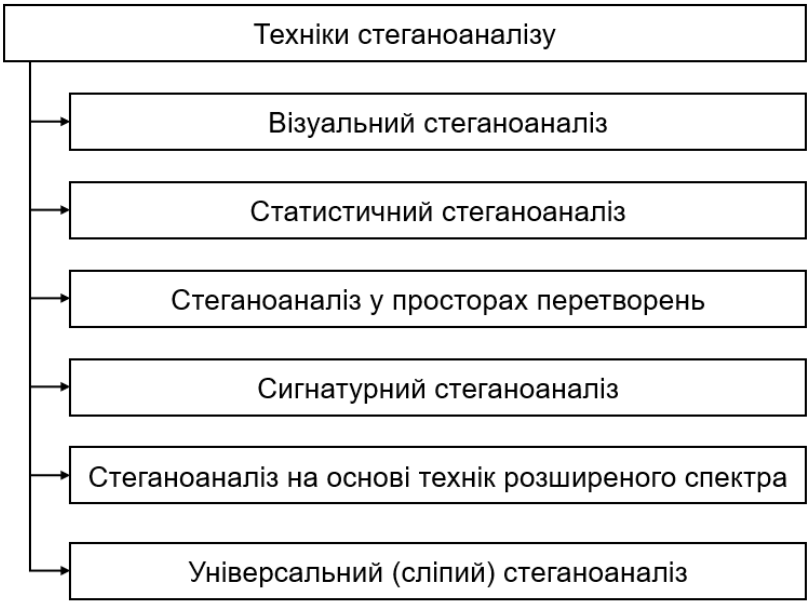


Рисунок 1.4 – Техніки атак на стеганосистеми

Розглянемо докладніше можливі атаки стеганоаналізу:

1. *суб'єктивна атака* – найпростіший вид атаки, при якому атакуючий аналізує контейнер «на око» або «на слух», намагаючись виявити наявність прихованого повідомлення. Використовується проти незахищених стеганосистем і зазвичай є першим етапом практичного дослідження стеганосистеми;

2. *статистичний стеганоаналіз* базується на дослідженні статистичних властивостей сигналу. Наприклад, розподіл молодших бітів (LSB) часто має шумовий характер і використовується для прихованих повідомлень. Виявлення змін статистики сигналу дозволяє аналітику

визначити наявність стеганоканалу. Для непомітного вбудовування стеганокодер розв’язує три завдання:

– виділити підмножину бітів, зміна яких мінімально впливає на якість;

– вибрати необхідну кількість бітів відповідно до розміру прихованого повідомлення;

– змінити їх, зберігаючи статистичні властивості контейнера. Попереднє шифрування або стиснення повідомлення забезпечує подібність характеристик вбудованих даних і шуму;

3. *стеганоаналіз у областях перетворень* – один із найпотужніших методів, який використовує властивості змін сигналу після стеганоперетворення в областях перетворень, як-от:

– дискретне косинусне перетворення (DCT);

– перетворення Уолша-Адамара;

– сингулярне розкладання (SVD).

Ці методи є найбільш точними та ефективними для сучасних мультимедійних сигналів [11, 120];

4. *сигнатурні методи* засновані на синтаксичному аналізі послідовностей термінальних символів у контейнері. Виявлення специфічних сигнатур дозволяє однозначно визначити використану стеганосистему. Перевага методів – точна ідентифікація системи, недолік – малий відсоток стеганопрограм (менше 10%), що залишають у контейнерах сигнатури [121];

5. *методи на основі розширеного спектра* дозволяють виявляти широкосмугові стеганоповідомлення, вбудовані із залученням багатьох частотних компонентів контейнера [122];

6. *сліпий стеганоаналіз* використовує теорію статистичного розпізнавання образів із методами класифікації та машинного навчання. Ці підходи дозволяють будувати автоматизовані системи виявлення прихованих повідомлень навіть без попередньої інформації про стеганосистему [123, 124].

Сучасні методи стеганоаналізу охоплюють широкий спектр технік: від суб’єктивного аналізу до складних алгоритмів на основі перетворень і машинного навчання. Основною метою є своєчасне виявлення стеганоканалів і підвищення стійкості мультимедійних систем до прихованої передачі даних.

Класичний підхід до стеганоаналізу ґрунтується на поетапному виділенні інформативних ознак, що дозволяють виявити наявність прихованого повідомлення. На першому етапі здійснюється формування ознак, яке може виконуватися безпосередньо в просторовій області зображення або ж у різних областях перетворень. Подальший аналіз та класифікація здійснюються вже на основі отриманих ознак. На рис. 1.5 наведена схема класифікації основних областей перетворення, що застосовуються для стеганоаналізу.

Стеганоаналіз у просторовій області [127…131] є одним із найпоширеніших підходів, який поєднує численні методи, серед яких: аналіз пар пікселів, аналіз триплетів, вивчення гістограм, аналіз сусідніх відліків, а також методи на основі локальних контрастів і текстурних ознак. Ці підходи дозволяють виявляти зміни безпосередньо у значеннях пікселів або вибірках сигналу, що робить їх ефективними для деяких стеганосистем [126, 127]. Водночас через потребу аналізувати великі обсяги даних без попередніх перетворень виділення ознак у просторовій області часто є обчислювально затратним і може виявитися повільнішим за методи, які працюють в областях перетворень.

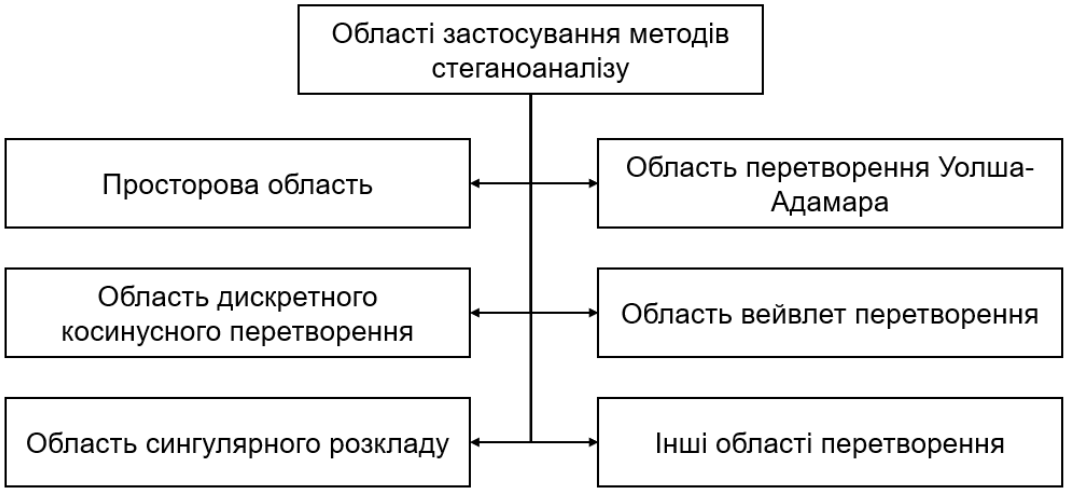


Рисунок 1.5 – Області перетворень, що застосовуються для стеганоаналізу

Дискретне косинусне перетворення [132] є одним із найпоширеніших інструментів цифрової обробки сигналів та зображень. Його популярність зумовлена високою енергетичною компактністю: більша частина енергії сигналу концентрується в невеликій кількості низькочастотних коефіцієнтів. Саме ця властивість робить ДКП базою для стандартів стиснення зображень (JPEG) та відео (MPEG), а також привабливою основою для стеганографічних методів. Використання ДКП дозволяє здійснювати вбудовування інформації у частотну область, що забезпечує кращу стійкість до атак проти методів просторової області. ДКП визначається таким співвідношенням [14]:

$$S = C_N X C_N^T, \tag{1.1}$$

де X – фрагмент вихідного зображення розміру $N \times N$;

C_N – $N \times N$-матриця ДКП, елементи $C(i,j)$, $i,j = 0,1,...,N-1$ якої обчислюються відповідно до формули:

$$C(i,j) = \begin{cases} \dfrac{1}{\sqrt{N}}, & \text{при } i = 0; \\ \sqrt{\dfrac{2}{N}} \cos(2j+1) \cdot i \cdot \pi, & \text{при } i > 0. \end{cases} \tag{1.2}$$

Трансформанти ДКП показують розподіл контенту блоку зображення по частотних складових. При цьому відомо [133], що чутливість стеганоповідомлення до збурювальних впливів залежить від того, які саме частотні складові зазнали збурення в процесі стеганоперетворення. Так, гарантована стійкість стеганографічного методу до атак проти вбудованого повідомлення забезпечується за рахунок збурення низькочастотних складових контейнера. Такі зміни з великою ймовірністю негативно вплинуть на надійність сприйняття стеганоповідомлення, через що на практиці часто використовується «компромісний варіант», коли вбудовування ДІ проводиться так, щоб збурення зазнали середньочастотні складові цифрового контента [132]. Однак, такий підхід забезпечує стійкість стеганографічного методу лише до незначних збурюваних впливів, в той час як в реаліях атаки проти вбудованого повідомлення можуть бути значними (наприклад, стиснення стеганоповідомлення з малим коефіцієнтом якості) [11].

Розподілення частотних складових у матриці трансформант ДКП блока X наведено на рис. 1.6.

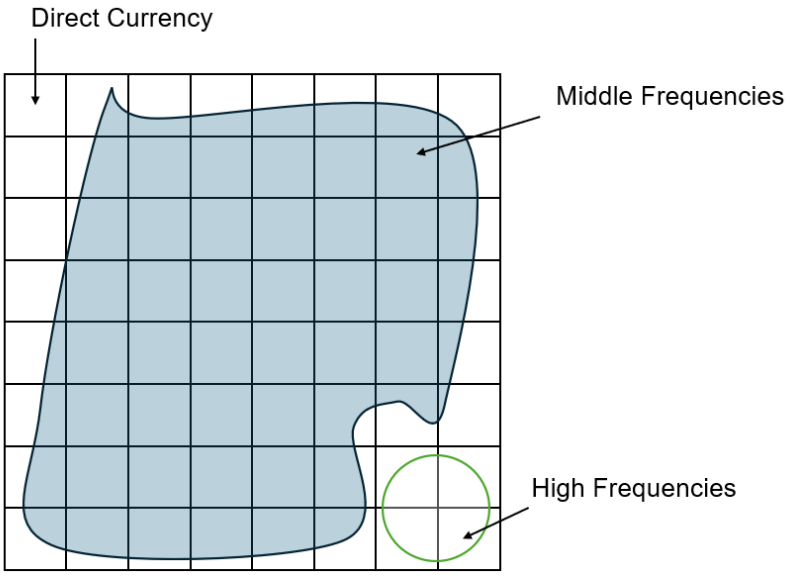


Рисунок 1.6 – Розподілення частотних складових у трансформантах ДКП

З погляду стеганоаналізу, використання ДКП дозволяє локалізувати вбудовану інформацію у різних частотних діапазонах зображення. При цьому саме характер збурень частотних складових є ключовим індикатором наявності прихованих даних [133]. Низькочастотні зміни легко виявляються через помітні артефакти, високочастотні – швидко руйнуються при обробці сигналу, а середньочастотні, що найчастіше використовуються стеганографами як компроміс, утворюють специфічні статистичні відхилення, які й може зафіксувати стеганоаналітик [132…136].

Під сингулярним розкладом матриці-блока X розміру $N \times N$ розуміють його подання у вигляді [137…139]:

$$X = U\Sigma V^T, \tag{1.3}$$

де U, V ортогональні матриці розміру $N \times N$, стовпці $u_1,...,u_8$ матриці U, називають лівими сингулярними векторами, стовпці $v_1,...,v_8$ матриці V називають правими сингулярними векторами матриці X, тоді як $\Sigma = diag(\sigma_1,...,\sigma_8)$ – діагональна матриця сингулярних чисел.

Сингулярне розкладання (SVD) у стеганоаналізі дозволяє подати зображення у вигляді трьох матриць, де сингулярні значення відображають енергетичну структуру контенту [137, 138]. Це дає змогу виявляти приховані повідомлення шляхом аналізу відхилень у розподілі сингулярних значень, оскільки навіть невеликі зміни, внесені стеганографічними методами, можуть змінювати статистичні властивості матриць. Використання SVD забезпечує ефективний інструмент для дослідження впливу стеганоперетворень на мультимедійні дані та підвищує точність виявлення прихованих каналів [139].

Перспективним видом перетворень, що використовується для побудови сучасних стеганографічних методів, є двовимірне перетворення Уолша-Адамара [140, 141], яке задається за допомогою співвідношення:

$$W_X = H'_N X H'^T_N, \tag{1.4}$$

де $H'_N = \frac{1}{\sqrt{N}} H_N$, X – матриця розміру $N \times N$, а матриця Адамара H_N порядка N визначається за допомогою конструкції Сильвестра:

$$H_{2^k} = \begin{bmatrix} H_{2^{k-1}} & H_{2^{k-1}} \\ H_{2^{k-1}} & -H_{2^{k-1}} \end{bmatrix}, \; H_1 = 1. \tag{1.5}$$

Перетворення Уолша-Адамара у стеганоаналізі забезпечує подання сигналу у вигляді ортогональної бази з елементів +1 та -1, що дозволяє виявляти закономірності та аномалії, внесені стеганографічними методами [142]. Воно характеризується дуже низькою обчислювальною складністю, що робить його ефективним для швидкого аналізу великих обсягів мультимедійних даних та точного виявлення стеганоканалів у зображеннях і інших носіях інформації [143].

Дискретне вейвлет-перетворення сигналу $x[n]$ полягає у проєкції сигналу на базисні функції-вейвлети $\psi_{j,k}[n]$, які масштабуються та зсуваються у часі [144…147]:

$$W_{j,k} = \sum x[n] \cdot \psi_{j,k}[n], \tag{1.6}$$

де j – рівень масштабування (частота), а k – позиція зсуву у часі (локалізація).

У контексті стеганоаналізу дискретне вейвлет-перетворення дозволяє виділяти деталі сигналу на різних масштабах, що робить приховані повідомлення більш помітними у відповідних коефіцієнтах вейвлет-перетворення. Це дає змогу ефективно виявляти стеганоповідомлення як у високочастотних трансформантах (деталі), так і у низькочастотних (основна структура сигналу) [148…150].

Окрім дискретного косинусного перетворення, перетворення Уолша-Адамара, вейвлет-перетворення та сингулярного розкладу, у стеганоаналізі застосовуються й інші види перетворень для виявлення прихованих повідомлень [148, 149]. Наприклад, дискретне Фур’є-перетворення (DFT) [151,

152] дозволяє аналізувати фазові та амплітудні характеристики сигналу, що підвищує ймовірність виявлення стеганоканалів, навіть після компресії або модифікації даних. Дискретне Чирп-Зет-перетворення (CZT) [153…156] забезпечує гнучкий аналіз локальних частотних діапазонів, відкриває можливості для детектування адаптивних методів приховування інформації. Перспективними у стеганоаналізі також є сингулярно-хвильові та гібридні перетворення, які поєднують сильні сторони кількох математичних апаратів, дозволяючи підвищувати точність виявлення прихованих повідомлень та робити аналіз більш стійким до різноманітних атак на цифровий контент.

Попри наведені класифікації методів стеганоаналізу, наш аналіз наукових джерел показує, що для сучасних вебзастосунків ключовим критерієм є швидкодія вибраного методу [153, 154]. Це зумовлено тим, що вебзастосунки часто обробляють великі обсяги мультимедійної інформації – зображення, відео, аудіо – і навіть незначні затримки при обробленні даних можуть негативно вплинути на роботу системи та користувацький досвід [2, 155, 156]. Використання методів стеганоаналізу на основі дискретного косинусного перетворення, сингулярного розкладу вимагає значних обчислювальних ресурсів, що суттєво уповільнює обробку і робить їх менш придатними для роботи у складі вебзастосунків у реальному часі [157, 158].

На наш погляд, оптимальним підходом є поєднання нересурсозатратних перетворень із методами ШІ, що дозволяє забезпечити одночасно високу точність та прийнятну швидкодію [15]. Проте на сьогодні ефективних рішень такого типу мало, що визначає актуальність подальших досліджень у сфері швидкого та точного стеганоаналізу для вебзастосунків.

Висновки до розділу 1

1. Безпека є невіддільною частиною життєвого циклу розробки ПЗ. Сучасні підходи DevSecOps інтегрують заходи безпеки на всіх етапах – від планування і кодування до розгортання та експлуатації. Це дозволяє виявляти

вразливості завчасно, знижувати вартість їх усунення та підвищувати загальну стійкість систем. Дихотомічна класифікація безпеки вебзастосунків є доцільною та практичною. Вона чітко розділяє безпеку на дві складові:

– безпека розробки (DevSec) – превентивні заходи, спрямовані на запобігання вразливостям на етапах створення ПЗ (SAST, IAST, Code Review, SCA тощо);

– безпека експлуатації (Runtime Security) – захист під час роботи застосунку, орієнтований на виявлення та нейтралізацію атак у реальному часі (DAST, RASP, WAF, моніторинг).

Безпека розробки (DevSec) є добре вивченою та стандартизованою. Є чіткі методики, інструменти та процеси інтеграції безпеки в CI/CD, що робить цю галузь передбачуваною й ефективною. Безпека експлуатації (Runtime Security) залишається менш дослідженою, особливо в контексті захисту від сучасних складних загроз, серед яких:

– фішингові атаки, які використовують соціальну інженерію, підроблені сторінки та серверні вразливості;

– приховані канали передачі інформації (наприклад, за допомогою стеганографії в мультимедіа), які важко виявити традиційними методами.

Потрібен розвиток адаптивних методів захисту, зокрема на основі ШІ і машинного навчання, для ефективної протидії фішингу та виявлення стеганографічних каналів у реальному часі без значних обчислювальних затрат.

2. Фішингові атаки є критично небезпечними для вебзастосунків, оскільки вони спрямовані на викрадення конфіденційних даних користувачів (логіни, паролі, фінансова інформація) та підривають довіру до сервісу. Вебзастосунки часто є ціллю через їхню доступність, інтерактивність та залежність від взаємодії з користувачем. Інтеграція методів виявлення фішингу безпосередньо у серверну частину вебзастосунку є ключовою з кількох причин:

– серверна частина має доступ до повного контексту запитів, поведінки користувачів, метаданих та логів, що дозволяє аналізувати підозрілу активність у реальному часі;

– блокування шкідливих запитів на серверному рівні запобігає їхньому потраплянню до клієнта, що підвищує безпеку та зменшує навантаження на кінцевого користувача;

– сервер може досліджувати та порівнювати дані з різних джерел (наприклад, трафік, логи, поведінкові метрики), що ускладнює обхід захисту зловмисниками.

Наявні методи виявлення фішингу мають значні недоліки, які обмежують їхню ефективність при інтеграції в сучасні вебзастосунки:

– статичні методи (на основі сигнатур, чорних списків URL) не ефективні проти нових атак, які швидко змінюються;

– клієнтські рішення (наприклад, браузерні розширення) покладаються на обробку даних на стороні користувача, що створює затримки та не захищає від серверної компрометації;

– зовнішні системи (наприклад, API для перевірки URL) можуть уповільнити роботу застосунку через мережеві затримки та залежність від сторонніх сервісів;

– відсутність адаптивності: традиційні підходи не вміють швидко адаптуватися до нових схем фішингу, таких як використання Unicode-доменів (IDN homograph attacks) або динамічних редиректів.

Потрібна розробка нових, адаптивних методів, які:

– можуть бути інтегровані безпосередньо в серверну логіку вебзастосунку, щоб забезпечити швидке та ефективне блокування атак без зовнішніх залежностей;

– використовують машинне навчання та ШІ для аналізу поведінки, контенту і метаданих у реальному часі, що дозволить виявляти невідомі атаки;

– ефективно працюють у реальному часі без значних обчислювальних затрат, щоб не уповільнювати роботу вебзастосунку.

Така інтеграція забезпечить проактивний захист, зменшить кількість хибних спрацьовувань і підвищить загальну стійкість вебзастосунків до сучасних кіберзагроз. Інтеграція засобів захисту на серверному рівні є ключовою для ефективного протидіяння атакам, оскільки традиційні клієнтські або мережеві рішення часто недостатньо ефективні.

3. Стеганоаналіз є критично важливим для захисту вебзастосунків, оскільки вони часто допускають завантаження мультимедійних файлів (зображень, аудіо, відео), які можуть бути використані зловмисниками для створення прихованих каналів передачі інформації. Це становить загрозу конфіденційності, цілісності даних та може призвести до витоку критичної інформації. Наявні методи стеганоаналізу розроблені недостатньо для ефективного застосування у вебсередовищі. Вони часто:

– орієнтовані на окремі типи файлів або прості стеганографічні методи, тоді як сучасні алгоритми стали складнішими та адаптивними;

– вимагають значних обчислювальних ресурсів (наприклад, методи на основі ДКП, сингулярного розкладу або вейвлет-перетворень), що робить їх непридатними для обробки даних у реальному часі за умов вебсервера;

– не в змозі ефективно виявляти складні стеганографічні схеми, такі як адаптивне вбудовування в середньочастотні діапазони, використання метаданих або гібридні методи.

Вебзастосунки потребують швидких та ефективних рішень, оскільки вони оперують великими обсягами мультимедійних даних і навіть незначні затримки в обробленні можуть погіршити користувацький досвід. Серверна частина має аналізувати файли негайно після завантаження, щоб запобігти передачі прихованих повідомлень до того, як вони будуть збережені або оброблені. Потрібна розробка нових, оптимізованих методів стеганоаналізу, які:

– мають низьку обчислювальну складність і можуть працювати в реальному часі без уповільнення роботи вебзастосунку;

– використовують комбінацію простих перетворень (наприклад, перетворення Уолша-Адамара) з методами ШІ, що дозволить досягти високої точності детектування при мінімальних ресурсних витратах;

– адаптуються до нових типів атак і здатні аналізувати різноманітні формати файлів (зображення, аудіо, відео) з високою ефективністю;

– інтегровані безпосередньо в серверну логіку вебзастосунку, що забезпечить автоматичну перевірку завантажених файлів без необхідності використання зовнішніх систем.

Такий підхід дозволить ефективно протидіяти сучасним стеганографічним загрозам, зменшити ризики витоку даних і підвищити загальний рівень безпеки вебзастосунків без погіршення продуктивності.

РОЗДІЛ 2. РОЗРОБЛЕННЯ ТЕОРЕТИЧНОГО БАЗИСУ ЗАХИСТУ ВІД ПРИХОВАНИХ КАНАЛІВ У ВЕБЗАСТОСУНКАХ

2.1. Обґрунтування доцільності виявлення прихованих каналів у вебзастосунках в області перетворення Уолша-Адамара

У сучасному життєвому циклі розробки ПЗ тестування безпеки є критично важливим етапом, спрямованим на виявлення вразливостей, які можуть бути використані для несанкціонованого доступу до даних або порушення роботи систем. Особливої актуальності набуває захист вебзастосунків, які перетворилися на багатофункціональні платформи для зберігання та обміну мультимедійними даними. Така функціональність створює передумови для виникнення прихованих каналів передавання інформації, зокрема за рахунок використання стеганографічних методів. Завантаження зображень, аудіо- та відеофайлів без належного контролю дозволяє зловмисникам приховувати дані у контейнерах і використовувати вебзастосунки як транспортні вузли для витоку конфіденційної інформації або координації атак.

Якщо стеганографічні методи попередніх поколінь були відносно легко виявлюваними класичними методами стеганоаналізу [159], то сучасні підходи, зокрема методи з кодовим управлінням вбудовуванням, а також сучасні взірці стеганографічних методів на основі сингулярного розкладу, демонструють високу стійкість до виявлення [160]. Такі методи дають змогу вибірково впливати на окремі частотні складові контейнера, забезпечуючи стійкість до атак, високу надійність сприйняття, стійкість до атак проти вбудованого повідомлення, а в разі стеганографічних методів з кодовим управлінням – можливість роботи на платформах з обмеженими обчислювальними ресурсами, включно з мобільними пристроями, IoT та БПЛА. Так, дослідження показують, що стеганографічні методи з кодовим управлінням можуть

перевершувати за стійкістю найкращі відомі алгоритми, що працюють в областях перетворень, зокрема у просторі сингулярного розкладу.

Разом з тим сучасні підходи до стеганоаналізу часто ґрунтуються на обчислювально складних методах, таких як сингулярний розклад, дискретне косинусне перетворення та їх комбінації [161, 162]. Хоча ці методи демонструють високу точність виявлення прихованих даних, їхня обчислювальна складність обмежує можливості застосування у вебсередовищі, де потрібна потокова обробка великого обсягу мультимедійних даних у реальному часі. Методи глибокого навчання, які останнім часом широко застосовуються у стеганоаналізі, також вимагають значних обчислювальних ресурсів і спеціалізованого апаратного забезпечення, часто ґрунтуючись на ресурсоємних перетвореннях [163].

Перспективним напрямом розвитку є використання області перетворень Уолша-Адамара, яка характеризується низькою обчислювальною складністю порівняно з іншими ортогональними перетвореннями, зберігаючи при цьому високу інформативність трансформант [164]. Застосування цього апарату дає змогу створювати методи виявлення прихованих каналів з мінімальними обчислювальними витратами, що особливо важливо для масштабованих вебплатформ. Особливу увагу заслуговує завдання виявлення прихованих каналів, побудованих на основі концепції кодового управління та сингулярного розкладу, оскільки вони відзначаються високою стійкістю та здатністю зберігати статистичні характеристики контейнера.

Таким чином, є об’єктивна потреба у розробці теоретичного базису, який би забезпечував ефективне виявлення сучасних стеганографічних методів, включно з тими, які використовують сингулярні вектори у просторі перетворень Уолша-Адамара з урахуванням вимог високої швидкодії та низьких обчислювальних витрат у вебзастосунках.

Тому завданнями цього розділу є:

1. розробка теоретичного базису виявлення прихованих каналів на основі концепції кодового управління у вебзастосунках за рахунок аналізу

трансформант перетворення Уолша-Адамара;

2. розробка теоретичного базису виявлення прихованих каналів у вебзастосунках на основі SVD UV-методів за рахунок аналізу трансформант перетворення Уолша-Адамара.

2.2. Основні поняття та математичний апарат дослідження стеганоаналізу методу з кодовим управлінням

У сучасному життєвому циклі розробки програмного забезпечення тестування безпеки є критично важливим етапом, спрямованим на виявлення вразливостей, які можуть бути використані для несанкціонованого доступу до даних або порушення роботи систем. У цьому сенсі галузь тестування програмного забезпечення є на межі з галуззю кібербезпеки у сенсі важливості виявлення вразливостей та загроз [3, 159].

Для вебзастосунків однією з важливих та малодосліджених на сьогодні загроз є загроза їх використання для передавання прихованої інформації методами стеганографії [14]. При цьому, якщо стеганографічні методи минулих поколінь [161] були такими, що досить легко виявляються класичними методами стеганоаналізу, сучасні стеганографічні методи є досить резистивними до класичних стеганоаналітичних методів.

Одним з таких методів є сучасний метод з кодовим управлінням вбудовуванням додаткової інформації, представлений вітчизняними дослідниками у роботі [162]. Зазначений метод оперує в просторовій області, але при цьому володіє можливістю вибіркового впливу на задані частотні складові контейнера, що дозволяє отримувати задані властивості стеганоповідомлення, а саме: стійкість до атак проти вбудованого повідомлення, високий рівень надійності сприйняття тощо. При цьому робота стеганографічного методу з кодовим управлінням в просторовій області контейнера дозволяє говорити про можливості його застосування на платформах з обмеженими обчислювальними можливостями, як-от: мобільні

пристрої, пристрої IoT, БПЛА – і при достатній пропускній спроможності про забезпечення надійності сприйняття та, як показують сучасні дослідження [163], при стійкості до атак проти вбудованого повідомлення, що перевершує стійкість кращих відомих методів стеганографії, які оперують в областях перетворень, зокрема, в області сингулярного розкладання матриць блоків контейнера.

У роботі [143] проведено дослідження стійкості стеганографічного методу з кодовим управлінням до атак стеганоаналізу, які показали високу стійкість зазначеного методу до відомих стеганоаналітичних атак, зокрема до атаки із застосуванням відомої утиліти StegExpose, яка навіть за ідеальних умов виявилася не здатною виявляти стеганоповідомлення, створені із застосуванням стеганографічного методу з кодовим управлінням. У зазначеній роботі виявлено декілька ознак, які дозволили створити стеганоаналітичний метод виявлення стеганоповідомлень, створених за допомогою методу з кодовим управлінням вбудовуванням додаткової інформації, втім створений метод забезпечує достовірність виявлення на рівні ~80%, при цьому ефективність методу стрімко падає зі зменшенням відсотку блоків, в які виконано вбудовування додаткової інформації, або із застосуванням формату зберігання з втратами, що допускається з огляду на стійкість стеганографічного методу з кодовим управлінням до атак проти вбудованого повідомлення.

Загалом концепція кодового управління вбудовуванням стала основою стеганографічних методів з кодовим управлінням і складає значну небезпеку для сучасних вебзастосунків через можливість створення прихованих каналів передавання інформації стеганографічними засобами. Це потребує як нового теоретичного, так і практичного базису, що включав би ефективні методи виявлення та закриття таких прихованих каналів передавання інформації.

Математичний апарат перетворення Уолша-Адамара є ефективною основою для побудови таких стеганоаналітичних методів через свою високу

обчислювальну ефективність, прозорість фізичного змісту трансформант перетворення Уолша-Адамара.

Так, одним із важливих інструментів для обробки цифрових зображень у контексті стеганографії та стеганоаналізу є двовимірне перетворення Уолша-Адамара (Walsh-Hadamard Transform, WHT) [164]. Це ортогональне перетворення базується на базисних функціях, які приймають значення лише +1 та −1, і дозволяє ефективно подати сигнал у вигляді послідовності коефіцієнтів, які характеризують його частотні компоненти.

Нехай задано блок цифрового зображення X розміру $N \times N$. Тоді перетворення Уолша-Адамара цього блока визначається як

$$W_X = H'_N X H'^T_N, \tag{2.1}$$

де $H'_N = \frac{1}{\sqrt{N}} H_N$, X – матриця розміру $N \times N$, а матриця Адамара H_N порядку N задається за допомогою конструкції Сильвестра

$$H_{2^k} = \begin{bmatrix} H_{2^{k-1}} & H_{2^{k-1}} \\ H_{2^{k-1}} & -H_{2^{k-1}} \end{bmatrix}, \; H_1 = 1. \tag{2.2}$$

Окрім двовимірного перетворення Уолша-Адамара, відомий також його одновимірний варіант, який для вектора Y задається як:

$$V = YH_N, \tag{2.3}$$

при цьому у роботі [165] встановлений взаємозв’язок між двовимірною та одновимірною версіями перетворення Уолша-Адамара, в рамках чого було доведено, що:

$$\tilde{W} = \tilde{X} H_{N^2}, \tag{2.4}$$

де запис $\tilde{W}$ і $\tilde{X}$ означає подання відповідних матриць розміру $N \times N$ у вигляді вектора довжини N^2 шляхом послідовної конкатенації рядків відповідної матриці, тоді як обчислення коефіцієнтів Уолша-Адамара виконується з точністю до нормувального коефіцієнта $1/N$.

Вираз (2.4) став основою концепції кодового управління вбудовуванням додаткової інформації, яка полягає в тому, що вбудовування відбувається за рахунок подання кожного інформаційного біта d_i у вигляді кодового слова T, що вибірково впливає на ту чи іншу трансформанту перетворення Уолша-Адамара, яке адитивно вбудовується у відповідний блок контейнера:

$$\tilde{M} = \tilde{X} + \tilde{T}, \tag{2.5}$$

тоді

$$\tilde{W} = \tilde{M} H_{N^2} = (\tilde{X} + \tilde{T}) H_{N^2} = \tilde{X} H_{N^2} + \tilde{T} H_{N^2}. \tag{2.6}$$

Як бачимо з (2.6), вплив на трансформанти перетворення Уолша-Адамара блока контейнера X повністю визначається видом трансформант перетворення Уолша-Адамара вибраного кодового слова, і, оскільки воно вибірково впливає на відповідну трансформанту перетворення Уолша-Адамара, це дозволяє говорити про вбудовування додаткової інформації саме в цю трансформанту.

Розглянемо конкретний приклад для розміру блока 8×8. Нехай задано блок контейнера, для якого знайдемо матрицю трансформант перетворення Уолша-Адамара:

$$X = \begin{bmatrix} 93 & 102 & 102 & 106 & 110 & 115 & 116 & 118 \\ 99 & 109 & 102 & 112 & 117 & 114 & 116 & 115 \\ 103 & 114 & 106 & 111 & 121 & 116 & 123 & 111 \\ 113 & 116 & 113 & 115 & 115 & 114 & 121 & 116 \\ 113 & 113 & 115 & 113 & 113 & 109 & 109 & 115 \\ 128 & 111 & 114 & 117 & 114 & 114 & 117 & 116 \\ 126 & 111 & 112 & 116 & 109 & 111 & 128 & 130 \\ 118 & 117 & 110 & 114 & 111 & 107 & 111 & 120 \end{bmatrix},$$

$$W_X = \begin{bmatrix} 7256 & -20 & -64 & 40 & -128 & -40 & 80 & 20 \\ -36 & -4 & -40 & -28 & -68 & 0 & 0 & -8 \\ -102 & -22 & 6 & 2 & -30 & 14 & -74 & -10 \\ -70 & -34 & -6 & -14 & 34 & 18 & -22 & -58 \\ -108 & -48 & 0 & -88 & -156 & -108 & -88 & -20 \\ -44 & -4 & 28 & -8 & -20 & -8 & -20 & 4 \\ -62 & -54 & -54 & 22 & -70 & 14 & 42 & -6 \\ 62 & 26 & -46 & 10 & -10 & 62 & 50 & 62 \end{bmatrix}, \tag{2.7}$$

а також кодове слово, що впливає вибірково на трансформанту перетворення Уолша-Адамара (5,1), яка є низькочастотною, а отже, згідно з [162], при вбудовуванні додаткової інформації забезпечує найбільшу стійкість до атак проти вбудованого повідомлення:

$$T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}. \tag{2.8}$$

Тоді, згідно з (2.6) результуюче стеганоповідомлення та його трансформанти перетворення Уолша-Адамара виглядатимуть як:

$$M = \begin{bmatrix} 94 & 103 & 103 & 107 & 111 & 116 & 117 & 119 \\ 100 & 110 & 103 & 113 & 118 & 115 & 117 & 116 \\ 104 & 115 & 107 & 112 & 122 & 117 & 124 & 112 \\ 114 & 117 & 114 & 116 & 116 & 115 & 122 & 117 \\ 112 & 112 & 114 & 112 & 112 & 108 & 108 & 114 \\ 127 & 110 & 113 & 116 & 113 & 113 & 116 & 115 \\ 125 & 110 & 111 & 115 & 108 & 110 & 127 & 129 \\ 117 & 116 & 109 & 113 & 110 & 106 & 110 & 119 \end{bmatrix};$$

$$W_M = \begin{bmatrix} 7256 & -20 & -64 & 40 & -128 & -40 & 80 & 20 \\ -36 & -4 & -40 & -28 & -68 & 0 & 0 & -8 \\ -102 & -22 & 6 & 2 & -30 & 14 & -74 & -10 \\ -70 & -34 & -6 & -14 & 34 & 18 & -22 & -58 \\ -44 & -48 & 0 & -88 & -156 & -108 & -88 & -20 \\ -44 & -4 & 28 & -8 & -20 & -8 & -20 & 4 \\ -62 & -54 & -54 & 22 & -70 & 14 & 42 & -6 \\ 62 & 26 & -46 & 10 & -10 & 62 & 50 & 62 \end{bmatrix}. \tag{2.9}$$

Порівнюючи вирази (2.9) та (2.6), робимо висновок про те, що вбудовування додаткової інформації відбулося саме в трансформанту (5,1), адже тільки вона серед інших трансформант перетворення Уолша-Адамара зазнала змін.

2.3. Аналіз властивостей трансформант перетворення Уолша-Адамара під впливом кодового управління

Виявлення стеганоповідомлень потребує більш пильного дослідження патернів, дії яких піддається контейнер під час стеганоперетворення. Виявлення таких патернів потребує розуміння ймовірнісних та структурних характеристик трансформант перетворення Уолша-Адамара реальних зображень.

Твердження 1. Нехай задано матриці W_{X_j} розміру $N \times N$ трансформант перетворення Уолша-Адамара блоків X_j, $j = 1,2,...,n$ зображення, кожна з яких має форму:

$$W_{X_j} = \begin{bmatrix} w_{X_j,11} & w_{X_j,12} & \cdots & w_{X_j,1N} \\ w_{X_j,21} & w_{X_j,22} & \cdots & w_{X_j,2N} \\ \vdots & \vdots & \ddots & \vdots \\ w_{X_j,N1} & w_{X_j,N2} & \cdots & w_{X_j,NN} \end{bmatrix}, \tag{2.10}$$

тоді послідовність трансформант $u_{kl} = \begin{bmatrix} w_{X_1,kl} & w_{X_2,kl} & ... & w_{X_n,kl} \end{bmatrix}$ має нульове математичне очікування $\mathrm{E}[u_{kl}] = 0$ для будь-яких k,l окрім $k = l = 1$.

Доказ. Для доказу Твердження 1 зазначимо, що відповідно до (2.4) кожна матриця W_{X_j} може бути подана у вигляді вектора трансформант перетворення Уолша-Адамара $W_{X_j} = X_j H_{N^2} = [w_{X_1,11} \quad w_{X_1,12} \quad ... \quad w_{NN}]$, Таким чином, кожний коефіцієнт $w_{X_1,kl}$, отриманий шляхом множення відповідного вектора X, сформованого шляхом послідовної конкатенації рядків матриці блока X на відповідний рядок матриці Уолша-Адамара H_{N^2}, який за побудовою є функцією Уолша h_g довжини N^2. Тоді можемо записати відповідний коефіцієнт $w_{X_j,kl}$ як:

$$w_{X_j,kl} = \sum_{\alpha=1}^{N^2} h_{g,\alpha} x_\alpha \,. \tag{2.11}$$

Іншими словами, оскільки значення інтенсивності пікселів довільного зображення x_α не залежать від елементів функцій Уолша $h_{g,\alpha}$, математичне очікування $\mathrm{E}[w_{X_j,kl}]$ для кожного $w_{X_j,kl}$ буде визначатися як:

$$\mathrm{E}[w_{X_j,kl}] = \mathrm{E}[h_{g,\alpha}x_\alpha] = \mathrm{E}[h_{g,\alpha}]\mathrm{E}[x_\alpha]. \quad (2.12)$$

Оскільки за своєю побудовою функції Уолша є збалансованими при будь-якому $g \neq 1$, то $\sum_{\alpha=1}^{N^2} h_{g,\alpha} = 0$, а тому і добуток $\mathrm{E}[w_{X_j,kl}] = 0$.

Враховуючи, що $\mathrm{E}[w_{X_j,kl}] = 0$ для $\forall k,l$ окрім $k = l = 1$, тобто коли $g \neq 1$, отримуємо, що $\mathrm{E}[u_{kl}] = 0$ для будь-яких k,l окрім $k = l = 1$, що доводить умови Твердження 1.

Обчислювальний експеримент 1.

Для практичної перевірки умов Твердження 1 проведемо такий обчислювальний експеримент. Для вибірки з 500 зображень з бази NRCS [166] сформуємо вектор u_{kl} для блоків розміру 4×4, 8×8, 16×16, після чого будемо знаходити середнє значення елементів вектора u_{kl} для всіх значень k,l відповідно до розміру блока.

Результати обчислювального експерименту для блоків розмірів 4×4 та 8×8 показані в табл. 2.1.

Таблиця 2.1 – Емпірично побудовані середні значення трансформант перетворення Уолша-Адамара

Розмір блока 4×4				
k/l	1	2	3	4
1	$1.8\cdot10^3$	0	0	0
2	0.1	0	0	0
3	0.4	0	0	0
4	0	0	0	0

Розмір блока 8×8								
k/l	1	2	3	4	5	6	7	8

1	$7.3\cdot10^3$	0	0	0.2	-0.2	0	0	0
2	0.7	0	0	0.1	0	0	0	0
3	1.7	0	0	0	0	0	0	0
4	0.3	0	0	0.2	0	0	0	0
5	3.0	0.2	0	-0.1	0	-0.1	0	-0.1
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0

Зазначимо при цьому, що для блоків розміру 16×16 середнє значення елемента (1,1) дорівнює $2.9\cdot10^4$, тоді як інші значення при обчислювальному експерименті фактично дорівнюють 0.

Аналіз наведених у табл. 2.1 даних призводить до практичного підтвердження Твердження 1, адже середні значення трансформант перетворення Уолша-Адамара при усередненні по блоках дійсно на практиці близькі до 0, окрім випадку значень $k=l=1$.

Відзначимо при цьому, що середньоквадратичне відхилення і дисперсія значень векторів u_{kl} на практиці дуже сильно залежать від конкретного зображення та його структури, що, на нашу думку, встановлює низку обмежень щодо їх узагальнення та обмежує їх застосування на практиці для виявлення стеганоповідомлень.

Для більшої наочності сформуємо гістограми розподілу величин елементів векторів u_{15} і u_{51} для розміру блоків 8×8 (рис. 2.1) [19].

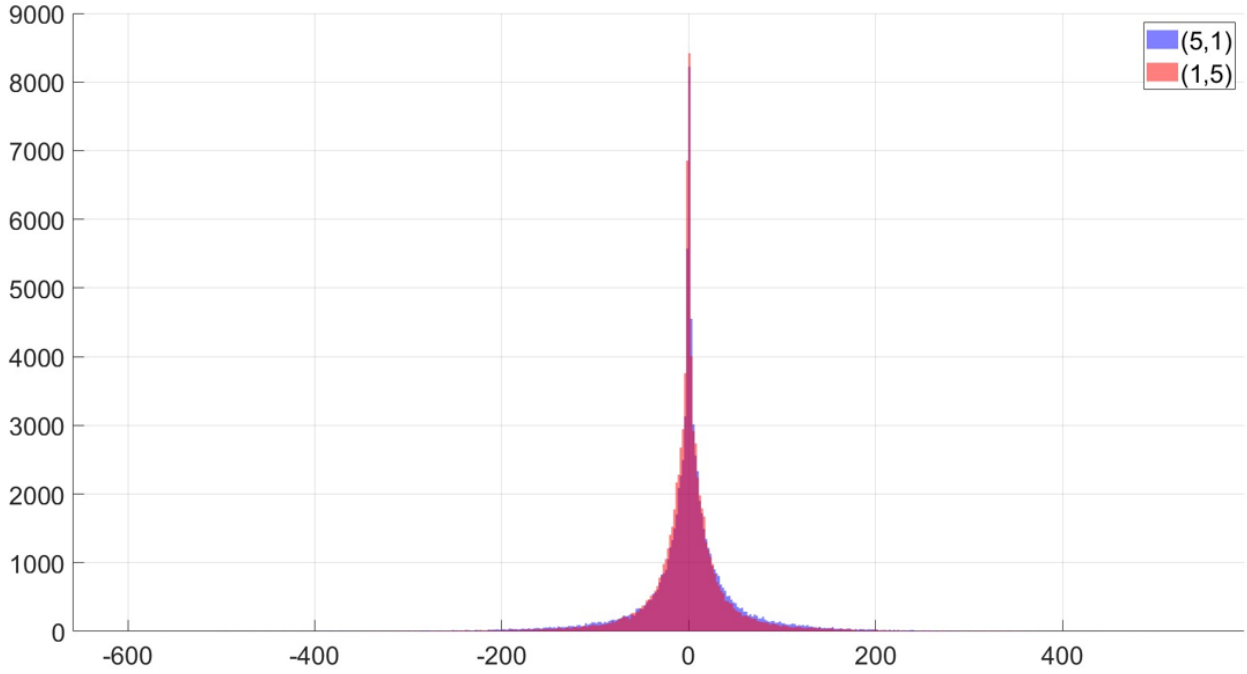


Рисунок 2.1 — Гістограма розподілу величин елементів u_{15} і u_{51}

Дослідимо вплив стеганоперетворення із застосуванням методу з кодовим управлінням вбудовуванням додаткової інформації на статистичні характеристики векторів u_{kl}.

Відповідно до умов (2.6) застосування стеганографічного методу з кодовим управлінням із кодовими словами, що селективно впливають на задану трансформанту перетворення Уолша-Адамара, призводить до зміни її значення у кожному блоці на величину N^2.

Твердження 2. Гістограма розподілу векторів u_{kl}, в які здійснювалося вбудовування додаткової інформації із застосуванням стеганографічного методу з кодовим управлінням із кодовими словами на основі функцій Уолша, матимуть двомодальний характер із максимумами в точках $\pm N^2$.

Доказ. Для доказу Твердження 2 зазначимо, що однією із важливих складових частин стеганографічної системи є прекодер, функцією якого є формування послідовності $\{d_j\}$ за допомогою операцій аналогово-цифрового перетворення (за потреби), ефективного кодування, завадостійкого кодування інформації, шифрування. Застосування високоякісних алгоритмів шифрування

призводить до рівномірного розподілу символів «0» та «1» у послідовності $\{d_j\}$. Тому в контексті застосування стеганографічного методу з кодовим управлінням вважатимемо, що розподіл символів послідовності $\{d_j\}$ є рівномірним.

Із врахуванням зазначеного із статистичної точки зору вектори u'_{kl} стеганоповідомлення можна подати як:

$$u'_{kl} = \begin{cases} u'_{kl} + N^2, \text{ з ймовірністю } 0.5; \\ u'_{kl} - N^2, \text{ з ймовірністю } 0.5. \end{cases} \quad (2.13)$$

Згідно з умовами Твердження 1 густина ймовірності $f_{u_{kl}}$ має максимум (оскільки випадкова величина u_{kl} розподілена за симетричним унімодальним розподілом, що підтверджується отриманими емпіричними даними для заданої вибірки зображень, то максимум густини ймовірностей $f_{u_{kl}}$ збігатиметься з його математичним очікуванням) у точці 0. Тоді густина ймовірностей $f_{u_{kl}}(u_{kl} - N^2)$ має максимум при $u_{kl} - N^2 = 0,$ тобто при $u_{kl} = N^2$. Аналогічно $f_{u_{kl}}(u_{kl} + N^2)$ матиме максимум при $u_{kl} + N^2 = 0,$ тобто при $u_{kl} = -N^2$. Отже, $f_{u'_{kl}}$ матиме два максимуми: у точках $-N$ та N, оскільки обидва доданки вносять свої максимуми незалежно.

Викладене доводить умови Твердження 2.

На рис. 2.2 наведено гістограми розподілу u_{51} для оригінального зображення, а також u'_{51} стеганоповідомлення для трансформанти перетворення Уолша-Адамара (5,1), в яку була вбудована додаткова інформація із застосуванням відповідного кодового слова (2.8) розміру 8×8.

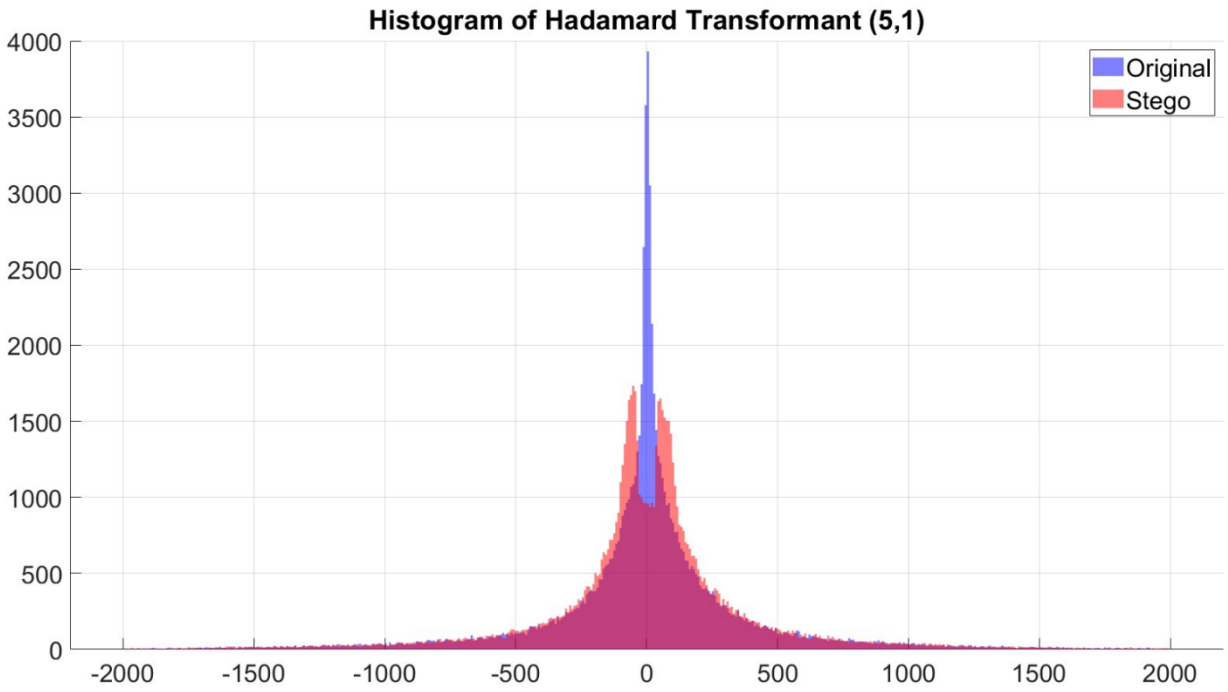


Рисунок 2.2 – Гістограма розподілу u_{51} для оригінального зображення, а також u'_{51} стеганоповідомлення

Аналіз даних рис. 2.2 наочно підтверджує умови Твердження 2: для стеганоповідомлення, на відміну від оригінального зображення, розподіл трансформанти перетворення Уолша-Адамара, яка зазнала вбудовування додаткової інформації, носить двомодальний характер із максимумами у величинах $\pm N = \pm 64$, що підтверджує, що величина блока μ, в який відбулося вбудовування дійсно складає $\mu = 8$.

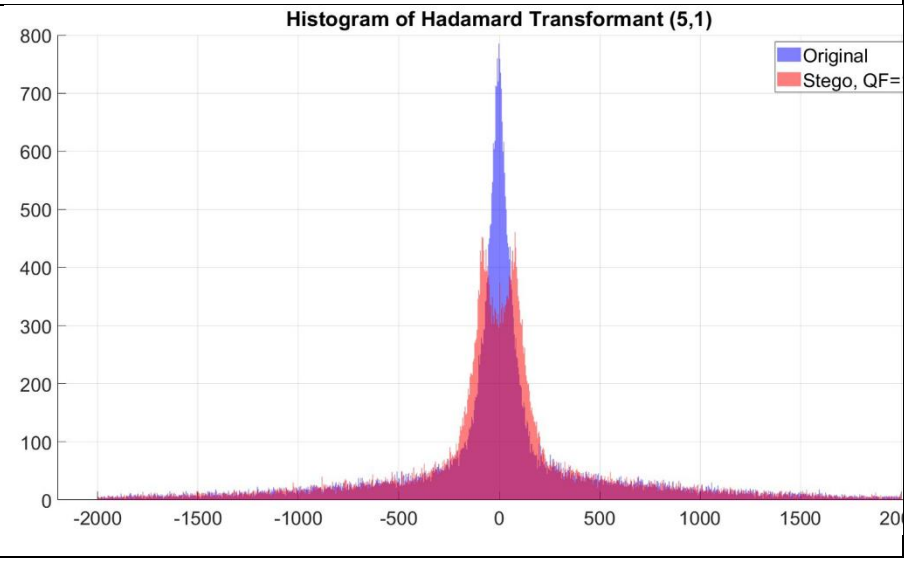
Очевидно, враховуючи умови Твердження 2, такий вид відхилення гістограми розподілу векторів u_{kl} від класичного одномодального розподілу ймовірностей є ознакою застосування стеганографічного методу з кодовим управлінням для вбудовування додаткової інформації у задану трансформанту перетворення Уолша-Адамара.

Важливим фактором, від якого залежатимуть можливості практичного застосування Твердження 2 для виявлення вбудовування додаткової інформації із застосуванням стеганографічного методу з кодовим управлінням, є його виконання за умов збурних дій, зокрема атаки стисненням, яка є однією з

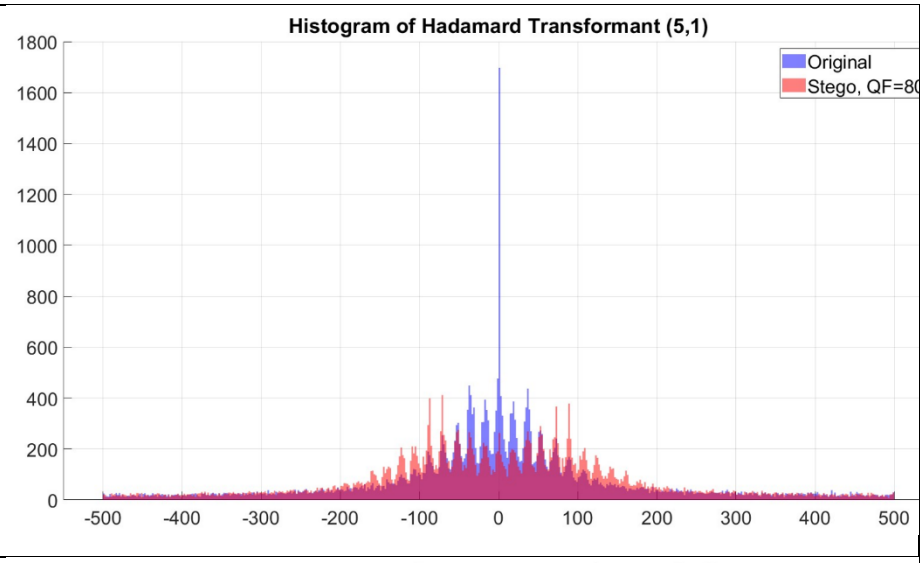
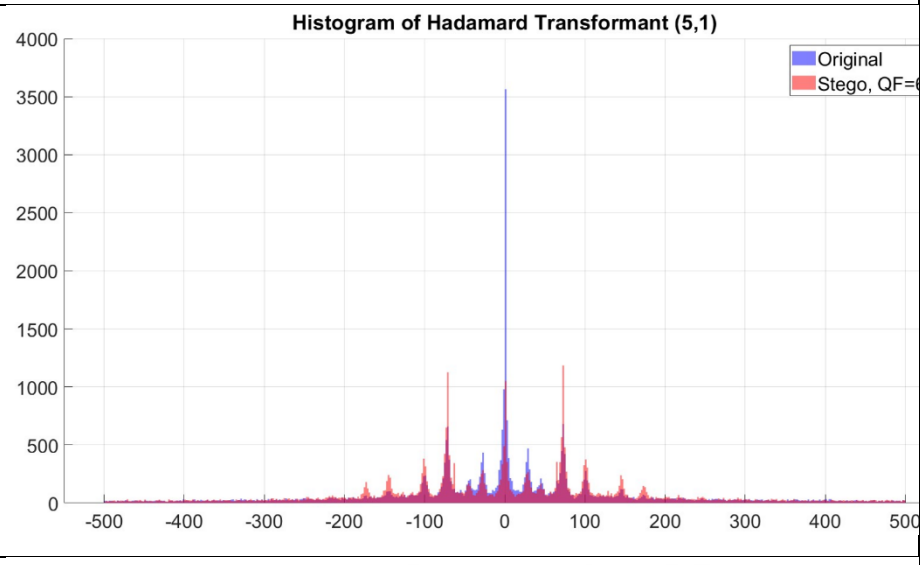
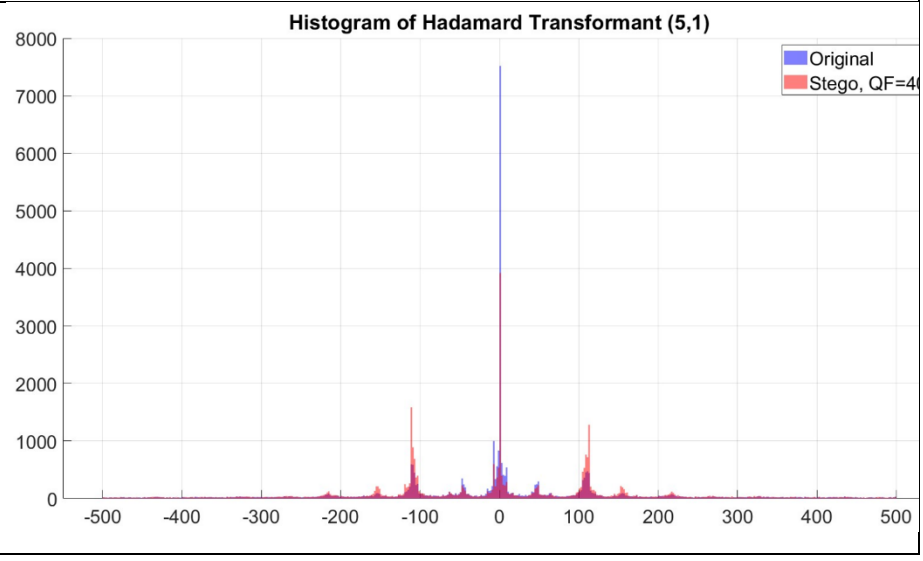
найбільш розповсюджених атак проти вбудованого повідомлення. Експериментальні дані [162] підтверджують стійкість цього методу до атак проти вбудованого повідомлення.

У табл. 2.2 наведені гістограми векторів u'_{51} стеганоповідомлень, які піддавалися атаці стисненням проти вбудованого повідомлення з різними значеннями коефіцієнта якості $QF = \{100, 80, 60, 40, 20\}$.

Таблиця 2.2 – Гістограми векторів u'_{51} стеганоповідомлень для різних рівнів стискання QF

Рівень стиску QF	Гістограма векторів u'_{51}
QF=100	

Закінчення табл. 2.2

QF=80	
QF=60	
QF=40	

Аналіз даних табл. 2.2 призводить до висновку, що попри те, що стиснення зображення призводить до мультимодальності розподілу

трансформант перетворення Уолша-Адамара, бачимо, що для стеганоповідомлення стеганографічне втручання лишається помітним через значно більший рівень концентрації значень трансформанти перетворення Уолша-Адамара у бічних пелюстках гістограми, в той час як для оригінальних зображень ця концентрація залишається значною біля нульового значення. Означене робить можливим виявлення вбудовування додаткової інформації із застосуванням стеганографічного методу з кодовим управлінням за допомогою умов Твердження 2, навіть після атаки стисненням проти вбудованого повідомлення.

2.4. Основні поняття та математичний апарат дослідження стеганоаналізу SVD UV-методу

Сьогодні все частіше використовуються методи стеганографії на основі сингулярного розкладу, зокрема SVD UV, для вбудовування додаткової інформації у цифрові зображення. Такі методи відзначаються високою стійкістю до стандартних атак та здатністю зберігати візуальну якість об’єктів, що робить їх привабливими для легітимного використання. Водночас активне застосування SVD UV-методів може становити загрозу для вебзастосунків, оскільки дозволяє приховано передавати інформацію та потенційно обходити механізми безпеки, що підвищує ризик несанкціонованого доступу або втрати конфіденційності даних. Тому дослідження ефективних методів стеганоаналізу для таких методів залишається актуальною задачею в контексті забезпечення безпеки сучасних інформаційних систем.

Зважаючи на потенційні загрози, пов’язані з використанням SVD UV-методів для прихованого передавання інформації, особливу актуальність набуває питання ефективного виявлення та аналізу таких вбудованих даних. Саме цій проблемі присвячено численні дослідження у сфері стеганоаналізу, які пропонують різні підходи для розпізнавання та оцінки втручання у цифрові об’єкти.

Один з відомих напрямів таких досліджень стосується комбінування SVD з іншими перетвореннями, зокрема дискретним косинусним перетворенням (DCT). У роботі [167] сингулярні значення DCT-коефіцієнтів використовуються для універсального стеганоаналізу, що забезпечує можливості виявлення різних типів вбудовувань. Водночас подвійне застосування DCT та SVD суттєво підвищує обчислювальне навантаження, роблячи такий підхід малопридатним для сценаріїв з обмеженими ресурсами.

Значні теоретичні досягнення щодо застосування SVD-простору для побудови універсальних методів стеганоаналізу представлені в роботі [168], проте слід враховувати, що використання таких підходів пов’язане з високою обчислювальною складністю. Аналогічно, у роботі [169], присвяченій методу, який базується на взаємозв’язку між трансформантами перетворення DCT, також застосовуються ресурсоємні перетворення, а також методи штучного інтелекту для аналізу залежностей між трансформантами DCT.

Сучасні методи стеганоаналізу часто засновані на машинному навчанні. У статті [170] автори досліджують використання моделей глибокого навчання для стеганоаналізу зображень. Вони застосовують різні архітектури нейронних мереж для виявлення прихованих даних у зображеннях, порівнюючи їх ефективність та точність. Результати показують, що моделі з глибоким навчанням можуть ефективно виявляти стеганографічні дані, однак їхня точність залежить від якості навчальних даних та налаштувань моделей. Робота [171] надає огляд застосування методів машинного навчання для стеганоаналізу. Автор аналізує різні підходи та алгоритми, які використовуються для виявлення прихованих даних у мультимедійних файлах, зокрема зображеннях. Також обговорюються переваги та обмеження кожного з методів, підкреслюючи важливість вибору відповідних алгоритмів та налаштувань для досягнення високої точності виявлення.

У роботі [172] автори пропонують модель стеганоаналізу, яка поєднує глибоку нейронну мережу з алгоритмом θ-NSGA-III для налаштування гіперпараметрів. Це дозволяє досягти високої точності виявлення прихованих

даних у зображеннях, зокрема на наборі даних STEGRT1. Однак, як і в більшості моделей з глибоким навчанням, є ризик перенавчання та потреба в значних обчислювальних ресурсах. У статті [173] подано HSDetect-Net – метод на основі глибокого навчання, який використовує нечітку логіку для виявлення прихованих даних у цифрових зображеннях. Запропонований підхід показує високу точність стеганоаналізу; однак він вимагає значних обчислювальних ресурсів через складність моделі та обробку великих обсягів даних. Важливий огляд застосування методів машинного та глибокого навчання в стеганоаналізі наведено у [174]. Автори детально розглядають різні підходи, включно з використанням нейронних мереж для виявлення прихованих повідомлень у зображеннях. Особливу увагу приділено методам, які дозволяють ефективно виявляти стеганографічні дані навіть у складних умовах зашумлених каналів передачі.

Проте без спеціальних математичних напрацювань, представлених у цьому розділі, підходи засновані на методах машинного навчання вимагають значних обчислювальних потужностей, спеціалізованого апаратного забезпечення й оптимізованих обчислювальних середовищ, що суперечить вимогам до швидкодії та масштабованості у вебзастосунках.

Таким чином, більшість наявних рішень характеризується достатньою ефективністю, але надмірною обчислювальною складністю, що обмежує їх практичне використання у вебплатформах, де необхідна обробка великого потоку мультимедійних файлів у реальному часі при мінімальному навантаженні на сервер. Це створює об’єктивну потребу у пошуку нових підходів, які б поєднували високу точність детектування з обчислювальною простотою, зберігаючи можливість масштабованого оброблення даних.

Для вебзастосунків, які обробляють великі обсяги мультимедійних даних і потребують мінімальної затримки взаємодії з користувачем, ключовим фактором стає не лише точність, а й швидкодія методів стеганоаналізу. Тому постає завдання розробки алгоритмів, які поєднують високу чутливість до

стеганографічних вбудовувань із можливістю ефективного використання обчислювальних ресурсів у динамічному середовищі вебплатформ.

Перспективним напрямом підвищення ефективності стеганоаналізу за умов обмежених обчислювальних ресурсів є використання області перетворень Уолша-Адамара. Це перетворення характеризується нижчою обчислювальною складністю, порівняно з іншими часто застосовуваними ортогональними перетвореннями, що робить його привабливим для використання у вебзастосунках, де критичною є швидкодія алгоритмів. При цьому вище в цьому розділі було подано підходи, які використовують простір перетворення Уолша-Адамара для виявлення прихованих каналів, зокрема побудованих на основі концепції кодового управління. Останні є особливо складними для детектування через їхню високу стійкість і здатність зберігати статистичні характеристики контейнера, що додатково підкреслює актуальність дослідження методів стеганоаналізу інших стеганографічних алгоритмів саме в цій області перетворень.

Загалом розвиток методів стеганоаналізу в області перетворень Уолша-Адамара відкриває шлях до створення універсальних алгоритмів виявлення прихованих каналів із мінімальними обчислювальними витратами. Це особливо важливо для вебзастосунків, які працюють із великими обсягами мультимедійних даних і потребують високої швидкодії систем безпеки без зниження точності детектування. Формування такого підходу дозволило б значно підвищити рівень захисту від витоку інформації та зловживань, пов’язаних із використанням стеганографії у сучасних інтерактивних платформах.

Отже, актуальним завданням є подальший розвиток методів стеганоаналізу, які працюють у просторі перетворень Уолша-Адамара. Важливо не лише забезпечити ефективне виявлення прихованих каналів, побудованих на основі концепції кодового управління, а й розширити можливості таких підходів для детектування інших високоефективних стеганографічних методів. Зокрема, значний науковий інтерес становлять

розповсюджені на практиці алгоритми, які використовують сингулярні вектори, оскільки вони забезпечують високу стійкість прихованих повідомлень і зберігають надійність сприйняття, що робить їх особливо складними для виявлення.

Стеганографічні методи на основі SVD сьогодні є одними з найпоширеніших завдяки їхній високій стійкості до атак проти вбудованого повідомлення. Розрізняють два основні підходи: 1) методи, які модифікують матрицю сингулярних значень (Σ-методи), та 2) методи, які працюють із матрицями лівих і правих сингулярних векторів (UV-методи). Другі, UV-методи, характеризуються значно більшою стійкістю до різних видів спотворень і забезпечують високий рівень надійності сприйняття прихованої інформації, вони є більш поширеними на практиці.

Одним із потужних підходів до приховування даних у мультимедійних контейнерах є запропонований в роботі [175] стеганографічний метод, який заснований на модифікації значень сингулярних векторів, що відповідають найбільшим сингулярним числам блока. Такий підхід дозволяє забезпечити високу стійкість прихованого каналу завдяки використанню властивостей сингулярного розкладу: навіть незначні зміни власних векторів можуть переносити інформацію без помітної деградації якості контейнера та зберігаючи при цьому високу стійкість до атак проти вбудованого повідомлення. Далі для повноти викладу матеріалу ми розглянемо метод, що реалізує зазначений принцип шляхом поетапної модифікації лівих або правих сингулярних векторів для кожного блока контейнера, а також відповідні процедури вбудовування та вилучення даних.

Введемо необхідні визначення. Сингулярний розклад є одним із найпотужніших інструментів обробки сигналів і зображень, що знаходить широке застосування у стеганографії. Завдяки здатності відокремлювати енергію сигналу в сингулярних числах та відображати геометричні особливості у власних векторах, SVD дозволяє створювати методи вбудовування, які поєднують високу прихованість і стійкість до атак.

Для блока B визначимо сингулярний розклад як [139]

$$B = U\Sigma V^T, \tag{2.14}$$

де U,V – ортогональні матриці лівих лексикографічно позитивних та правих сингулярних векторів, відповідно, $\Sigma = diag(\sigma_1,...,\sigma_8)$ – матриця сингулярних чисел.

Вбудовування додаткової інформації [175]

Крок 1. Матриця контейнера F стандартно розбивається на 8×8-блоки; B – довільний блок.

Крок 2. У черговий блок B вбудовується черговий біт p_i додаткової інформації:

2.1. Для блока B будується сингулярний розклад (2.14); u_1 і v_1 – лівий і правий сингулярні вектори блока B, відповідно, що відповідають сингулярному числу σ_1.

2.2. (Вбудовування): *Якщо* $p_i = 1$, то $\overline{u}_1 = n^o$, де $\overline{u}_1$ – збурений у ході стеганоперетворення u_1, n^o – n-оптимальний вектор. Привести ліві сингулярні вектори $u_2,...,u_8$ блока B до ортонормованого з $\overline{u}_1$ лексикографічно позитивного вигляду. Результат – $\overline{u}_2,...,\overline{u}_8$.

Інакше $\overline{v}_1 = n^o$ де $\overline{v}_1$ – збурений у ході стеганоперетворення v_1. Привести праві сингулярні вектори $v_2,...,v_8$ блока B до ортонормованого з $\overline{v}_1$ виду. Результат – $\overline{v}_2,...,\overline{v}_8$.

2.3. (Формування блока стеганоповідомлення, що відповідає блоку контейнера). Якщо $p_i = 1$, то $\overline{B} = \overline{U}\Sigma V^T$ де $\overline{U} = (n^o,\overline{u}_2,...,\overline{u}_8)$, інакше $\overline{B} = U\Sigma\overline{V}^T$, де $\overline{V} = (n^o,\overline{v}_2,...,\overline{v}_8)$.

Вилучення додаткової інформації [175]

Крок 1. Матриця стеганоповідомлення $\overline{F}$ розбивається стандартним чином на 8×8-блоки; $\overline{B}$ – довільний блок.

Крок 2. З чергового блока $\overline{B}$ витягується черговий біт $\overline{p}_i$ додаткової інформації:

2.1. Для $\overline{B}$ будується сингулярний розклад: $\overline{B} = \overline{U}\overline{\Sigma}\overline{V}^T$; $\bar{u}_1$ і $\bar{v}_1$ – лівий і правий сингулярні вектори блока $\overline{B}$ відповідно, що відповідають максимальному сингулярному числу $\overline{\sigma}_1$.

2.2. (Витягування $\overline{p}_i$). Знайти UN_B і VN_B – кути між векторами $\bar{u}_1$, $\boldsymbol{n}^o$ і $\bar{v}_1$, $\boldsymbol{n}^o$, відповідно.

Якщо $UN_B < VN_B$, то $\overline{p}_i = 1$, інакше $\overline{p}_i = 0$.

Враховуючи природу розглянутого методу, зміни, які він вносить у блоки контейнера, не є детермінованими та залежать від початкового вигляду відповідних блоків. Саме ця особливість забезпечує високі показники стійкості вбудованого повідомлення до атак, оскільки приховані дані адаптуються до структури контейнера та залежать від його структури. Водночас така ж характеристика ускладнює встановлення чіткої відповідності між модифікаціями в області сингулярного розкладу та їх відображенням у просторі перетворень Уолша-Адамара. Це зумовлює потребу не лише теоретичного осмислення механізмів впливу таких змін, а й проведення практичних експериментів для виявлення закономірностей та оцінки ефективності стеганоаналізу в цій області.

2.5. Теоретичні основи побудови несліпого методу виявлення стеганоповідомлень на основі SVD UV-методу

Для ефективного стеганоаналізу методу, який базується на модифікації сингулярних векторів, надзвичайно важливо дослідити, як ці зміни впливають на трансформанти Уолша-Адамара блоків контейнера. Оскільки сам метод вносить недетерміновані модифікації, які залежать від початкової структури блоків, аналіз трансформант Уолша-Адамара дозволяє виявити характерні

ознаки впливу вбудованої інформації. Це дослідження є ключовим для розробки стеганоаналітичних алгоритмів, здатних надійно та швидко ідентифікувати присутність прихованих повідомлень у просторі перетворень Уолша-Адамара, що має важливе значення для використання у реальних вебзастосунках.

Обчислювальний експеримент 2.

Задля дослідження впливу стеганографічного методу [175] на ті чи інші трансформанти перетворення Уолша-Адамара проведемо такий експеримент на множині з 530 зображень у форматі без втрат розміру 1024x1024 пікселів з бази [166]. Наведемо алгоритм проведеного експерименту, який застосовувався для обробки кожного із зображень:

1. Виконати у чергове зображення F вбудовування додаткової інформації відповідно до алгоритму [175] (в якості якої застосовувалася псевдовипадкова рівномірно розподілена послідовність), в результаті чого отримуємо стеганоповідомлення $\overline{F}$.

2. Сегментувати отримані зображення F і $\overline{F}$ на блоки B_i і $\overline{B_i}$, $i = 1,2,...,16384$ розміру 8x8, стандартним чином.

3. Відповідно до (2.1) знайти для кожного блока B_i і $\overline{B_i}$ матриці двовимірного перетворення Уолша-Адамара W_{B_i} і $W_{\overline{B_i}}$.

4. Для кожної пари перетворень W_{B_i} і $W_{\overline{B_i}}$ знайти матрицю різниць, яка відображатиме зміну у трансформантах перетворення Уолша-Адамара у контейнері, що обумовлена стеганоперетворенням в області сингулярного розкладання:

$$\Delta_i = \left| W_{\overline{B_i}} - W_{B_i} \right|, \quad i = 1,2,...,16384. \tag{2.15}$$

5. Усереднити значення кожного з елементів матриць Δ_i по всіх блоках серед всіх зображень, які беруть участь в експерименті.

На рис. 2.3 показано теплову карту змін у трансформантах перетворення Уолша-Адамара, які виникають через застосування стеганографічного методу [175].

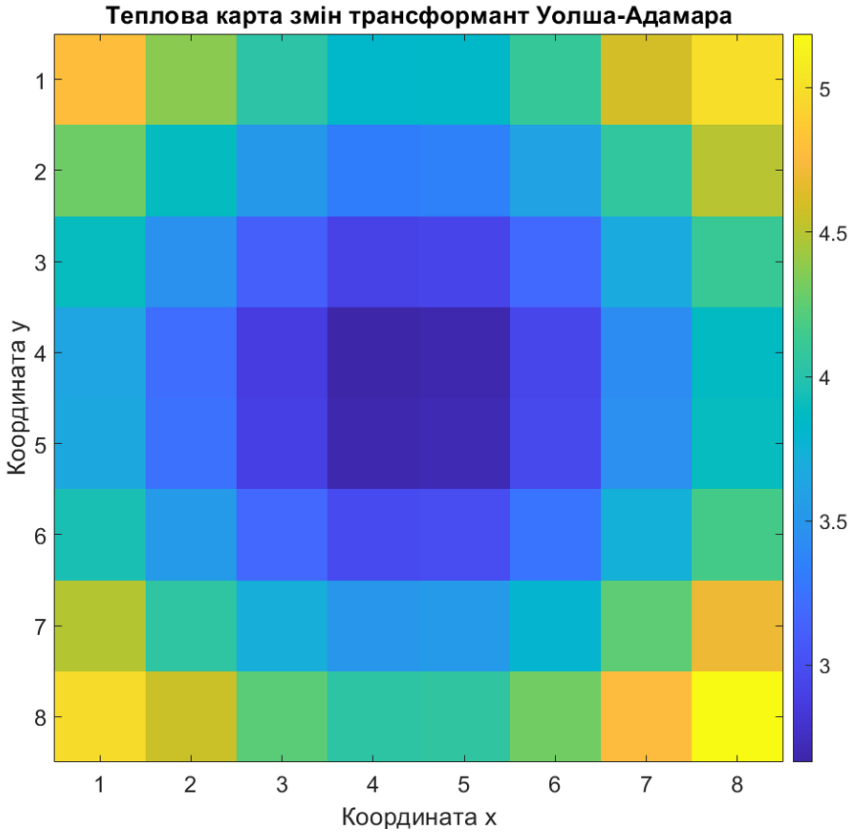


Рисунок 2.3 – Теплова карта змін у трансформантах перетворення Уолша-Адамара, обумовлених стеганоперетворенням

Аналіз даних на рис. 2.3 призводить до висновку, що зміни, зумовлені стеганоперетворенням в області сингулярного розкладання згідно з методом [175] зосереджуються здебільшого в кутових областях матриці трансформант перетворення Уолша-Адамара, при цьому найбільшу амплітуду впливу мають трансформанти (8,8), (1,8), (8,1), (1,1).

Проведений аналіз 530 зображень, кожне з яких було представлене у вигляді набору блоків розміром 8×8, дозволив визначити позиції трансформант Уолша-Адамара, які найчастіше зазнавали максимальних змін за модулем. У результаті з’ясовано, що в більшості випадків, а саме у 494 з 530 зображень,

найбільші зміни спостерігалися в позиції (8,8), що може свідчити про високу чутливість або змінність високочастотного коефіцієнта, відповідного цій позиції у базисі трансформанти. Дещо рідше, у 21 випадку, максимум припадав на позицію (8,1), а ще у 13 – на (1,8). У двох випадках найбільша зміна фіксувалася в позиції (1,1). В усіх інших положеннях матриці коефіцієнтів максимумів змін не зафіксовано.

За результатами, зображеними на рис. 2.3, можна стверджувати, що в рамках вибраного експериментального алгоритму виявлено певні трансформанти перетворення Уолша-Адамара, які демонструють найбільшу амплітуду змін при порівнянні стеганоповідомлення з оригінальним контейнером. Однак потрібно чітко підкреслити: ці спостереження виникають з аналізу різниці стеганоповідомлення / контейнер, тобто вони є результатом не сліпого детектування, а радше інструментом для сценаріїв, де обидва зразки доступні або коли є велика група еталонних контейнерів для порівняння. Такий підхід корисний в якості евристичного або в якості складової несліпого методу, але його висновки не автоматично переносяться на випадок, коли є лише саме стеганоповідомлення.

У випадку несліпого стеганоаналізу та підозри на вбудовування прихованих даних замість дослідження всіх 64 коефіцієнтів достатньо проаналізувати лише чотири, що значно скорочує обчислювальні витрати та спрощує процес виявлення до 16 разів.

Водночас у випадку, коли контейнер недоступний, використання лише цих позицій без додаткового обґрунтування може виявитися недостатньо надійним. Коли ми говоримо про аналіз стеганоповідомлення, необхідно врахувати, що великі абсолютні зміни не завжди означають сильний статистичний зсув. Наприклад, якщо значення трансформанти (8,8) сильно коливаються в обидва боки (збільшуються та зменшуються), середнє значення може залишатися майже незмінним. Менші, але систематичні зміни можуть давати виразніший зсув. Наприклад, якщо деяка інша трансформанта

стабільно зростає (навіть незначно), це призведе до помітного зсуву в її розподілі.

2.6. Теоретичні основи побудови сліпого методу виявлення стеганоповідомлень на основі SVD UV-методу

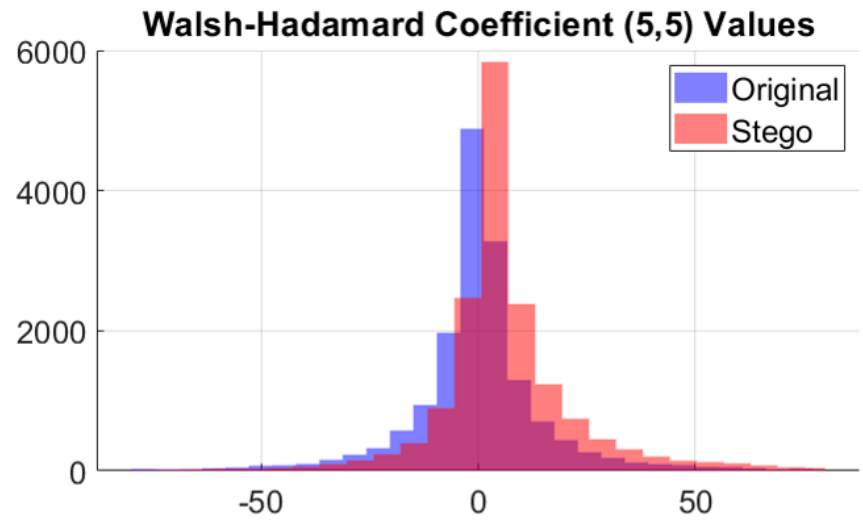
Задля проведення більш тонкого аналізу статистичних характеристик трансформант перетворення Уолша-Адамара стеганоповідомлення розглянемо оригінальне зображення з бази [166], а також стеганоповідомлення, створене із застосуванням методу [175] на його основі.

Зазначимо, що вище в цьому розділі вже доведено Твердження 1, згідно з яким, якщо задано матриці W_{X_j} розміру $N \times N$ трансформант перетворення Уолша-Адамара блоків X_j, $j = 1,2,...,n$ зображення, кожна з яких має форму (2.10), послідовність трансформант $u_{kl} = \left[w_{X_1,kl} \quad w_{X_2,kl} \quad ... \quad w_{X_n,kl} \right]$ має нульове математичне очікування $\mathrm{E}[u_{kl}] = 0$ для будь-яких k, l окрім $k = l = 1$.

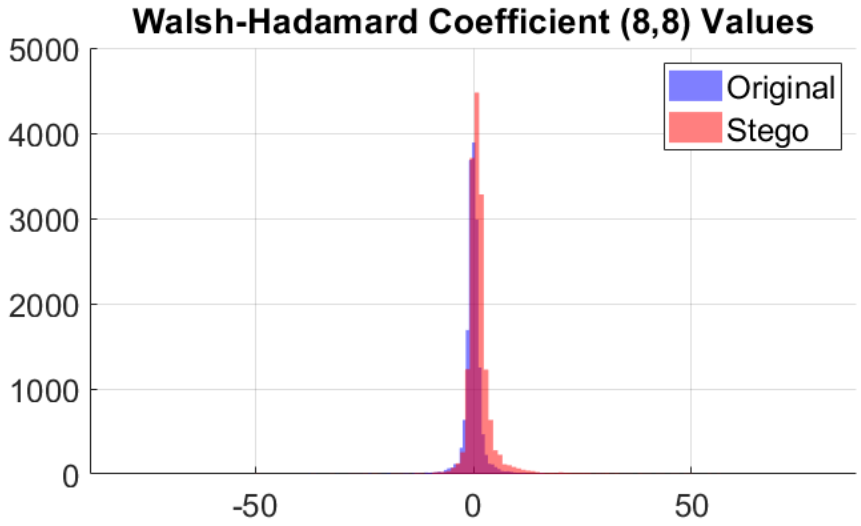
Зазначене твердження є дуже важливим з точки зору можливостей стеганоаналізу, оскільки дозволяє дійти висновку про відсутність або наявність збурення у зображенні на основі даних про їх статистику.

Наприклад, на рис. 2.4 показані гістограми розподілу трансформант перетворення Уолша-Адамара (5,5) і (8,8). Відзначимо, що згідно з рис. 2.3 трансформанта (8,8) відноситься до тих, які входять у зону максимального впливу стеганографічного методу [175], для вихідного зображення і зображення, що зазнало стеганографічного втручання.

Як бачимо з рис. 2.4, гістограма значень трансформанти (5,5) зазнала більш значного зсуву, як порівняти з гістограмою трансформанти (8,8), про можливість чого було зазначено раніше. Максимальні значення амплітуди впливу на ту чи іншу трансформанту перетворення Уолша-Адамара не обов’язково означають максимальний зсув її статистичних особливостей.



а)



б)

Рисунок 2.4 – Гістограми розподілу трансформант а) – (1,8), б) – (8,8) для оригінального зображення і стеганоповідомлення

Наведемо матриці, які міститимуть середні значення модулів всіх послідовностей u_{kl} і $\bar{u}_{kl}$ трансформант перетворення Уолша-Адамара для досліджуваного зображення і стеганоперетворення:

$$\mathrm{E}[u_{kl}] = \begin{bmatrix} 1091.2 & -0.03 & -0.11 & 0 & -0.06 & 0.02 & -0.04 & -0.01 \\ 0.04 & 0 & 0 & 0 & -0.01 & 0 & 0 & 0 \\ 0.02 & 0.01 & 0.01 & -0.01 & 0.09 & 0.02 & 0.01 & -0.01 \\ 0.03 & 0 & 0.02 & -0.01 & -0.06 & 0 & -0.02 & 0 \\ 0.04 & 0 & -0.02 & 0.04 & 0 & 0 & 0 & 0 \\ 0.01 & 0 & -0.02 & 0 & 0.05 & 0 & 0.02 & 0 \\ -0.08 & 0 & -0.02 & -0.02 & 0.18 & 0.01 & 0.03 & -0.05 \\ 0.07 & 0 & 0.02 & -0.02 & 0.03 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathrm{E}[\bar{u}_{kl}] = \begin{bmatrix} 1093.2 & 0.12 & 0.18 & -0.03 & 0.46 & -0.06 & -0.09 & 0 \\ 0.06 & 0.75 & 0.68 & 0.17 & 1.02 & 0.36 & 0.04 & 0.09 \\ 0.05 & 0.55 & 2.91 & 0.21 & 1.89 & 0.12 & 0.27 & 0.06 \\ -0.02 & 0.31 & 0 & 1.58 & 0 & -0.33 & -0.05 & 0.37 \\ 0.12 & 1.03 & 1.84 & 0.03 & 7.32 & 0.10 & 0.21 & -0.95 \\ -0.01 & 0.36 & -0.04 & -0.16 & 0.05 & 1.14 & 1.24 & 0.15 \\ -0.07 & 0.08 & 0.14 & -0.19 & 0.13 & 1.14 & 3.94 & 0.18 \\ 0.06 & -0.08 & 0.32 & 0.38 & -0.95 & 0.35 & 0.04 & 1.37 \end{bmatrix}. \quad (2.16)$$

Бачимо, що найбільший зсув відбувся у трансформанті перетворення Уолша-Адамара (5,5). Задля подальших практичних досліджень цього феномену проведемо такий експеримент.

Обчислювальний експеримент 3.

В якості вихідного матеріалу для цього експерименту були задіяні 530 стеганоповідомлень розміру 1024x1024 у форматі без втрат (PNG), які були згенеровані на основі бази [115].

Порядок дій у даному експерименті подамо у вигляді конкретних кроків, які мають бути виконані для кожного із зображень.

Крок 1. Завантажити стеганоповідомлення $\bar{F}$. Виконати його розбиття на блоки X_i розміру 8x8 стандартним чином.

Крок 2. Для кожного отриманого блока відповідно до (2.1) знайти матрицю трансформант перетворення Уолша-Адамара W_{X_j}, при цьому кожна з цих матриць матиме структуру (2.16).

Крок 3. Побудувати послідовності трансформант перетворення Уолша-Адамара $\bar{u}_{kl} = \begin{bmatrix} w_{X_1,kl} & w_{X_2,kl} & \dots & w_{X_n,kl} \end{bmatrix}$ стеганоповідомлення.

Крок 4. Знайти середні значення $\mathrm{E}[\bar{u}_{kl}]$.

Крок 5. Провести статистичне оброблення отриманих значень на всіх задіяних в експерименті зображеннях, побудувати теплову карту зсувів середніх значень $\mathrm{E}[\bar{u}_{kl}]$ від теоретично розрахованого значення для оригінальних контейнерів $\mathrm{E}[u_{kl}]=0$.

Отримана в результаті виконання зазначеного експерименту теплова карта показана на рис. 2.5.

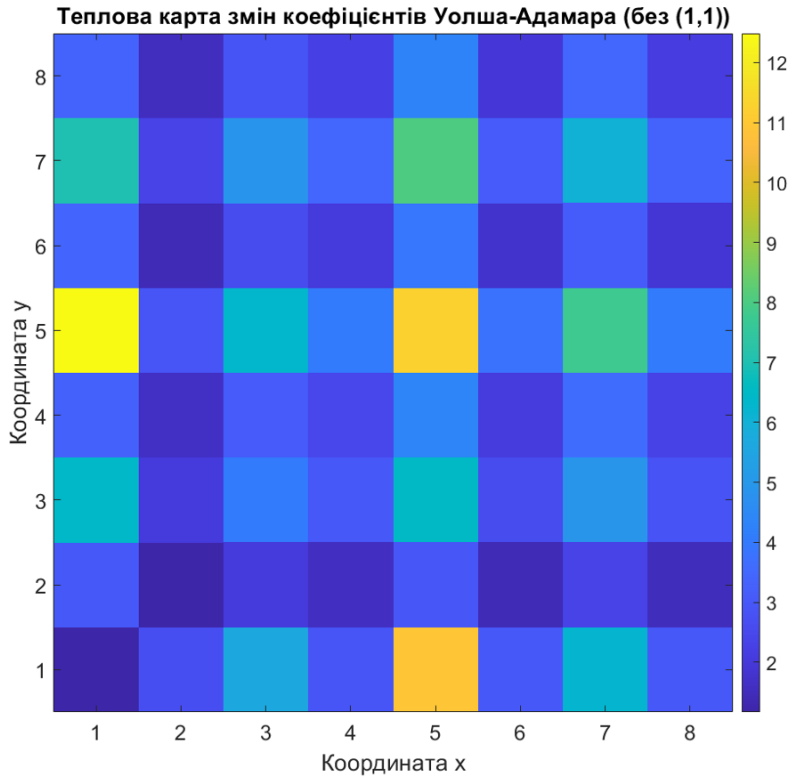


Рисунок 2.5 – Теплова карта зсувів гістограм трансформант перетворення Уолша-Адамара

На побудованій на рис. 2.5 тепловій карті відображено середні зміни трансформант перетворення Уолша-Адамара після обробки зображень у рамках стеганографічного методу. Аналіз показує, що найбільший зсув спостерігається для трансформант із координатами (1,5), (5,1) та (5,5). Ці позиції відповідають низькочастотним коефіцієнтам, які зберігають загальну структуру зображення [165]. Відомо [162, 163], що саме в такі трансформанти доцільно виконувати вбудовування додаткової інформації у стеганографічному

методі з кодовим управлінням, оскільки це дозволяє підвищити прихованість та стійкість методу при збереженні високої якості зображення.

Водночас ці самі трансформанти стають ключовими для аналізу у разі сліпого стеганоаналізу та підозри на вбудовування прихованих даних: замість дослідження всіх 64 коефіцієнтів достатньо проаналізувати лише три, що значно скорочує обчислювальні витрати та спрощує процес виявлення до 21 разу.

Висновки до розділу 2

1. Запропоновано теоретичні основи методу стеганоаналізу, що базується на виявленні порушення одномодальності розподілу трансформант перетворення Уолша-Адамара у зображеннях, до яких застосовано стеганографічне вбудовування з кодовим управлінням. Показано, що така ознака дозволяє формалізувати критерій наявності прихованої інформації, який може бути використаний для автоматизованого виявлення прихованих каналів у вебзастосунках. Отримані аналітичні твердження підтверджено обчислювальними експериментами, які демонструють високу чутливість гістограм трансформант до факту вбудовування. Зокрема встановлено, що селективне вбудовування призводить до формування двомодальних розподілів, що є надійною ознакою прихованого впливу. Запропонований підхід має практичне значення для систем захисту вебзастосунків, у яких дозволено завантаження зображень користувачами. Теоретична база, розроблена в цьому розділі, буде використана як основа для навчання моделей штучного інтелекту, здатних виявляти нетипові патерни у трансформантах зображень, які сигналізують про наявність прихованих повідомлень. Інтеграція такого аналізу у процеси CI/CD або моніторинг контенту дозволить підвищити рівень інформаційної безпеки веборієнтованих систем, доповнюючи класичні методи пошуку вразливостей механізмами контролю прихованих каналів передавання даних.

2. Проведене дослідження встановило ключові закономірності впливу стеганографічних перетворень у просторі сингулярного розкладу на коефіцієнти перетворення Уолша-Адамара. Отримані результати мають фундаментальне значення для розвитку методів стеганоаналізу, зокрема у двох основних напрямках:

2.1. для несліпого аналізу (з доступом до оригінального контейнера): встановлено чотири ключові трансформанти (8,8), (1,8), (8,1) та (1,1), які демонструють максимальну амплітуду змін; розроблено математичний апарат порівняння пар зображень, що дозволяє виявляти факт вбудовування з високою точністю; показано, що найбільш інформативними є кутові коефіцієнти перетворення, які відповідають високочастотним компонентам;

2.2. для сліпого аналізу (без доступу до оригіналу): виявлено три критично важливі низькочастотні трансформанти (1,5), (5,1) та (5,5), які демонструють найбільший статистичний зсув; запропоновано новий підхід до аналізу на основі дослідження змін у розподілі коефіцієнтів; доведено, що систематичні зміни в цих позиціях є надійним маркером наявності прихованих повідомлень.

3. Розкрито механізми взаємодії між сингулярним розкладанням і перетворенням Уолша-Адамара при стеганоперетворенні, обґрунтовано вибір оптимальних трансформант для аналізу, що є основою для розробки нових методів виявлення стеганографічних втручань. Досягнуті результати дозволяють говорити про можливе зниження обчислювальної складності аналізу в 16-21 разів, розроблені підходи адаптовані для реалізації у вебсередовищі з обмеженими ресурсами, отримані результати відкривають нові можливості для створення ефективних систем захисту.

РОЗДІЛ 3. РОЗРОБЛЕННЯ МЕТОДІВ НА ОСНОВІ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ЗАХИСТУ ВЕБЗАСТОСУНКІВ ВІД ПРИХОВАНИХ КАНАЛІВ

3.1. Комплексний підхід до інтелектуального стеганоаналізу вебзастосунків

У попередньому розділі розроблено математичні засади виявлення прихованих каналів на основі перетворення Уолша-Адамара. Цей апарат дав змогу формалізувати процес дослідження прихованих інформаційних потоків, визначити їхні характерні ознаки та створити теоретичний базис для подальшої побудови методів стеганоаналізу. Здобуті результати забезпечують можливість переходу від суто математичних моделей до практично орієнтованих рішень, здатних ефективно виявляти приховані канали у складних інформаційних системах.

Сформовані засади є основою для створення сучасних, потужних методів виявлення прихованих каналів у вебзастосунках, де традиційні підходи часто виявляються недостатніми через динамічність середовища, варіативність протоколів і багаторівневу структуру даних. Використання перетворення Уолша-Адамара дозволяє виявляти приховані закономірності у трафіку та підвищувати точність класифікації потенційно небезпечних потоків.

Базисом для створення сучасних методів виявлення прихованих каналів у вебзастосунках є технології ШІ. Їх використання зумовлене тим, що приховані канали мають різноманітну природу, можуть проявлятися у вигляді нетипових закономірностей у структурі мультимедійних даних, їх статистичних та інформаційних характеристиках. Традиційні статистичні чи евристичні методи часто виявляються недостатньо ефективними для їх надійного виявлення, особливо у разі високої варіативності середовища та адаптивності зловмисників.

Методи ШІ дозволяють врахувати цю багатогранність. Зокрема, нейронні мережі здатні автоматично формувати багаторівневі представлення даних та виокремлювати приховані ознаки, які неможливо виявити класичними аналітичними інструментами. Це особливо важливо для аналізу вебзастосунків, де інформація передається у різних форматах: графічному, аудіо чи відео. Завдяки цьому ШІ ідеально підходить для розв'язання завдання виявлення прихованих каналів, оскільки забезпечує універсальність, адаптивність і високу чутливість до аномальних відхилень у даних.

Таким чином, поєднання математичних засад, побудованих на основі перетворення Уолша-Адамара, з технологіями ШІ створює потужний фундамент для побудови нових підходів до стеганоаналізу, орієнтованих на роботу в складних умовах сучасного вебсередовища.

Завдання цього розділу:

1. розробка методів виявлення прихованих каналів на основі кодового управління у вебзастосунках;

2. розробка методів виявлення прихованих каналів на основі SVD UV-методів у вебзастосунках;

3. оцінка ефективності розроблених методів та їх порівняння з наявними аналогами.

3.2. Основні засади застосування штучного інтелекту в стеганографії

Розвиток ідей штучних нейронних мереж (ШНМ) розпочався у середині XX століття. Перші концептуальні моделі штучного нейрона були запропоновані В. МакКаллоком та В. Піттсом (1943 р.), які описали математичну модель біологічного нейрона як елемента, здатного виконувати логічні операції. Подальший прорив відбувся у 1958 р., коли Ф. Розенблатт запропонував перцептрон (рис. 3.1.) [176] – першу практичну модель нейронної мережі, яка навчалася на основі корекції вагових коефіцієнтів.

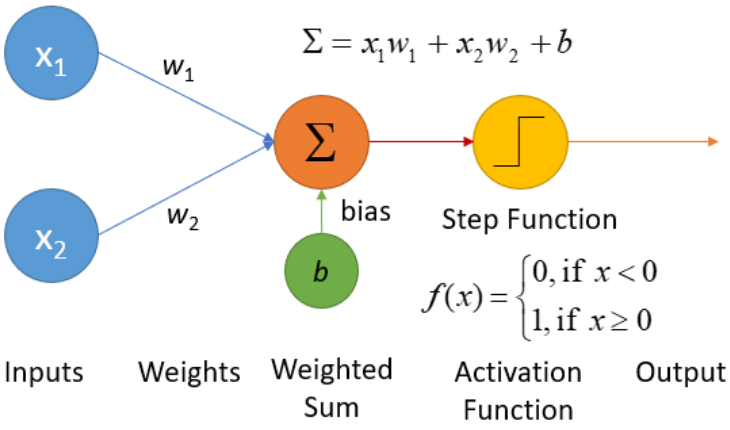


Рисунок 3.1 – Перцептрон

Перцептрон мав обмежені можливості (зокрема, він не міг розв'язувати задачі, які не є лінійно роздільними), однак саме він започаткував епоху досліджень у цій сфері.

У 1970–1980-х роках розвиток ШНМ був уповільненим через так звану «зиму штучного інтелекту». Лише поява алгоритму зворотного поширення помилки (backpropagation) та зростання обчислювальних потужностей у 1980-х роках дозволили перейти до навчання багатошарових нейронних мереж. Це стало відправною точкою для формування сучасного напряму глибокого навчання (deep learning). У 2000–2010-х роках завдяки широкому впровадженню графічних процесорів (GPU) та великих масивів даних ШНМ почали активно застосовуватися у розпізнаванні образів, обробленні природної мови, прогнозуванні часових рядів та інших складних задачах.

Базовим елементом будь-якої штучної нейронної мережі є штучний нейрон, який моделює роботу біологічного нейрона. Кожен нейрон отримує на вході вектор сигналів, які зважуються відповідними коефіцієнтами (вагами). Далі обчислюється зважена сума сигналів, до якої додається поріг (зміщення). Отримане значення подається на активаційну функцію, яка визначає вихід нейрона. Активаційні функції можуть бути різних типів: порогові, сигмоїдальні, гіперболічний тангенс, ReLU та інші. Вибір функції визначає здатність мережі моделювати нелінійні залежності.

Процес навчання нейронної мережі полягає у корекції вагових коефіцієнтів на основі порівняння очікуваного та фактичного виходу мережі. У випадку навчання з учителем для цього використовується функція втрат (loss function), яка відображає різницю між бажаним та отриманим результатом. Алгоритм зворотного поширення помилки дозволяє ефективно знаходити оптимальні значення ваг, використовуючи градієнтні методи оптимізації.

Таким чином, штучні нейронні мережі поєднують у собі ідеї моделювання біологічних процесів і сучасні математичні методи оптимізації, що робить їх універсальним інструментом для обробки даних складної структури.

Штучні нейронні мережі (ШНМ) мають широкий спектр архітектурних рішень і методів навчання, що визначає їх ефективність у різних прикладних задачах. Для систематизації підходів їх класифікують за архітектурою, методом навчання та сферою застосування.

1. Класифікація за архітектурою:

– *одношарові мережі* – найпростіший тип мереж, у якому всі вхідні дані напряму з'єднані з виходом. Вони здатні розв'язувати лише лінійно роздільні задачі;

– *багатошарові мережі* (MLP, Multilayer Perceptron) складаються з кількох шарів нейронів (вхідного, прихованих і вихідного). Завдяки прихованим шарам такі мережі можуть апроксимувати довільні нелінійні функції. Схема багатошарової нейронної мережі показана на рис. 3.2;

– *згорткові нейронні мережі* (CNN, Convolutional Neural Networks) – спеціалізовані архітектури, які використовують згорткові та підвибіркові (pooling) шари. Вони ефективні для обробки структурованих даних, зокрема зображень і сигналів, де важливі локальні ознаки. Схема згорткової нейронної мережі показана на рис. 3.3 [125];

– *рекурентні нейронні мережі* (RNN, Recurrent Neural Networks) – мережі, в яких є зворотні зв'язки. Вони зберігають інформацію про попередні

стани, що робить їх придатними для роботи з часовими рядами та послідовними даними;

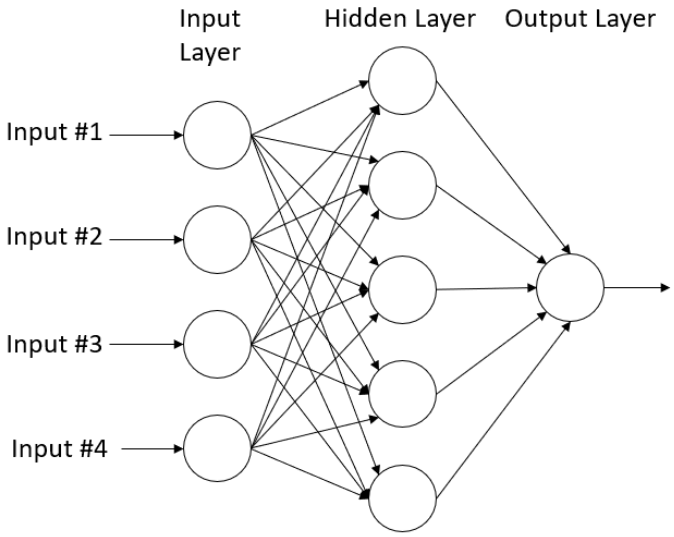


Рисунок 3.2. – Багатошаровий перцептрон

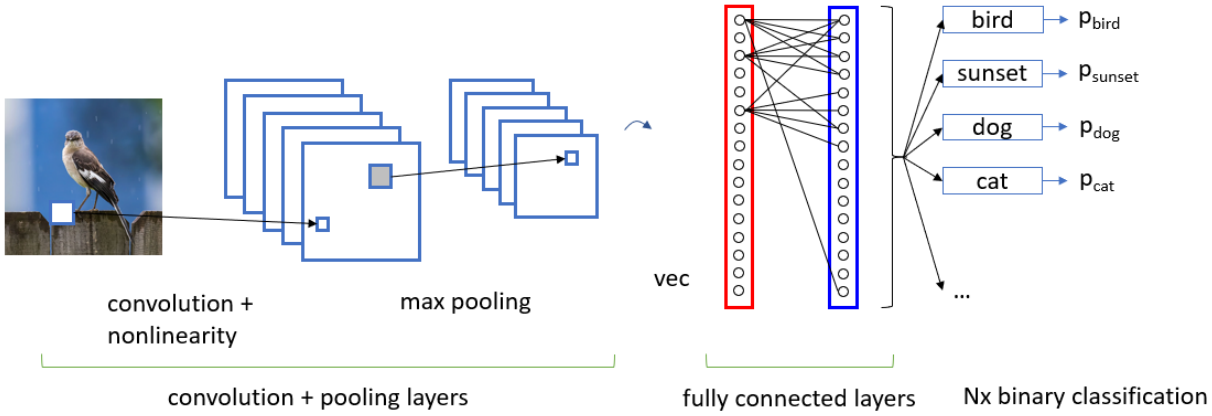


Рисунок 3.3. – Згорткова нейронна мережа

– *автоенкодери* (Autoencoders) – мережі, які навчаються стискати вхідні дані у компактне латентне подання та відновлювати їх. Використовуються для зменшення розмірності, виявлення аномалій і попередньої обробки даних;

– *генеративно-змагальні мережі* (GAN, Generative Adversarial Networks) – архітектура, що складається з двох мереж (генератора і

дискримінатора), які навчаються у змагальному режимі. Використовуються для генерації зображень, синтетичних даних та моделювання складних розподілів.

2. Класифікація за методом навчання:

– *навчання з учителем* (supervised learning) – модель навчається на основі пар «вхід – правильний вихід», що дозволяє будувати точні класифікаційні та регресійні алгоритми;

– *навчання без учителя* (unsupervised learning) – використовується для роботи з даними без міток. Мережа виявляє приховані структури (кластеризація, виявлення аномалій, зменшення розмірності);

– *навчання з підкріпленням* (reinforcement learning) – мережа набуває знань шляхом взаємодії з середовищем й отримання винагороди чи покарання за свої дії. Ефективна для оптимізації процесів прийняття рішень у динамічних умовах.

3. Класифікація за сферою застосування:

– *розпізнавання образів* – обробка зображень і відео, ідентифікація об’єктів, виявлення патернів;

– *обробка тексту* (NLP, Natural Language Processing) – аналіз природної мови, машинний переклад, чат-боти, системи пошуку інформації;

– *прогнозування часових рядів* – передбачення фінансових показників, трафіку, кліматичних змін, параметрів у телекомунікаційних мережах;

– *класифікація мережевого трафіку та виявлення аномалій* – застосування у сфері інформаційної безпеки для аналізу комунікацій, виявлення прихованих каналів і потенційних атак.

Таким чином, розмаїття архітектур та методів навчання дозволяє штучним нейронним мережам адаптуватися до широкого кола прикладних завдань. Далі у дослідженні основну увагу буде зосереджено на багатошарових перцептронах (MLP) та згорткових нейронних мережах (CNN), які найкраще відповідають завданням виявлення прихованих каналів. Їхні характеристики

роблять ці моделі найбільш придатними для обробки даних різної природи: від табличних і числових структур до зображень та мережевого трафіку.

MLP є класичною архітектурою штучних нейронних мереж, яка складається з вхідного шару, одного або декількох прихованих шарів і вихідного шару. Кожен нейрон у мережі з'єднаний з усіма нейронами попереднього шару, що забезпечує універсальність і здатність апроксимувати довільні нелінійні функції.

Основною перевагою MLP є його ефективність при роботі з табличними та числовими даними, де ознаки представлені у вигляді векторів. Саме така структура часто зустрічається у завданнях аналізу мережевого трафіку, обробки статистичних характеристик сигналів або векторів ознак, отриманих на основі математичних перетворень (зокрема, перетворення Уолша-Адамара). MLP добре підходить для задач класифікації та регресії, де важливо встановити складні нелінійні залежності між вхідними параметрами і вихідним результатом.

CNN є спеціалізованою архітектурою, орієнтованою на роботу з багатовимірними структурованими даними, зокрема із зображеннями та сигналами. На відміну від MLP, CNN використовують згорткові шари, які виділяють локальні ознаки, враховують просторову або часову близькість елементів даних. Це дозволяє моделі ефективно працювати з інформацією, де важливі локальні патерни, наприклад, текстури у зображеннях, характерні коливання у сигналах чи повторювані структури в мережевому трафіку.

Крім того, CNN застосовують пулінгові шари (subsampling), які зменшують розмірність даних і водночас зберігають найбільш значущі ознаки. Завдяки цьому CNN виявляють стійкі характеристики навіть у зашумлених даних і демонструють високу продуктивність у задачах класифікації образів, детекції аномалій та обробки часових послідовностей.

MLP оптимально застосовувати тоді, коли дані вже подані у вигляді векторів ознак і не потребують додаткового виділення локальних структур.

CNN ефективні, коли дані мають просторову чи часову організацію і важливим є збереження локальних залежностей.

Таким чином, поєднання MLP та CNN у єдиному підході до виявлення прихованих каналів забезпечує комплексний аналіз даних різної природи. MLP надає можливість обробляти узагальнені числові характеристики, тоді як CNN дозволяє виявляти приховані закономірності у складних структурах даних, що підвищує загальну ефективність стеганоаналітичних методів.

Розробка ефективних методів виявлення прихованих каналів потребує обґрунтованого вибору архітектури нейронної мережі, яка найкраще відповідає специфіці даних та особливостям поставленого завдання. Вбачається доцільним використання двох основних архітектур – багатошарового перцептрона (MLP) та згорткової нейронної мережі (CNN).

Перетворення Уолша-Адамара, розглянуте у попередньому розділі, дає змогу отримати числові вектори ознак, які відображають характерні закономірності у даних. Такі вектори є добре структурованими та придатними для аналізу у форматі «вхід–вихід», де потрібно встановити залежність між числовими параметрами та наявністю або відсутністю прихованого каналу.

Багатошаровий перцептрон є оптимальним вибором для обробки таких даних, оскільки:

– здатний апроксимувати складні нелінійні функції;

– ефективно працює з табличними та векторними ознаками;

– дозволяє будувати моделі класифікації, які базуються на числових характеристиках, отриманих у результаті математичних перетворень.

Таким чином, застосування MLP забезпечує коректний і гнучкий аналіз ознак, сформованих на основі перетворення Уолша-Адамара, що дозволяє виявляти навіть слабкі прояви прихованих каналів.

На відміну від MLP, згорткові нейронні мережі орієнтовані на роботу з багатовимірними та структурованими даними, такими як зображення, сигнали або мережевий трафік. У завданні виявлення прихованих каналів важливу роль

відіграють локальні патерни, наприклад, періодичні повтори у трафіку, характерні локальні зміни у зображеннях або особливості структур у послідовностях даних.

CNN демонструють високу ефективність у таких умовах завдяки:

– використанню згорткових шарів, які автоматично виділяють локальні ознаки, стійкі до шуму та спотворень;

– здатності узагальнювати інформацію за рахунок багаторівневого подання даних;

– застосуванню пулінгу, що зменшує розмірність і водночас зберігає найважливіші характеристики.

Завдяки цим особливостям CNN придатні для аналізу складних структур даних, у яких приховані канали можуть маскуватися під природні патерни. Це робить їх потужним інструментом у поєднанні з класичними методами обробки сигналів.

Використання MLP та CNN у комплексі забезпечує універсальний підхід до виявлення прихованих каналів. MLP дозволяє ефективно обробляти числові вектори ознак, тоді як CNN забезпечує виявлення прихованих закономірностей у багатовимірних структурах. Поєднання цих архітектур створює збалансовану основу для побудови сучасних стеганоаналітичних методів, що враховують як глобальні, так і локальні характеристики даних.

3.3. Загальні засади стеганоаналізу стеганографічного методу з кодовим управлінням

Однією з ключових проблем є виявлення можливих прихованих каналів передачі інформації, які можуть використовуватися зловмисниками для несанкціонованого обміну даними або приховування конфіденційної інформації. Контроль мультимедійної інформації в цьому контексті стає невіддільною складовою безпеки вебзастосунків, оскільки дозволяє запобігати потенційним загрозам, пов’язаним із вбудовуванням стеганографічних

повідомлень у зображення, аудіо- та відеофайли. Розробка ефективних методів аналізу та виявлення прихованих даних забезпечує надійність вебплатформ, підвищує довіру користувачів і захищає їхню інформацію від можливих зловживань.

Методи стеганографії активно розвиваються у вебзастосунках, і хоча вони здатні забезпечувати нові можливості для захисту та конфіденційності даних, водночас це створює передумови для їхнього використання у зловмисних цілях. Зокрема, сучасні підходи відкривають додаткові канали несанкціонованої передачі прихованої інформації, що ускладнює виявлення витоку даних та посилює загрози кібербезпеці.

У дослідженні [177] представлено вебзастосунок FHWA, який дозволяє приховувати будь-які типи файлів у зображеннях за допомогою стеганографії. Якщо для користувачів це розширює функціональні можливості, усуваючи обмеження наявних систем, які підтримують лише окремі формати файлів, то з точки зору кібербезпеки така універсальність створює нові ризики. FHWA підтримує приховування текстових документів, зображень, аудіо, відео та інших типів файлів, що робить його не лише зручним інструментом для легальних задач, а й потенційним засобом для кіберзлочинців. Додаткове інтегрування механізму шифрування з паролем підсилює захищеність прихованої інформації, але водночас ускладнює її виявлення та перехоплення. Згідно з результатами тестування 98% користувачів успішно використовували FHWA для приховування та відновлення файлів, що свідчить про його практичну ефективність, але також демонструє можливості для зловмисного використання у прихованих каналах передачі даних.

У роботі [178] запропоновано гібридний підхід до приховування даних у вебзастосунках, який поєднує стеганографію та візуальну криптографію. Цей метод дозволяє приховувати інформацію в зображеннях у такий спосіб, що її можна відновити лише за допомогою спеціального ключа, що підвищує рівень безпеки від несанкціонованого доступу. Використання візуальної криптографії забезпечує додатковий рівень захисту, дозволяє відновити оригінальні дані

лише при поєднанні певних частин зображень. Цей підхід є перспективним для застосування в системах, де необхідний високий ступінь конфіденційності та захисту даних.

Методи цифрового стеганоаналізу, які дозволяють виявляти приховану інформацію в мультимедійних об’єктах, також активно розвиваються. Зокрема, сучасні підходи часто базуються на глибокому навчанні та складних нейронних мережах, що забезпечує високу точність детектування навіть при використанні складних стеганографічних методів. Наприклад, дослідження [179] систематизує різноманітні методи стеганоаналізу на основі глибокого навчання, класифікує їх за типами даних, описує труднощі та перспективи розвитку, підкреслюючи високі обчислювальні вимоги сучасних моделей. Аналогічно, робота [180] пропонує комплексний огляд методів стеганоаналізу зображень із застосуванням глибокого навчання та виділяє основні напрямки, які потребують значних ресурсів для обробки великих масивів даних.

У статтях [181, 182] наголошено на сучасних методах аналізу цифрових зображень і відкритих проблемах, серед яких – потреба у високопродуктивних обчислювальних ресурсах, що обмежує можливість застосування цих методів у реальному часі на вебплатформах. Водночас автори роботи [183] пропонують більш легкі моделі для стеганоаналізу зображень на основі глибокого навчання, які демонструють високу ефективність при значно менших витратах ресурсів, що робить їх більш придатними для вебзастосунків.

Дослідження [170, 184, 185] також демонструють широкий спектр методів: від використання текстурних характеристик до детекції країв за допомогою моделей глибокого навчання, які дозволяють підвищити точність детектування. Проте всі ці підходи залишаються часто громіздкими та ресурсоємними, що обмежує їх використання у вебзастосунках із динамічною обробкою мультимедійної інформації.

Таким чином, хоча цифровий стеганоаналіз швидко розвивається та демонструє високі результати, для інтеграції у вебзастосунки все ще

актуальним є пошук легких, ресурсоефективних рішень, які поєднуватимуть точність детектування та продуктивність у реальному часі.

Проте особливо стійким виявляється метод із кодовим управлінням [162], і на сьогодні практично відсутні ефективні методи його стеганоаналізу. У роботі [20] було створено теоретичну базу для побудови такого стеганоаналізу, проте сам метод поки що не розроблений. У цьому розділі пропонується розробка двох ефективних методів стеганоаналізу прихованих повідомлень із кодовим управлінням на основі застосування області перетворень Уолша-Адамара та сучасних методів штучного інтелекту.

Стеганографічний метод з кодовим управлінням характеризується високою стійкістю до виявлення та становить значну складність для наявних стеганоаналітичних підходів. У цьому методі для вбудовування додаткової інформації використовуються збалансовані функції Уолша з невеликою амплітудою впливу ±1 на кожний піксель контейнера, при цьому вбудовування додаткової інформації відбувається в адитивний спосіб, що також додає методу стійкості до сучасних методів стеганоаналізу. Дійсно, зазначений спосіб вбудовування практично не впливає на статистичні властивості зображення, що забезпечує мінімальне спотворення носія та високу стійкість методу до сучасних атак стеганоаналізу. Через це стандартні методи аналізу, засновані на простих ознаках або частотних характеристиках, демонструють низьку ефективність.

Проте у другому розділі було здійснено значний крок у напрямку стеганоаналізу цього методу. Запропоновано строгий математичний апарат, який дозволив виявити характерні зміни в гістограмі розподілу трансформант перетворення Уолша-Адамара, а саме – характеристику двомодальності.

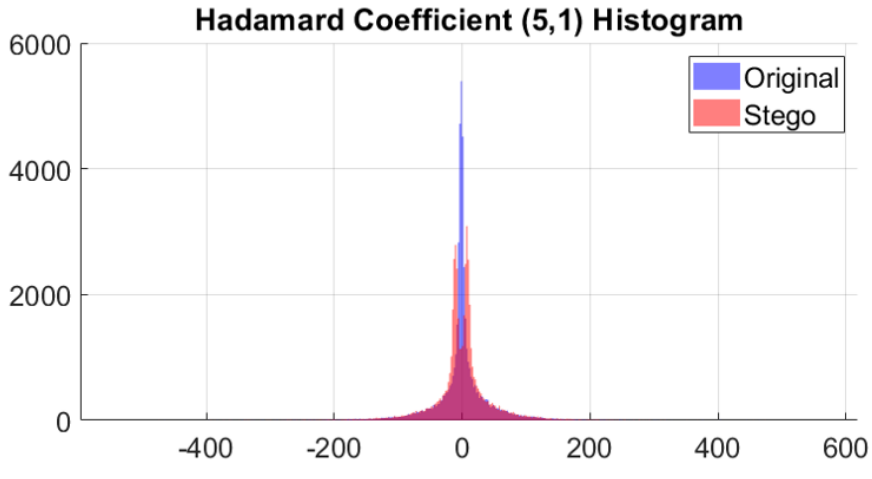
Згідно з результатами, поданими у другому розділі, якщо задано матриці W_{X_j} розміру $N \times N$ трансформант перетворення Уолша-Адамара n блоків $X_j, j = 1,2,...,n$ зображення, кожна з яких має форму (2.10), та якщо ми складемо гістограми для послідовностей трансформант

$u_{kl} = \left[w_{X_1,kl} \quad w_{X_2,kl} \quad \ldots \quad w_{X_n,kl} \right]$, в які здійснювалося вбудовування додаткової інформації із застосуванням стеганографічного методу з кодовим управлінням із кодовими словами на основі функцій Уолша, то вони матимуть двомодальний характер із максимумами в точках $\pm N^2$.

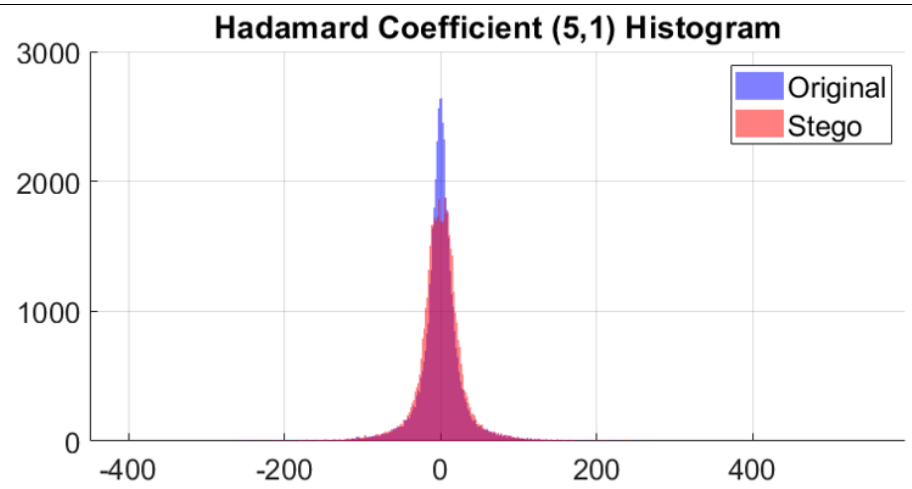
Попри отримані теоретичні результати, на практиці гістограми розподілу трансформант перетворення Уолша-Адамара виявляють значну різноманітність і виражену залежність від конкретного зображення. Така мінливість зумовлена як особливостями просторово-частотної структури вхідних даних, так і відмінностями у статистичних характеристиках зображень різних класів. Як наслідок, навіть за наявності наперед відомих інформативних ознак проведення ефективного стеганоаналізу суттєво ускладняється.

У табл. 3.1 показано приклади гістограм розподілу послідовностей трансформант (5,1) для декількох зображень.

Таблиця 3.1 – Гістограми розподілу послідовностей трансформант (5,1) перетворення Уолша-Адамара для оригінальних зображень та стеганоповідомлень

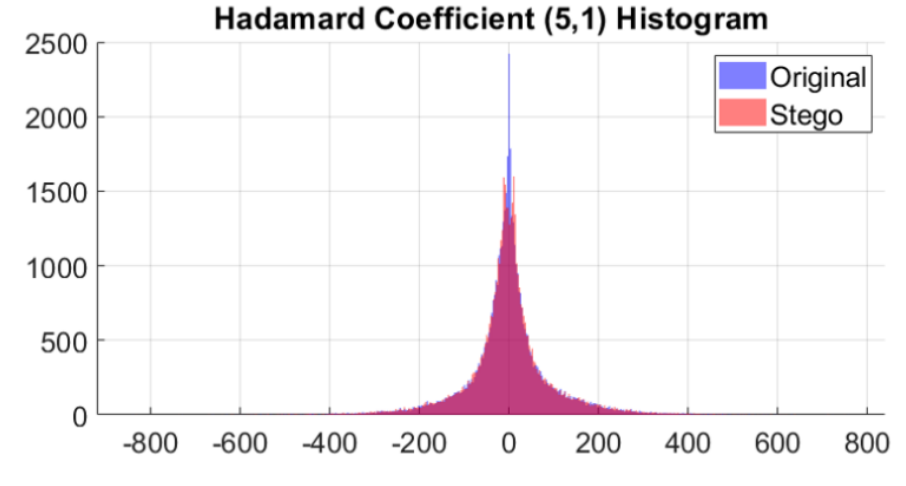


Зображення 1

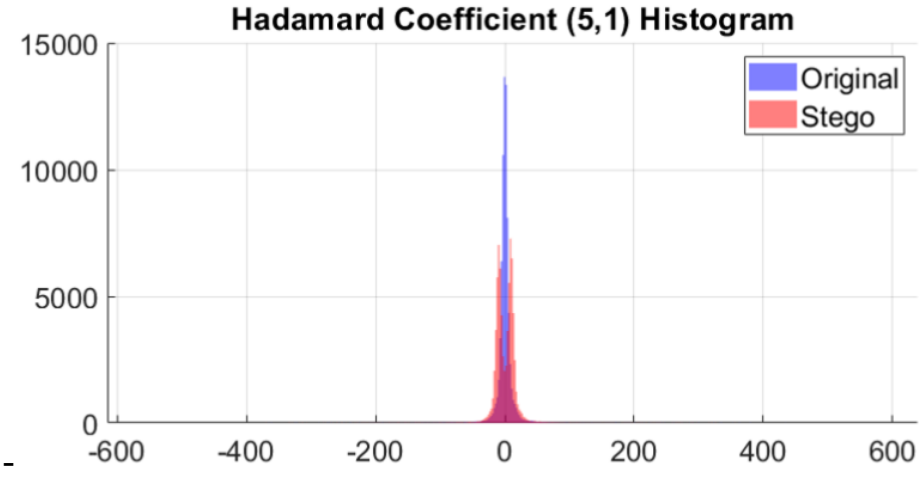


Зображення 2

Закінчення табл. 3.1



Зображення 3



Зображення 4

Аналіз даних у табл. 3.1 показує, що конкретний вид гістограми розподілу трансформант перетворення Уолша-Адамара значною мірою залежить від вибраного зображення та його внутрішніх характеристик. Така висока варіативність означає, що класичні підходи до стеганоаналізу, засновані лише на заздалегідь відомих ознаках, часто виявляються недостатньо ефективними.

Через це найбільш раціональним представляється застосування сучасних методів штучного інтелекту, здатних автоматично виділяти стійкі ознаки на тлі значних варіацій у даних. Ми пропонуємо два підходи:

1) метод, який базується на попередньому обробленні вхідних даних із використанням огинаючої («envelope») гістограми, що дозволяє виділити ключові пікоподібні ознаки і зменшити вплив дрібних коливань;

2) метод на основі ансамблю згорткових нейронних мереж (CNN), який безпосередньо працює з необробленими даними та здатен автоматично виявляти закономірності, характерні для стеганосигналів, навіть за умов значної варіативності гістограм.

Таке поєднання класичних і сучасних підходів забезпечує більш надійну й універсальну систему стеганоаналізу, орієнтовану на використання у вебзастосунках.

3.4. Стеганоаналітичний метод на основі обчислення огинаючої

Метод із використанням огинаючої заснований на виділенні глобальної структури сигналу або розподілу, ігноруючи локальні флуктуації. У нашому випадку огинаюча застосовується до гістограм розподілу трансформант, що дозволяє виявляти характерні піки, пов'язані з одномодальністю або двомодальністю даних. Далі ми застосовуємо таке визначення огинаючої.

Визначення 1. Для дискретного сигналу $x[n]$ огинаюча $e[n]$ може бути обчислена як верхня огинаюча, яка задається як:

$$e[n] = \sqrt{x[n]^2 + H\{x[n]\}^2}\,, \tag{3.1}$$

де $H\{x[n]\}$ – перетворення Гільберта сигналу $x[n]$, що забезпечує фазовий зсув на $\pi/2$ і дозволяє побудувати аналітичний сигнал. Такий підхід гарантує, що $e[n] \geq 0$ для всіх n і огинаюча «прямує» за амплітудними змінами сигналу.

Далі виконується обчислення піків огинаючої сигналу, в результаті чого формуються вектори, що містять кількість піків та їхнє положення. Отримані дані нормалізуються та подаються на вхід нейронної мережі. В якості класифікатора використовується багатошарова повнозв'язна нейронна мережа (Multilayer Perceptron, MLP) з такою специфікацією:

– вхідний шар: 3 нейрони, які відповідають розмірам векторів ознак (кількість піків та позиції двох перших піків);

– приховані шари: чотири шари з кількістю нейронів 400, 250, 100 та 50, відповідно; функція активації – ReLU (Rectified Linear Unit);

– вихідний шар: 2 нейрони, які відповідають класам одномодального та двомодального розподілу; функція активації – softmax для отримання ймовірностей класів;

– функція втрат: крос-ентропія (cross-entropy), яка дозволяє коректно навчати класифікацію двох класів;

– оптимізатор: алгоритм Adam з початковою швидкістю навчання 0.001.

Після навчання нейронна мережа прогнозує ймовірність приналежності нового зображення до кожного класу. Клас із максимальною ймовірністю визначається як результат класифікації, що дозволяє робити висновок про наявність або відсутність стеганосигналу. Така архітектура забезпечує здатність моделі автоматично виявляти ключові закономірності у гістограмах, навіть за високої варіативності вхідних даних.

Опишемо запропонований стеганоаналітичний метод за допомогою покрокового подання трьох основних операцій: підготовка вхідних даних для методу, навчання нейронної мережі, прийняття рішення щодо наявності додаткової інформації у заданому зображенні.

Метод М(1)1. Підготовка вхідних даних

1.1. Вхідне зображення F розбивається на 8×8-блоки X_i стандартним чином.

1.2. Для кожного блока будується двовимірне перетворення Уолша-Адамара W_{X_i} (2.1).

1.3. Для зображення для кожної з трансформант перетворення Уолша-Адамара (*k*,*l*) або для трансформанти, щодо якої є підозри наявності додаткової інформації, вбудованої із застосуванням стеганографічного методу з кодовим управлінням, будується послідовність $u_{kl} = \left[w_{X_1,kl} \quad w_{X_2,kl} \quad \ldots \quad w_{X_n,kl} \right]$.

1.4. Будуємо гістограму розподілу послідовності u_{kl}, що складатиметься з 500 рівномірно вибраних на інтервалі $[\min(u_{kl}), \max(u_{kl})]$ бінів.

1.5. Будуємо огинаючу (3.1) для гістограми, побудованої на кроці 1.4, знаходимо вектор, що міститиме такі ознаки: кількість піків та позиції двох перших піків отриманої огинаючої.

Метод М(1)2. Навчання нейронної мережі

2.1. Для навчання нейронної мережі формується вибірка, яка має 530 чистих зображень та 530 зображень зі стеганоповідомленнями. Для кожного зображення за допомогою методу М(1)1 виконується обчислення огинаючої гістограми та визначення ключових ознак, зокрема кількості піків і їхнього розташування. Отримані вектори ознак формують основу вхідних даних для нейронної мережі.

2.2. Кожен вектор ознак позначається як належний до класу: «1» – оригінальні (чисті) зображення; «2» – зображення зі стеганоповідомленням.

2.3. Вибірка розбивається на навчальну (80%) та тестову (20%) частини. Такий поділ дозволяє оцінювати здатність мережі узагальнювати закономірності на нових даних, не використаних під час навчання.

2.4. Проводимо навчання нейронної мережі: навчання мережі проводиться методом зворотного поширення помилки (backpropagation) з оптимізатором Adam. Вхідні вектори ознак нормалізуються для підвищення стабільності процесу навчання. Після завершення навчання мережа здатна прогнозувати ймовірність належності нового зображення до кожного класу, що дозволяє робити висновок про наявність або відсутність стеганоповідомлення.

Метод M(1)3. Прийняття рішення

3.1. На основі зображення, що поступає на вхід методу, будується вектор ознак на основі методу M(1)1.

3.2. Отриманий вектор ознак подається на вхід навченої багатошарової повнозв’язної нейронної мережі (MLP). Мережа обробляє вхідні дані, враховуючи взаємозв’язки між ознаками, і генерує ймовірності належності зображення до класів «чисте» або «зі стеганоповідомленням».

3.3. На основі вихідних ймовірностей нейронної мережі визначається клас з максимальною вірогідністю. Це дозволяє зробити однозначний висновок про наявність або відсутність додаткової інформації у зображенні. Таким чином, запропонований метод забезпечує автоматичну та надійну класифікацію вхідних зображень на основі статистичних ознак їх гістограм.

Для наочного подання розподілу ознак між класами зображень було застосовано метод t-Distributed Stochastic Neighbor Embedding (t-SNE). Цей метод дозволяє відобразити багатовимірні вектори ознак у двовимірному просторі, зберігаючи локальні структури даних та взаємне розташування точок. На рис. 3.4 показано результати проєкції векторів ознак одномодальних та двомодальних гістограм зображень, що дозволяє оцінити ступінь їх роздільності. Як видно, метод t-SNE наочно демонструє, що ознаки, отримані за допомогою обчислення піків огинаючої гістограми, забезпечують достатньо

інформативне розділення класів, що підтверджує ефективність застосування нейронної мережі для класифікації.

Бачимо, спостерігається чітке групування обох класів в окремі кластери, що свідчить про інформативність вибраних ознак та їхню здатність розрізняти вихідні зображення та стеганоповідомлення. Попри наявність зон часткового перекриття, що вказує на певну схожість ознак для окремих випадків, більшість точок зберігає належність до свого класу. Просторова форма кластерів має виражену нелінійну структуру, що підтверджує доцільність використання багатошарової нейронної мережі для моделювання складних залежностей між ознаками та підвищення точності класифікації.

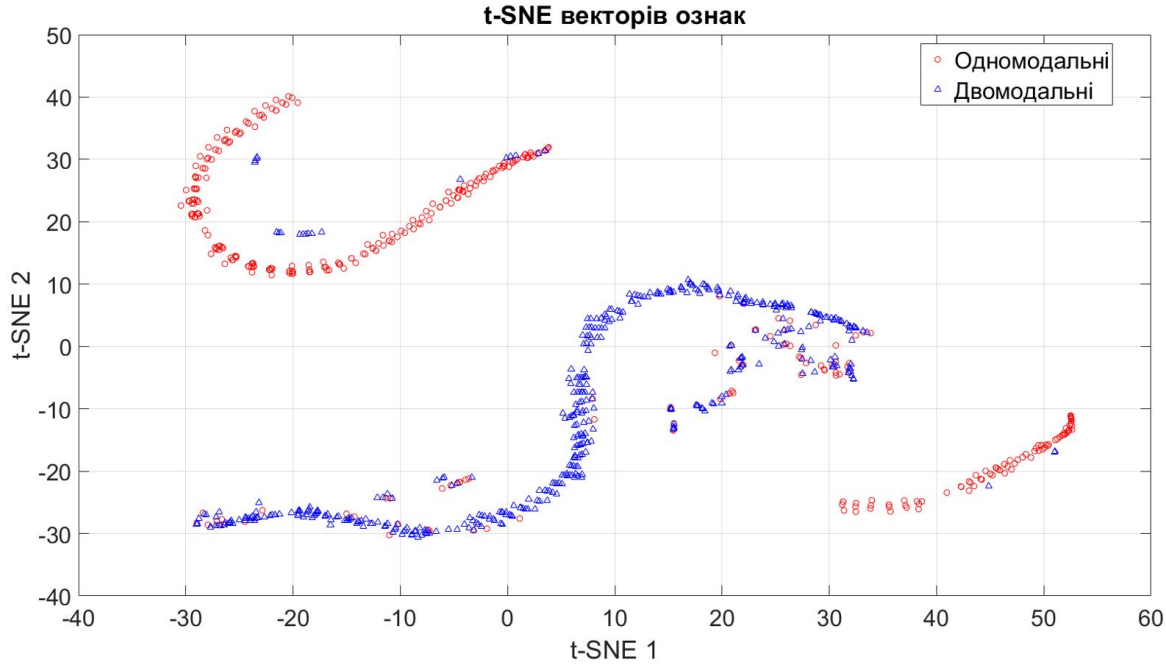


Рисунок 3.4 – t-NSE векторів ознак

Для кількісної оцінки ефективності розробленого методу виявлення стеганографічних повідомлень було проведено експериментальне тестування на контрольній вибірці даних. Для відображення результатів класифікації використано матрицю невідповідностей (confusion matrix) (рис. 3.5).

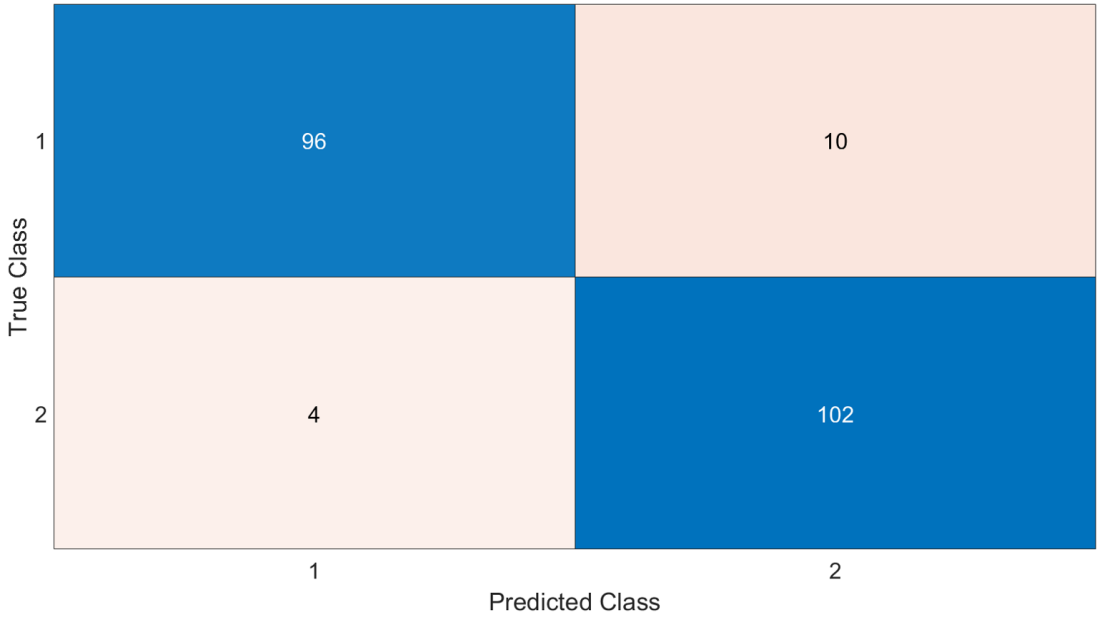


Рисунок 3.5 – Матриця невідповідностей для класифікації вибірки стеганоповідомлень та оригінальних зображень на основі методу з огинаючими

Рис. 3.5 дає змогу проаналізувати співвідношення між фактичними та передбаченими класами. По головній діагоналі матриці відображається кількість правильно класифікованих зразків, тоді як поза діагоналлю – кількість помилкових віднесень до інших класів. Такий підхід дозволяє комплексно оцінити чутливість (recall), точність (precision) та загальну точність (accuracy) алгоритму, а також виявити типові помилки класифікації

$$\begin{aligned} Precision &= \frac{TP}{TP+FP} = \frac{192}{192+10} \approx 0.95; \\ Recall &= \frac{TP}{TP+FN} = \frac{192}{192+4} \approx 0.98; \\ F1\text{-}score &= 2\frac{Precision \cdot Recall}{Precision + Recall} = \frac{0.95 \cdot 0.98}{0.95+0.98} \approx 0.96, \end{aligned} \tag{3.2}$$

де *TP* (True Positive) – кількість істинно позитивних рішень, *FP* (False Positive) – хибно позитивних, а *FN* (False Negative) – хибно негативних. Такий підхід дозволяє комплексно оцінити чутливість (recall), точність (precision) та

гармонійне середнє цих показників (F1-score), а також виявити типові помилки класифікації.

Отримані результати демонструють високу ефективність запропонованого методу виявлення стеганографічних повідомлень. Значення точності (Precision = 0.95) свідчить про те, що лише 5% зразків, віднесених алгоритмом до категорії «стеганоповідомлення», виявилися хибними. Водночас показник чутливості (Recall = 0.98) підтверджує здатність методу коректно виявляти переважну більшість вбудованих повідомлень, з мінімальною кількістю пропущених випадків. Обчислене гармонійне середнє цих показників (F1-score ≈ 0.96) узагальнено характеризує збалансованість методу між точністю та чутливістю, підтверджуючи його високу практичну придатність. Високе значення метрик у поєднанні з аналізом матриці невідповідностей свідчить про те, що запропонований підхід забезпечує надійне виявлення прихованих даних і може бути рекомендований для застосування у системах захисту інформації, де критично важливою є мінімізація помилкових рішень. При цьому загальна точність виявлення складає понад 93.3%.

Додатково було проведено оцінювання швидкодії методу. На комп’ютері з процесором AMD Ryzen 5 7500F (6 ядер), оперативною пам’яттю 32 ГБ та графічним адаптером NVIDIA GeForce RTX 3050 середній час обробки одного зображення розміру 3872x2592 для виділення ознак за допомогою запропонованого методу становить приблизно 0.45 секунди. Час класифікації одного зображення склав 0.0044 секунди. Це означає, що за наявності ознак система здатна класифікувати понад 200 зображень за секунду, що забезпечує можливість використання розробленого методу у системах виявлення стеганографічних повідомлень у режимі, наближеному до реального часу.

3.5. Стеганоаналітичний метод на основі ансамблю згорткових нейронних мереж

Попри отримані високі результати точності класифікації при використанні методу на основі аналізу огинаючої гістограм розподілу, цей підхід має низку обмежень. По-перше, він значною мірою залежить від попереднього етапу обробки сигналу та виділення ознак. Будь-які зміни у виборі параметрів, методів нормалізації чи способів оцінки огинаючої можуть призводити до відмінностей у якості класифікації, що знижує універсальність моделі. По-друге, такий підхід передбачає втручання дослідника у процес формування ознак, а отже, не є повністю автоматизованим. Це ускладнює масштабування методу на інші типи даних або задачі, де характерні ознаки апріорі невідомі.

Розробка методу, який ґрунтується на безпосередньому використанні нейронних мереж без попереднього формування огинаючої, дозволить усунути вищезазначені обмеження. Сучасні моделі глибокого навчання здатні самостійно навчатися ефективних представлень даних і виокремлювати суттєві закономірності, які можуть бути неочевидними або недосяжними для класичних методів аналізу ознак. Це створює передумови для підвищення загальної точності виявлення стеганографічних повідомлень та зменшення кількості хибних класифікацій.

Крім того, підхід із більшим застосуванням нейронних мереж та меншою попередньою обробкою забезпечує кращу узагальнювальну здатність, що особливо важливо при роботі з великими та різнорідними наборами даних. Він також відкриває можливість інтеграції уніфікованих архітектур (наприклад, ансамблів CNN або трансформерів), які довели свою ефективність у задачах комп’ютерного зору та аналізу сигналів. З практичного погляду, це дозволить створити метод, менш залежний від попередньої обробки, більш адаптивний до нових умов та здатний працювати в режимі реального часу.

Таким чином, розробка альтернативного методу на основі прямого використання нейронних мереж є логічним і перспективним напрямом дослідження, що дозволить як підвищити якість виявлення прихованих даних, так і забезпечити універсальність та масштабованість стеганографічного аналізу.

У цьому дослідженні для виявлення стеганографічних повідомлень використано ансамбль згорткових нейронних мереж (CNN), що дозволяє підвищити стійкість результатів класифікації та зменшити вплив випадкових похибок, характерних для окремих моделей. Ансамблевий підхід реалізовано шляхом побудови трьох різних архітектур CNN із варіацією розміру згорткових ядер та кількості фільтрів.

Кожна нейронна мережа має таку структуру:

– вхідний шар приймає нормалізовані гістограми з кількістю бінів $bins = 500$;

– перша згорткова операція виконується фільтрами з розмірами $[5\times1],[7\times1],[9\times1]$ для трьох моделей відповідно, що дає змогу аналізувати закономірності різної масштабності;

– нормалізація та ReLU-активація забезпечують стабілізацію навчання та врахування нелінійних залежностей;

– шари субдискретизації (max pooling) зменшують розмірність та виділяють найбільш значущі ознаки;

– друга згорткова операція із подвоєною кількістю фільтрів (128, 256, 384, відповідно) дозволяє глибше відобразити багаторівневу структуру даних;

– шар глобального усереднення (Global Average Pooling) виконує компактне агрегування просторової інформації;

– повнозв’язні шари (128 нейронів + Dropout 0.5) виконують класифікацію на вищому рівні абстракції, що підвищує узагальнювальну здатність;

– вихідний шар реалізує Softmax-класифікацію на два класи («одномодальні» та «двомодальні» розподіли).

Для навчання було використано алгоритм оптимізації Adam з початковою швидкістю навчання $5 \cdot 10^{-4}$, кількістю епох 50 та розміром мініпакета 32. Важливим компонентом є регуляризація Dropout, що знижує ризик перенавчання.

Запропонований стеганоаналітичний метод можна подати у вигляді послідовності трьох ключових етапів: 1) формування вхідних даних, 2) навчання нейронної мережі та 3) визначення наявності прихованої інформації в досліджуваному зображенні.

Метод М(2)1. Підготовка вхідних даних

Виконати кроки 1.1 – 1.4 Методу М(1)1.

Метод М(2)2. Навчання нейронної мережі

2.1. Для навчання нейронної мережі формується вибірка з 530 чистих зображень та 530 зображень зі стеганоповідомленнями. Для кожного зображення за допомогою методу М(2)1 виконується обчислення гістограм розподілення трансформант перетворення Уолша-Адамара, які подаються у вигляді нормалізованих векторів, що містять інформацію про $bins = 500$ бінів по яких розподілені значення трансформант.

2.2. Кожен вектор ознак позначається як належний до класу: «1» – оригінальні (чисті) зображення; «2» – зображення зі стеганоповідомленням.

2.3. Вибірка розбивається на навчальну (80%) та тестову (20%) частини.

2.4. Проводиться навчання нейронної мережі методом зворотного поширення помилки (backpropagation) з оптимізатором Adam.

Метод М(2)3. Прийняття рішення

3.1. На основі зображення, яке поступає на вхід, метод будує вектор, що містить відомості про розподілення значень заданої трансформанти перетворення Уолша-Адамара по $bins = 500$ бінам. Для покращення подання

даних гістограма подається у вигляді тензора розмірності $[bins \times 1 \times 1]$, придатного для оброблення згортковою нейронною мережею.

3.2. Вхідний тензор подається на ансамбль із трьох згорткових нейронних мереж (CNN). Кожна мережа незалежно обробляє дані, виділяє локальні та глобальні ознаки гістограми і генерує ймовірності належності зображення до класів «одномодальні» (чисте) або «двомодальні» (зі стеганоповідомленням).

3.3. Остаточне рішення приймається шляхом усереднення ймовірностей усіх трьох моделей ансамблю. Клас із максимальною середньою ймовірністю вибирається як остаточне рішення. Такий підхід дозволяє знизити вплив похибок окремих моделей та забезпечує більш надійне визначення наявності або відсутності прихованої інформації у зображенні.

Таким чином запропонований, метод забезпечує автоматичну, стійку та високоточну класифікацію вхідних зображень на основі статистичних ознак їхніх гістограм із використанням ансамблю нейронних мереж.

Аналогічно до опису методу, заснованого на аналізі огинаючих, для візуалізації ефективності класифікації на рис. 3.6 представлено матрицю невідповідностей (confusion matrix), яка відображає точність класифікації ансамблю нейронних мереж на тестовій вибірці.

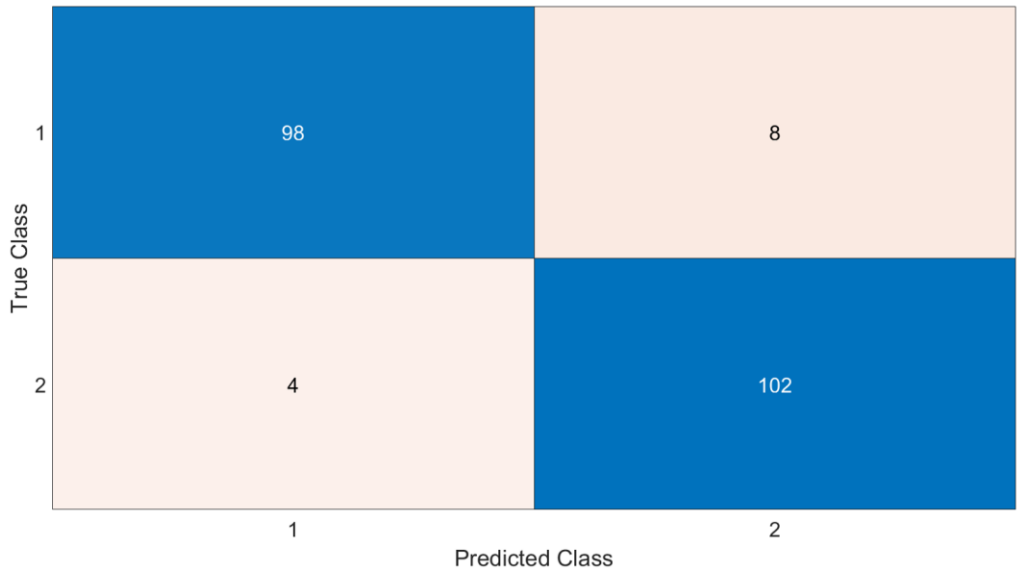


Рисунок 3.6 – Матриця невідповідностей для класифікації вибірки стеганоповідомлень та оригінальних зображень на основі методу на основі ансамблю згорткових нейронних мереж

Подібно до (3.2) на основі даних рис. 3.6 розрахуємо основні показники точності методу на основі ансамблю згорткових нейронних мереж

$$\begin{gathered} Precision = \frac{TP}{TP + FP} = \frac{200}{200 + 4} \approx 0.98; \\ Recall = \frac{TP}{TP + FN} = \frac{200}{200 + 8} \approx 0.96; \\ F1 - score = 2\frac{Precision \cdot Recall}{Precision + Recall} = 2 \cdot \frac{0.98 \cdot 0.96}{0.98 + 0.96} \approx 0.97. \end{gathered} \tag{3.3}$$

Отримані результати свідчать про дуже високу точність запропонованого методу: значення Precision досягло 0.98, Recall – 0.96, а F1-score становить приблизно 0.97. Варто зазначити, що ці показники навіть перевищують результати методу на основі огинаючих, що підтверджує високу ефективність вибраного підходу навіть без додаткової обробки даних. Такі високі показники свідчать про надійність та стабільність запропонованого методу для поставленого завдання. При цьому загальна точність виявлення складає понад 94.3%.

Відзначимо, що як час класифікації зображень нейронною мережею, так і час виділення ознак для методу на основі ансамблю згорткових нейронних мереж є фактично таким самим, що і для методу на основі обчислення огинаючої.

3.6. Порівняння запропонованих методів виявлення стеганоповідомлень із кодовим управлінням з наявними аналогами

Для забезпечення коректної оцінки ефективності запропонованих підходів до стеганоаналізу проведено порівняння з низкою відомих методів,

які широко застосовуються у практиці дослідження прихованих повідомлень. До таких належать класичні статистичні підходи – RS-аналіз, χ^2-метод та аналіз пар, які забезпечують базову перевірку ймовірності наявності прихованих даних у цифрових контейнерах. Додатково враховано сучасний метод стеганоаналізу кодового управління на основі збурень трансформант Уолша-Адамара, що характеризується високою чутливістю до структурних змін. Для комплексного зіставлення також використано інтегровану утиліту StegExpose, яка поєднує декілька алгоритмів виявлення та є практичним інструментом для реальних сценаріїв аналізу. Таке багатовекторне порівняння дозволяє всебічно оцінити переваги й обмеження розроблених методів у співвідношенні з усталеними аналогами.

У табл. 3.2 наведено результати проведених експериментів.

Таблиця 3.2 – Результати порівняльного аналізу

Метод	Точність
Стеганоаналітичний метод на основі обчислення огинаючої	93.3%
Стеганоаналітичний метод на основі ансамблю згорткових нейронних мереж	94.3%

Закінчення табл. 3.2.

Метод на основі аналізу трансформант перетворення Уолша-Адамара [143]	87.7%

RS-аналіз [186]	При проведенні RS-аналізу на вибірці з 530 зображень з повним (100%) заповненням пікселів встановлено: – при порозі 5% (значення >0.05) ознаки вбудованої інформації зафіксовано у 50 випадках, що становить 9.43% від загальної кількості зображень; – при порозі 10% (значення >0.10) впевнені ознаки виявили у 18 випадках, що становить 3.4%.
Аналіз пар [187]	Не детектує вбудовування із застосуванням кодового управління
χ^2-аналіз [187,188]	При проведенні χ^2-аналізу на вибірці з 530 зображень з повним (100%) заповненням пікселів було встановлено, що ознаки вбудованої інформації зафіксовано в 1 випадку, що становить 0.18% від загальної кількості зображень;
Утиліта StegExpose [188]	Не детектує вбудовування із застосуванням кодового управління

Для забезпечення коректності порівняння всі методи (табл. 2), включно із запропонованими автором, тестувалися на одному й тому самому наборі даних за однакових умов експерименту (однаковий розмір вибірки, параметри вбудовування, критерії оцінювання точності).

3.7. Метод виявлення стеганографічних повідомлень на основі SVD UV-методу

Наведені у другому розділі результати становлять основу для виявлення стеганоповідомлень, створених із застосуванням сучасних стійких стеганографічних методів, зокрема SVD UV-методів. Теоретичною базою для їх виявлення є аналіз зсувів у гістограмах трансформант перетворення Уолша-

Адамара, що відображають особливості змін у коефіцієнтах після прихованого вбудовування. У другому розділі було показано, що найбільший зсув спостерігається для трансформант із координатами (1,5), (5,1) та (5,5) [19, 189]. З погляду на це, запропонований метод ґрунтуватиметься на комплексному аналізі всіх зазначених трансформант із використанням класифікатора типу SVM, що дозволить підвищити точність класифікації та забезпечити надійне виявлення прихованих каналів.

Для стеганоаналізу даних, прихованих за допомогою методів на основі сингулярних векторів (U та V-матриць SVD), у цій роботі вибрано метод опорних векторів. Такий вибір зумовлений низкою причин. По-перше, SVM має високу ефективність у задачах бінарної класифікації, зокрема в умовах, коли кількість ознак значно перевищує кількість навчальних прикладів, що характерно для задач стеганоаналізу [190, 191]. По-друге, алгоритм SVM здатний будувати нелінійні розділяючі гіперплощини завдяки використанню ядрових функцій, що дозволяє виявляти складні залежності у високовимірному просторі ознак, сформованих на основі статистик SVD [192, 193]. По-третє, численні дослідження доводять: SVM характеризується високою стійкістю до перенавчання при коректному виборі параметрів, що є критично важливим при аналізі слабко виражених стеганографічних ознак [194, 195]. Таким чином, використання SVM забезпечує оптимальне поєднання узагальнювальної здатності та точності, що робить цей метод доцільним вибором для даної задачі.

Метод опорних векторів (Support Vector Machine, SVM) належить до класу потужних алгоритмів машинного навчання, які використовуються для задач класифікації, регресії та розпізнавання образів. Його ключова ідея полягає у знаходженні оптимальної роздільної гіперплощини, що максимально відокремлює об’єкти різних класів у просторі ознак.

Нехай задано навчальну вибірку

$$\{(x_i, y_i)\}_{i=1}^{N}, x_i \in R^n, y_i \in \{-1, +1\}, \tag{3.4}$$

де x_i – вектор ознак i -го об'єкта, а y_i – відповідна мітка класу. SVM шукає гіперплощину у вигляді

$$w^T x + b = 0, \tag{3.5}$$

де $w \in R^n$ – ваговий вектор, а $a, b \in R$ – зсув.

Метою SVM є максимізація відстані (margin) між гіперплощиною та найближчими об'єктами навчальної вибірки (опорними векторами). Ця задача формулюється як:

$$\min_{w,b} \frac{1}{2}\|w\|^2, \tag{3.6}$$

за умов

$$y_i(w^T x_i + b) \geq 1, i = 1,2,...,N, \tag{3.7}$$

де $C > 0$ – параметр регуляризації, що визначає баланс між шириною роздільної смуги та кількістю помилок.

У випадках, коли дані є нелінійно роздільними у вихідному просторі ознак, застосовується відображення:

$$\phi: \mathbb{R}^n \to \mathbb{R}^m, m > n, \tag{3.8}$$

яке переносить дані у простір вищої розмірності. Обчислення виконується неявно завдяки використанню ядерної функції:

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j). \tag{3.9}$$

Найбільш поширеними ядрами є [3]:

– лінійне ядро: $K(x_i, x_j) = x_i^T x_j$;

– поліноміальне ядро: $K(x_i, x_j) = (x_i^T x_j + c)^d$;

– радіально-базисне ядро (RBF): $K(x_i, x_j) = e^{-\gamma\|x_i - x_j\|^2}$.

Задамо загальну схему роботи SVM у вигляді конкретних кроків:

1. Вибір ознак і підготовка навчальних даних.

2. Побудова оптимальної роздільної гіперплощини (лінійної або за допомогою ядра).

3. Визначення опорних векторів, які задають межу розділення.
4. Класифікація нових об’єктів за правилом:

$$f(x) = sign(w^T x + b). \tag{3.10}$$

У цій роботі для класифікації стеганографічно вбудованих даних використано ансамбль класифікаторів на основі методу опорних векторів. Ансамбль складається з трьох незалежних SVM-класифікаторів, кожен з яких навчається на гістограмах окремого коефіцієнта сингулярного розкладу (SVD) зображення: (1,5), (5,1) і (5,5). Такий підхід дозволяє ефективно використовувати різну інформацію про модифікації в кожному коефіцієнті та підвищити точність класифікації.

Застосовано такі параметри окремих SVM:

– в якості ядра використано радіальну базисну функцію (RBF), яка забезпечує здатність класифікатора працювати з нелінійними розділеннями у багатовимірному просторі ознак;

– увімкнено стандартизацію, що гарантує нормалізацію вхідних даних для стабільності та коректної роботи RBF;

– для кожного SVM застосовано навчання з апостеріорними ймовірностями, що дозволяє отримувати ймовірності належності зразка до класів «чисте зображення» або «стеганоповідомлення».

Розроблений метод виявлення стеганографічних вбудовувань, реалізований на основі SVD UV, подано у вигляді трьох послідовних етапів. Перший етап стосується підготовки даних: формування гістограм окремих коефіцієнтів сингулярного розкладу зображень, нормалізація ознак та поділ вибірки на навчальну та тестову. Другий етап займається навчанням ансамблю SVM-класифікаторів, де кожен екземпляр моделі відповідає за окремий коефіцієнт SVD, використовуючи ядро RBF та апостеріорні ймовірності для підвищення точності класифікації. Третій етап – аналіз зображення, під час якого кожне тестове зображення обробляється всіма трьома SVM, а результати ансамбля усереднюються для отримання кінцевого передбачення класу

("cover" або "stego"). Такий підхід забезпечує надійну та ефективну класифікацію стеганографічно модифікованих даних.

Запишемо кожний етап у вигляді конкретних кроків.

Метод М(3)1. Підготовка вхідних даних

1.1. Вхідне зображення F розбивається на 8×8-блоки X_i стандартним чином.

1.2. Для кожного блока будується двовимірне перетворення Уолша-Адамара W_{X_i} (2.1).

1.3. При обробленні блоків зображення для трансформант перетворення Уолша-Адамара (1,5), (5,1) і (5,5) будуються три послідовності

$$\begin{aligned} u_{15} &= \left[w_{X_1,15} \quad w_{X_2,15} \quad \dots \quad w_{X_n,15} \right]; \\ u_{51} &= \left[w_{X_1,51} \quad w_{X_2,51} \quad \dots \quad w_{X_n,51} \right]; \\ u_{55} &= \left[w_{X_1,55} \quad w_{X_2,55} \quad \dots \quad w_{X_n,55} \right]. \end{aligned} \tag{3.11}$$

1.4. Будується гістограма розподілу послідовностей u_{15}, u_{51}, u_{55}, що складатиметься з 500 рівномірно вибраних на інтервалах $[\min(u_{15}), \max(u_{15})], [\min(u_{51}), \max(u_{51})], [\min(u_{55}), \max(u_{55})]$ бінів.

Метод М(3)2. Навчання SVM-класифікаторів (ансамбль)

2.1. Для навчання ансамблю SVM формується вибірка з 530 оригінальних зображень та 530 зображень зі стеганоповідомленнями. Для кожного зображення обчислюються гістограми ключових трансформант перетворення Уолша-Адамара ((1,5), (5,1) та (5,5)) і нормалізуються для підвищення стабільності процесу навчання. Отримані вектори ознак утворюють вхідні дані для трьох окремих SVM-класифікаторів.

2.2. Кожен вектор ознак позначається як належний до класу: "cover" – оригінальні зображення; "stego" – зображення зі стеганоповідомленням.

2.3. Вибірка розбивається на навчальну (80%) та тестову (20%) частини, що дозволяє оцінювати здатність моделей узагальнювати закономірності на нових даних, не використаних під час навчання.

2.4. Проводиться навчання трьох SVM-класифікаторів, по одному на кожен ключовий коефіцієнт. Для кожного екземпляра використовується ядро RBF та стандартна нормалізація ознак. Після навчання до моделей застосовується метод навчання з апостеріорними ймовірностями для отримання апостеріорних ймовірностей належності до класів.

Метод М(3)3. Прийняття рішення

4.1. На основі зображення, яке поступає на вхід методу, для кожного ключового коефіцієнта сингулярного розкладу (1,5), (5,1) та (5,5) обчислюється вектор ознак у вигляді гістограми та нормалізується для забезпечення стабільності класифікації.

4.2. Кожен вектор ознак подається на вхід відповідного навченого SVM-класифікатора з ядром RBF. Класифікатори обчислюють апостеріорні ймовірності належності зображення до класів "cover" або "stego".

4.3. Ймовірності, отримані від трьох класифікаторів, усереднюються для формування єдиного передбачення ансамбля. На основі отриманого значення визначається клас із максимальною ймовірністю: якщо усереднена ймовірність класу "stego" перевищує поріг 0.5, зображення відноситься до класу «зі стеганоповідомленням», інакше – до класу «чисте». Такий підхід забезпечує автоматичне, надійне прийняття рішення щодо наявності стеганоповідомлення у вхідному зображенні, враховуючи інформацію з усіх ключових коефіцієнтів перетворення Уолша-Адамара та підвищуючи точність класифікації.

Для оцінки ефективності запропонованого методу стеганоаналізу застосовується матриця невідповідностей (рис. 3.7), яка дозволяє наочно оцінити точність класифікації вхідних зображень. Матриця відображає кількість правильно та неправильно класифікованих зразків для кожного класу: «чисті» зображення та зображення зі стеганоповідомленням. Такий підхід дозволяє не лише оцінити загальну точність ансамблю SVM, а й виявити випадки хибних позитивних та хибних негативних класифікацій, що є важливим для аналізу надійності методу.

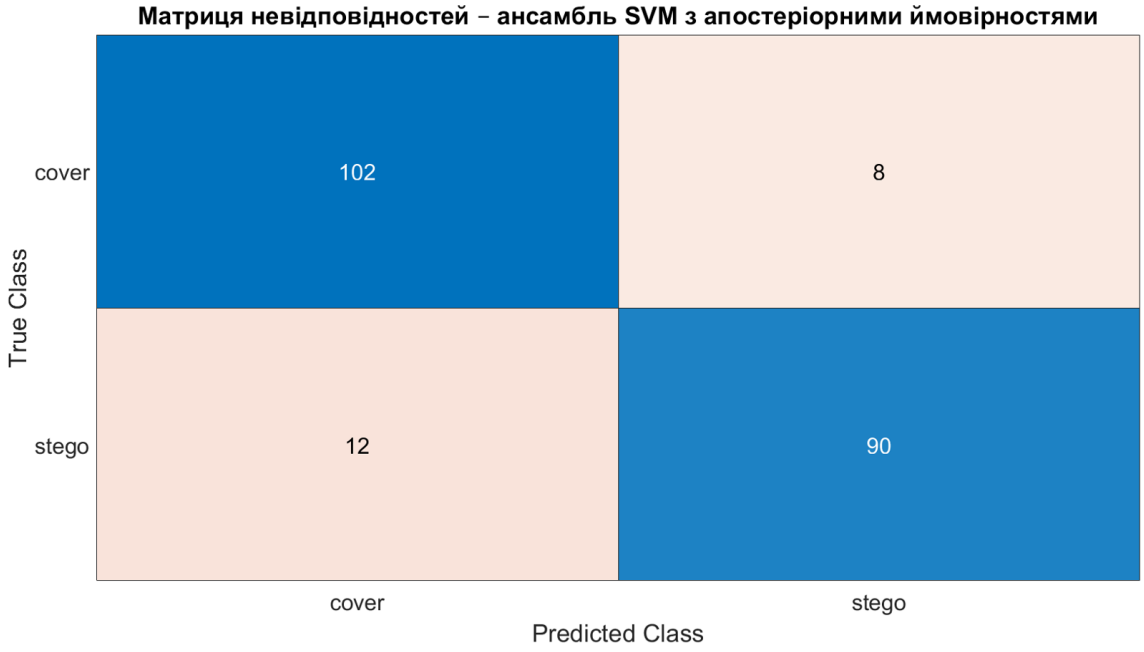


Рисунок 3.7 – Матриця невідповідностей результатів роботи методу виявлення стеганоповідомлень на основі ансамблю SVM

Для більш детальної оцінки роботи ансамблю SVM крім матриці невідповідностей застосовуються метрики Precision, Recall та F1-score:

$$\begin{aligned} Precision &= \frac{TP}{TP+FP} = \frac{192}{192+8} \approx 0.96; \\ Recall &= \frac{TP}{TP+FN} = \frac{192}{192+12} \approx 0.94; \\ F1-score &= 2\frac{Precision \cdot Recall}{Precision + Recall} = 2 \cdot \frac{0.96 \cdot 0.94}{0.96+0.94} \approx 0.95. \end{aligned} \tag{3.12}$$

Аналіз отриманих метрик підтверджує високу ефективність запропонованого ансамблю SVM. Значення $Precision \approx 0.96$ свідчить про те, що більшість зображень, віднесених алгоритмом до класу «зі стеганоповідомленням», були класифіковані правильно, тобто рівень хибнопозитивних спрацьовувань є низьким. $Recall \approx 0.94$ свідчить про те, що метод здатен виявляти майже всі реальні зразки зі стеганоповідомленнями, мінімізуючи пропуски. Високе значення $F1\text{-}score \approx 0.95$ відображає

збалансовану роботу класифікатора, поєднуючи точність та повноту. Таким чином, отримані результати свідчать про надійність і точність методу для автоматичного виявлення вбудованої інформації у зображеннях. При цьому загальна точність виявлення складає понад 90.57%.

Для оцінки практичної ефективності запропонованого методу стеганоаналізу було виміряно час обробки одного зображення тестової вибірки. Встановлено, що на комп'ютері з процесором AMD Ryzen 5 7500F (6 ядер), оперативною пам'яттю 32 ГБ та графічним адаптером NVIDIA GeForce RTX 3050 середній час обробки становить 0.00077 секунд, що свідчить про високу швидкодію методу. Такий результат забезпечує можливість застосування розробленого методу у вебзастосунках у режимі, наближеному до реального часу, а також дозволяє обробляти великі масиви цифрових зображень без значних витрат обчислювальних ресурсів. Висока продуктивність у поєднанні з точністю класифікації робить метод придатним для автоматизованих систем виявлення стеганоповідомлень.

Висновки до розділу 3

Дослідження запропонованих методів, які поєднують штучний інтелект з особливостями трансформант перетворення Уолша-Адамара, показало їхню високу ефективність щодо виявлення прихованих повідомлень у цифрових зображеннях. Таке поєднання дозволяє автоматично аналізувати великі обсяги даних, виділяючи характерні закономірності та відмінності між чистими та модифікованими зображеннями. Висока швидкодія та надійність алгоритмів роблять їх придатними для інтеграції у вебзастосунки, де обробка інформації відбувається у режимі, наближеному до реального часу, що дозволяє забезпечити автоматичний, ефективний стеганоаналіз у мережевому середовищі.

1. Розв'язано проблему виявлення прихованих повідомлень, сформованих за допомогою стеганографічного методу з кодовим управлінням,

який характеризується високою стійкістю до наявних підходів стеганоаналізу. Запропоновано два ефективні методи, які базуються на поєднанні спектральних перетворень і сучасних технологій ШІ: перший метод ґрунтується на аналізі огинаючої гістограм трансформант перетворення Уолша-Адамара та подальшій класифікації за допомогою багатошарової нейронної мережі; другий метод реалізує безпосередній аналіз розподілів трансформант із використанням ансамблю згорткових нейронних мереж, що дозволяє автоматично виокремлювати характерні закономірності без попередньої обробки даних. Проведені експерименти підтвердили високу ефективність обох підходів. Для методу на основі огинаючої гістограм точність виявлення прихованих повідомлень становила 93.3%, тоді як для методу на основі ансамбля згорткових нейронних мереж – 94.3%. При цьому показники Precision, Recall та F1-score перебувають на рівні 0.95…0.98, що свідчить про збалансованість алгоритмів між чутливістю та специфічністю. Водночас час обробки одного зображення не перевищує 0,005 секунди, що забезпечує можливість використання розроблених методів у вебзастосунках у режимі, наближеному до реального часу. Порівняння з класичними методами стеганоаналізу (RS-аналіз, χ^2 -аналіз, аналіз пар, утиліта StegExpose) показало, що вони майже не здатні виявляти приховану інформацію, вбудовану методом з кодовим управлінням. Це підтверджує суттєву перевагу розроблених методів, які демонструють високу точність, навіть у випадках, коли інші підходи є неефективними.

2. Розроблено метод стеганоаналізу на основі ШІ і застосування трансформант перетворення Уолша-Адамара для виявлення стеганографічних вбудовувань на основі SVD UV-методу. Проведений експеримент підтвердив високу ефективність запропонованого методу виявлення стеганоповідомлень на основі ансамблю SVM з апостеріорними ймовірностями. Загальна точність класифікації склала 90.57%, що свідчить про здатність методу правильно відносити значну більшість зображень до відповідного класу – «чисте» або «зі

стеганоповідомленням». Аналіз додаткових метрик підтверджує надійність моделі: Precision = 0.96 показує мінімальний рівень хибнопозитивних спрацьовувань, тобто лише 4% зображень, помилково визначених як стеганоповідомлення, не містять вбудованої інформації. Recall = 0.94 вказує на те, що метод успішно виявляє 94% реальних стеганоповідомлень, що гарантує мінімальні пропуски спотворених зображень. Високе значення F1-score = 0.95 підтверджує збалансованість роботи класифікатора, забезпечуючи оптимальне поєднання точності та повноти. Таким чином, запропонований ансамбль SVM показав високий рівень здатності узагальнювати закономірності на нових даних та ефективно ідентифікувати стеганоповідомлення у зображеннях. Результати експериментів демонструють, що метод можна застосовувати для автоматичної та надійної класифікації великого масиву цифрових зображень, що підтверджує практичну цінність розробленого підходу у задачах стеганоаналізу.

РОЗДІЛ 4. РОЗРОБЛЕННЯ МЕТОДІВ ПРОТИДІЇ ФІШИНГУ ТА ІНТЕГРАЦІЯ КОМПЛЕКСУ ІНТЕЛЕКТУАЛЬНИХ МЕТОДІВ У ПРОЦЕСИ ТЕСТУВАННЯ БЕЗПЕКИ ВЕБЗАСТОСУНКІВ

4.1. Захист вебзастосунків від фішингових атак: актуальність, загрози та підходи до вирішення

Сучасні вебзастосунки є основою для реалізації багатьох бізнес-процесів, сервісів електронної комерції, освітніх платформ та соціальних мереж. Їхня доступність через браузер робить їх зручними для користувачів, проте водночас створює численні вектори атак, серед яких особливу небезпеку становлять фішингові атаки. Вони спрямовані на введення користувачів в оману з метою отримання конфіденційних даних, зокрема облікових записів, фінансової інформації чи персональних даних.

Актуальність проблеми зумовлена тим, що безпечність переходу за посиланнями є ключовим аспектом загальної безпеки вебзастосунків. Зловмисники активно використовують шкідливі гіперпосилання як у зовнішньому середовищі (через електронну пошту, соціальні мережі чи месенджери), так і безпосередньо всередині вебзастосунків. Наприклад, однією з поширених загроз є Stored XSS-атаки, коли шкідливе посилання зберігається у базі даних вебзастосунку та згодом відображається на сторінках ресурсу. Це дозволяє атакувальнику автоматично поширювати фішингові переходи серед усіх користувачів, які взаємодіють із скомпрометованими даними.

Крім того, поширеними прикладами загроз є:

– Reflected XSS, коли зловмисне посилання формується динамічно у відповідь сервера та одразу використовується для атаки;

– Clickjacking, де користувача обманом змушують натиснути на невидимий або маскований елемент, що веде до шкідливого ресурсу;

– Open Redirects, коли вразливість у логіці перенаправлень дозволяє підмінювати легітимні посилання шкідливими;

– Man-in-the-Middle-атаки, що можуть змінювати вміст вебсторінок або підмінювати посилання під час передавання даних через незахищені канали.

Через це захист вебзастосунків від фішингових атак є пріоритетним завданням інформаційної безпеки. Його реалізація потребує розроблення спеціальних методів виявлення та блокування небезпечних посилань, аналізу їхнього походження та контексту використання. Це дозволить мінімізувати ризики компрометації даних користувачів і забезпечити довіру до вебсервісів.

Окрему увагу слід приділяти наявності потужного інструмента перевірки посилань на можливість фішингової атаки. Такий інструмент має забезпечувати не лише високу ефективність виявлення потенційно небезпечних ресурсів, а й здатність функціонувати у середовищах із обмеженими обчислювальними ресурсами, наприклад, на мобільних пристроях чи вбудованих системах. Це пояснюється тим, що саме ці платформи найчастіше використовуються для доступу до вебзастосунків і їхня вразливість може стати критичною для безпеки користувачів. Відтак розроблення методів захисту має враховувати баланс між обчислювальною складністю, точністю виявлення та універсальністю застосування у різних середовищах.

Завданням, яке вирішується у цьому розділі, є створення ефективного методу захисту вебзастосунків від фішингових атак.

4.2. Аналіз модальностей фішингових атак

Як було детально розглянуто у підрозділі 1.4, фішингові атаки є однією з найбільш системних загроз для сучасних вебзастосунків, ефективність яких базується на поєднанні методів соціальної інженерії та використанні технічних

вразливостей. Масштаби цієї загрози вражають: за даними досліджень [196] 2024 року, 94% організацій стикалися з фішинговими атаками.

Значний вплив має фішинг через електронну пошту, який є джерелом 90% атак програм-вимагачів. При цьому середній розмір збитків від однієї фішингової атаки на підприємство становить $4.65 мільйона, а глобальні фінансові втрати від фішингових атак за 2024 рік оцінили у $17.4 мільярда, що на 45% більше проти попереднього року [197].

З появою ШІ у кіберзлочинців з'явилася можливість створювати цільові фішингові кампанії, використовуючи генеративні моделі та алгоритми обробки природної мови. Генеративні можливості сучасних великих мовних моделей (Large Language Model, LLM), генеративно-змагальних мереж (Generative Adversarial Networks, GAN), моделі синтезу мовлення (Text-to-Speech, TTS) та клонування голосу, моделі обробки природної мови (Natural Language Processing, NLP) з контекстною адаптацією, моделі комп'ютерного зору (Computer Vision, CV) в комплексі суттєво розширили арсенал засобів, які використовують зловмисники у фішингових атаках [16]. Зокрема автоматизоване створення переконливих фішингових повідомлень, deepfake-контенту (аудіо, відео, зображень), а також фальшивих профілів користувачів дозволяє зловмисникам з високим ступенем правдоподібності імітувати людську поведінку. Це, зі свого боку, суттєво ускладнює виявлення таких загроз традиційними методами. Саме тому 43% користувачів не здатні правильно ідентифікувати фішинговий електронний лист, що свідчить про критичну потребу як у покращенні навчання з питань кібергігієни, так і в розробленні інноваційних автоматизованих систем виявлення фішингових атак [198].

Дослідженню проблем виявлення кібератак за допомогою алгоритмів ШІ та машинного навчання присвячені численні публікації науковців і дослідників. Так, дослідження [199] спрямоване на аналіз впливу технологій машинного навчання на зміну та адаптацію кіберзагроз, зокрема на способи використання ШІ зловмисниками для підвищення ефективності, масштабу та

складності атак. Автори статті [200] запропонували методику виявлення зловмисних команд, прихованих у графічних файлах за допомогою стеганографічних методів. Робота [201] висвітлює різні методи машинного навчання для тестування на проникнення. У дослідженні [202] запропоновано архітектуру та методологію впровадження системи Threat Guard AI для пом'якшення цифрових загроз. Дослідження [203] розглядає комбінований метод з алгоритмами глибокого навчання, машинного навчання та ройового інтелекту для виявлення фішингових атак. У роботі [203] запропоновано до розгляду програмний застосунок, який для аналізу введених URL-адрес щодо виявлення потенційно небезпечних посилань використовує три алгоритми машинного навчання: випадковий ліс (Random Forest, RF), логістична регресія (Logistic Regression) та опорно-векторне кластерування на основі методу опорних векторів (Support Vector Machine, SVM). У дослідженні [204] для виявлення фішингових вебсайтів використано дещо інший набір алгоритмів: бустинг градієнта (Gradient Boosting), RF, CatBoost та логістичної регресії. Автори статті [206] для навчання розробленої ними моделі для виявлення у мережевому трафіку різного роду кібератак на вебсайти використали відкритий набір даних CSE-CIC-IDS2017 та алгоритми машинного навчання: RF, наївний баєсів класифікатор, k-найближчих сусідів (K-Nearest Neighbor, KNN), SVM з використанням гауссівського ядра, адаптивний бустинг, бустинг градієнта.

Однак, попри велику кількість публікацій, присвячених боротьбі з фішинговими атаками, цю проблему не можна назвати вирішеною остаточно, про що свідчить безперервне зростання збитків, спричинених саме цим видом атак.

Фішингові атаки ґрунтуються на соціальній інженерії, оскільки зловмисники використовують фальсифіковані повідомлення здебільшого у вигляді електронної пошти або комунікацій у соціальних мережах з метою спрямування користувача на спеціально створені фішингові вебресурси [11]. За статистичними даними, щомісяця створюється приблизно 1.5 мільйона

нових фішингових вебсайтів [207]. Такі вебресурси зазвичай видають себе за відомі компанії, як-от Microsoft та Google, обманом змушуючи людей надавати особисту інформацію. Імітуючи легітимні сторінки авторизації або онлайн-форми, вони здійснюють несанкціоноване збирання конфіденційної інформації. Отримані дані в подальшому використовуються для реалізації шахрайських схем або крадіжки особистих відомостей. Так, 79% випадків захоплення облікових записів розпочинались саме з фішингових електронних листів [208]. Крім того, гіперпосилання, вміщені у подібних повідомленнях, нерідко слугують каналом доставки шкідливого програмного забезпечення (ПЗ) до інформаційної системи користувача.

Розрізняють різні види фішингу (рис. 4.1) [11].

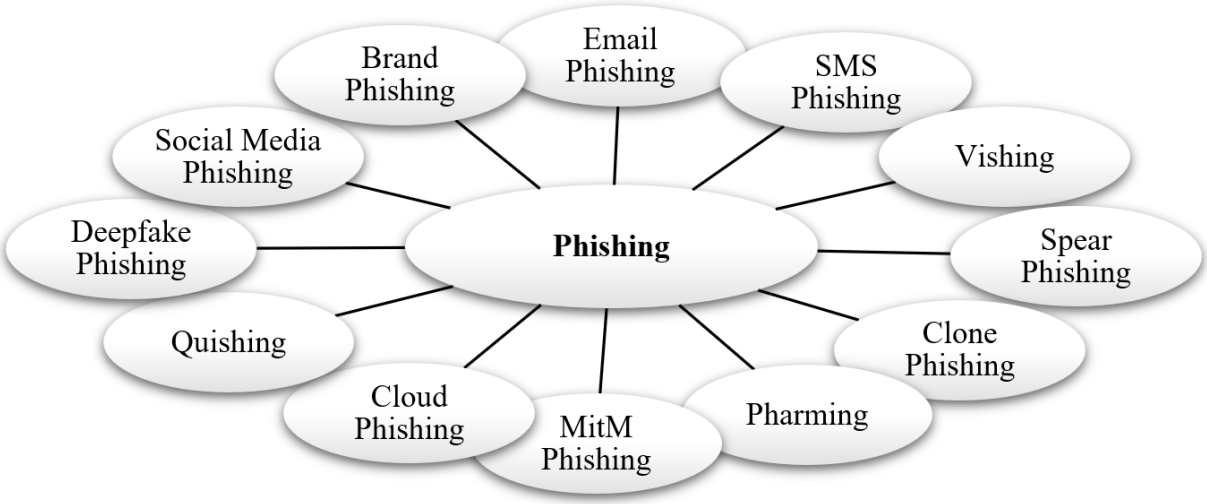


Рисунок 4.1 – Поширені різновиди фішингу

– *Фішинг електронною поштою (Email Phishing)* є класичним найпоширенішим різновидом, з яким зіткнулося вже 86% компаній у всьому світі [196]. Шахрайство через електронну пошту зросло на 70% проти минулого року завдяки здатності ШІ імітувати тон, підробляти внутрішні електронні листи та персоналізувати повідомлення з вражаючою точністю [209].

– *Смішинг (SMS Phishing)* є фішинговою шахрайською схемою через текстові повідомлення, які спонукають людину перейти за посиланням, завантажити шкідливе ПЗ або надати особисту інформацію. Популярність смішингу зумовлена тим, що повідомлення виглядають переконливо та вимагають миттєвих дій, що є приманкою і робить зайняту на роботі людину легкою здобиччю. Поширено зловмисники використовують фальшиві повідомлення про неоплачені збори або доставки, щоб спонукати користувачів перейти за шкідливими посиланнями. Приблизно три з чотирьох організацій у всьому світі стикалися з цим типом атаки, причому 2024 року кількість зареєстрованих смішингових атак зросла на 25% [210].

– *Вішинг (Vishing)* або голосовий фішинг є шахрайством через телефонні дзвінки або голосові повідомлення, щоб обманом виманити у людей конфіденційні дані. Наприклад, зловмисник телефонує, представляється працівником банку та намагається дізнатися PIN-код або CVV-код картки. Зловмисники використовують інструменти на базі генеративного ШІ, щоб видавати себе за керівників компаній або державних службовців, обманюють своїх жертв, змушуючи їх думати, що вони розмовляють з кимось, кого вони знають і кому довіряють. Цей вид фішингу є класичним прикладом атак соціальної інженерії. Сім з десяти організацій стикалися з шахрайськими схемами вішингу, причому 2024 року кількість атак вішингу на фахівців компаній зросла на 28% [211].

– *Цільовий фішинг (Spear Phishing)* є атакою на конкретну людину або організацію задля збору особистої інформації. Наприклад, персоналізований фішинг-лист, надісланий бухгалтеру з підробленої адреси нібито від керівника з використанням імен, посад, деталей про компанію. Такий лист від «вашого колеги» з проханням подивитися важливий договір (а насправді – шкідливий файл) виглядає максимально переконливим і спрямований він на те, щоб змусити жертву клікнути на шкідливе посилання, відкрити файл або розкрити конфіденційні дані. Не дивно, що 65% успішних фішингових атак пов’язані

саме з цільовим фішингом, причому 2024 року спостерігалося збільшення цього виду атак на 30% [211].

– *Фішинг у соціальних мережах (Social Media Phishing)* задля того, щоб отримати логін/пароль або змусити встановити шкідливе ПЗ, використовує фальшиві профілі або повідомлення від друзів у Facebook, Instagram тощо. Приблизно 33% спроб фішингу 2024 року були спрямовані через соцмережі [198].

– *Клонування фішингу (Clone Phishing)* стосується копіювання реального листа, наприклад, з повідомленням від банку, але посилання чи вкладення в ньому замінені на шкідливі. Приблизно 19% зареєстрованих фішингових електронних листів є спробами клонування фішингу, причому їхня кількість 2024 року зросла на 25% проти попереднього року [212].

– *Фармінг (Pharming)* використовує переспрямування користувачів з легітимних вебсайтів на підроблені шкідливі. Атаки фармінгу часто використовують вразливості в DNS-серверах або шкідливе ПЗ на пристрої користувача і перенаправляють користувачів з легітимних вебсайтів на підроблені шкідливі. 2024 року 12% фішингових атак використовували цю тактику, здебільшого спрямовану на вебсайти онлайн-банкінгу або електронної комерції [198].

– *Фішинг «людина посередині» (Man-in-the-Middle, MitM Phishing)* – це різновид атак з перехопленням даних вебсесії, наприклад, між користувачем та онлайн-платіжним сервісом, який намагається обдурити кожну з двох сторін, змусивши їх думати, що він є іншою стороною, захоплює та/або змінює інформацію, яка передається між жертвою та законною стороною, приміром, під час доступу до онлайн-банкінгу, коли зловмисник може перехопити запити або змінити дані в реальному часі (особливо у незахищених мережах Wi-Fi). Атаки MitM часто використовують вразливості в мережах Wi-Fi або скомпрометованих маршрутизаторах. Зловмисники можуть використовувати методи перехоплення пакетів, захоплення сеансів і підміни HTTPS для

перехоплення каналів зв’язку та отримання несанкціонованого доступу до конфіденційних даних. Атаки MitM стосуються перехоплення та маніпулювання зв’язком між двома сторонами без їхнього відома. При цьому зазвичай жертва навіть не здогадується, що її з’єднання перехоплено, а тому це вкрай небезпечний тип атак. Наприклад, користувач здійснює великий платіж надійному постачальнику, але стороння особа перехоплює транзакцію та непомітно змінює банківські реквізити. Користувач вважає, що платіж безпечний, але його перенаправляють на рахунок злочинця. 14% випадків фішингу припадає на цей тип атак, коли зловмисники перехоплюють зв’язок між жертвою та довіреною особою і в реальному часі крадуть дані, зокрема під час онлайн-банківських транзакцій. 2024 року атаки MitM становили 23% кіберінцидентів, пов’язаних з ідентифікацією [213]. Через свою примарну природу їх легко недооцінювати, але не варто цього робити. Атаки MitM можуть призвести до різних злочинних дій: від масштабного шахрайства до захоплення облікових записів та фінансових втрат [11].

– *Хмарний фішинг (Cloud Phishing)* є серйозною проблемою, оскільки все більше компаній покладаються на хмарні сервіси. Така атака для отримання облікових даних імітує інтерфейси Google Drive, Microsoft OneDrive, Dropbox тощо. Наприклад, користувач отримує посилання, схоже на сторінку входу в Google Workspace або Microsoft 365, де пропонується підтвердити доступ і ввести пароль. Але при цьому відбувається переспрямування на підроблену сторінку входу до хмарного сервісу. 37% фішингових атак 2024 року були спрямовані саме на хмарні сервіси.

– *Фішинг через QR-коди (Quishing)* популярний через повсюдне поширення QR-кодів. Зловмисник поширює підроблений QR-код у ресторанах, листівках, оголошеннях тощо. Після сканування користувач переходить на шкідливий сайт або дає дозвіл на встановлення ПЗ. При цьому здебільшого відбувається обхід фільтрів безпеки, оскільки зазвичай системи не аналізують QR-коди так, як звичайні URL. Наразі приблизно 5% сучасних

атак здійснюються через підроблені QR-коди, які спрямовують користувача на шкідливі вебсайти, а 2024 року їхня кількість зросла на 25% проти попереднього року [214].

– *Бренд-фішинг (Brand Phishing)* є видом атак, коли зловмисники імітують відомий бренд (наприклад, Google, PayPal, Microsoft, Netflix, Apple), тобто створюють підроблені вебсайти, логотипи, домени тощо. Метою є переконання користувача у тому, що він взаємодіє з офіційним джерелом, щоб отримати його облікові дані або змусити здійснити платіж. За даними Check Point, 2024 року 40% фішингових атак були пов'язані з імітацією відомих брендів: Microsoft у фішингових атаках склав 32% усіх [215].

– *Deepfake-фішинг* – новий, небезпечний вид фішингу, в якому використовуються штучно згенеровані відео, аудіо або зображення. Зловмисники, застосовуючи генеративний ШІ ChatGPT, Sora, ElevenLabs, імітують голос або зовнішність відомих осіб, зокрема керівників компаній, державних службовців або колег. Така атака може здійснюватися через відеодзвінки, голосові повідомлення або навіть Zoom-конференції. Через розвиток ШІ 2024 року кількість deepfake-атак зросла на понад 50% [216].

Використання ШІ значно підвищує ефективність фішингових атак, робить їх більш переконливими та важкими для виявлення. Тому важливою складовою стратегії кібербезпеки є впровадження комплексних заходів захисту, включно з навчанням користувачів, використанням багатофакторної автентифікації та постійним оновленням систем безпеки.

Як видно з наведеної класифікації, значна частина сучасних фішингових атак передбачає взаємодію користувача саме з вебінтерфейсом – підробленою вебсторінкою, яка імітує легітимний ресурс (банківський сайт, хмарний сервіс, корпоративний портал тощо) [11]. Такі фішингові сайти є центральною ланкою багатьох типів атак: натиснувши на посилання у фішинговому листі, повідомленні чи QR-коді, користувач потрапляє на візуально правдоподібну, але шкідливу вебсторінку. Згідно з даними за 2024 рік, понад 70% фішингових

атак у підсумку зводяться до взаємодії жертви з підробленим вебінтерфейсом. Це означає, що навіть попереднє виявлення фішингового повідомлення не гарантує повної безпеки, якщо користувач все ж відкриє сайт і візуально не зможе відрізнити його від справжнього.

4.3. Наявні підходи протидії фішингу

Важливим аспектом аналізу фішингових атак є врахування підходів впливових міжнародних організацій, які систематично досліджують цю проблему та публікують рекомендації для розробників і фахівців із кібербезпеки.

У сучасному цифровому середовищі кіберзагрози стають дедалі складнішими та різноманітнішими. Попри розвиток технічних засобів захисту, фішинг зберігає стабільну присутність у списках найбільших кіберзагроз, що підтверджується міжнародними рейтингами та стандартами, зокрема OWASP Top 10, NIST Cybersecurity Framework та рекомендаціями ENISA.

Організація OWASP (Open Web Application Security Project) щорічно публікує рейтинг Top 10 найбільш критичних вразливостей вебзастосунків [217]. Хоча безпосередньо фішинг у Top 10 не виділяється як окремий пункт, його вплив проявляється у категоріях, що стосуються аутентифікації та контролю доступу, а також у ширшому контексті соціальної інженерії. Фішингові атаки використовують психологічні механізми обману, що дозволяє обійти навіть складні технічні механізми захисту, і тому вони систематично вважаються однією з найбільших загроз для вебзастосунків.

Американський Національний інститут стандартів і технологій (NIST) розробив Cybersecurity Framework (https://www.nist.gov/cyberframeworki), який слугує керівництвом для управління ризиками у сфері інформаційної безпеки. У рамках цього стандарту фішинг розглядається як ключовий ризик, пов’язаний з людським фактором. Важливість фішингових атак полягає у тому, що вони здатні обходити навіть високорівневі технічні контролі та створюють

пряму загрозу для конфіденційної інформації та безпеки облікових записів. NIST рекомендує інтегрувати заходи протидії фішингу у стратегію управління ризиками, що включає підвищення обізнаності користувачів, багатофакторну аутентифікацію та використання автоматизованих систем раннього попередження про підозрілі листи.

Агентство ЄС з кібербезпеки (ENISA) також приділяє значну увагу фішингу як системній загрозі. У регулярних звітах про кіберзагрози фішинг стабільно входить до десятки найнебезпечніших ризиків, що впливають на організації різного масштабу. Європейські рекомендації акцентують не лише на технічних аспектах захисту, а й на соціальному компоненті: ефективність фішингу значною мірою залежить від поведінки користувачів [218]. ENISA радить організаціям розробляти політики інформаційної безпеки, проводити тренінги та симульовані фішингові кампанії, а також інтегрувати програмне забезпечення для блокування підозрілих повідомлень і посилань. Комплексний підхід, що поєднує технічні та організаційні заходи, дозволяє значно знизити ризики успішних атак.

Фішинг зберігає стабільну позицію серед глобальних кіберзагроз через низку факторів. Перш за все, його ефективність ґрунтується на людському факторі: користувачі, навіть при наявності сучасних засобів безпеки, можуть стати слабким місцем системи. По-друге, розвиток технологій дозволяє зловмисникам створювати персоналізовані та автоматизовані фішингові кампанії, що підвищує ймовірність успіху атак. По-третє, наслідки фішингу мають значний вплив на бізнес-процеси, оскільки він може призводити до фінансових втрат, компрометації даних і порушення діяльності організацій. Аналіз міжнародних стандартів демонструє, що систематичне включення фішингу у ТОП-10 загроз відображає його значимість та потребу комплексного підходу до протидії, який поєднує технічні засоби, навчання персоналу та моніторинг інцидентів.

Міжнародні стандарти та рекомендації у сфері кібербезпеки, зокрема OWASP Top 10, NIST Cybersecurity Framework та ENISA, підкреслюють, що

фішинг є однією з найбільш системних загроз сучасного інформаційного середовища. Його ефективність пояснюється поєднанням соціальної інженерії та технічних вразливостей, а стабільна присутність у рейтингах загроз вказує на критичну необхідність комплексного управління ризиками. Для організацій та користувачів важливо інтегрувати практики підвищення обізнаності, застосовувати багатофакторну аутентифікацію та підтримувати сучасні технічні засоби захисту. Такий підхід дозволяє мінімізувати вплив фішингу і підвищити загальний рівень кібербезпеки.

Розглянемо далі конкретні методи протидії фішинговим атакам. Зокрема, йтиметься про використання алгоритмів машинного навчання для автоматизованого виявлення підозрілих посилань і вебресурсів, впровадження багатофакторної автентифікації як засобу мінімізації ризиків компрометації облікових даних, застосування сучасних браузерних технологій захисту (наприклад, фільтрації URL та системи попереджень користувачів). Аналіз цих методів дозволить окреслити їхні переваги та недоліки, визначити ефективність у реальних умовах і сформувати підґрунтя для розроблення власного підходу до захисту вебзастосунків від фішингових атак.

У роботі [98] проведено огляд методів глибокого навчання для виявлення фішингу (URL, контент, DOM, візуальні та гібридні ознаки), метрик, датасетів і відкритих проблем. У цій роботі встановлена сильна залежність роботи методів від статичних датасетів і відсутність «temporal split» (моделі погано переносяться на нові кампанії); перевага офлайн-оцінювання без реальних обмежень продуктивності/ресурсів; обмежена інтерпретованість DL-рішень і слабкі перевірки на стійкість до обходів/адверсарних прикладів.

У роботі [219] автори побудували конвеєр для класифікації фішингових URL з ансамблю моделей і мета-класифікатора (заявлена висока точність на популярному URL-датасеті). Попри це, недоліки цієї роботи: типові публічні URL-датасети містять витік ознак (наприклад, доменні чорні списки/токени), що завищує метрики; переважно випадкове розбиття без часової валідації;

відсутні тести на evasion-стійкість (маніпуляції з піддоменами/шляхами); не показано затримки/пам’ять для вбудованих чи мобільних середовищ.

У роботі [220] порівняно три DL-архітектури для виявлення фішингових вебсайтів і встановлено, що LSTM-CNN демонструє найкращі метрики на кількох наборах. Однак моделі притаманні такі недоліки: не показано роботу на практиці (реальні браузерні події, динамічний контент); можливе перенавчання на статичних знімках/особливостях; немає аналізу чутливості до обфускацій (JS-редиректи, крауд-контент); ресурсоємність LSTM-CNN не співвіднесена з вимогами до мобільних/edge-платформ.

Дослідження [221] вивчає використання генеративно-змагальних мереж для покращення детектора (генерація/балансування зразків, підвищення узагальнення). У моделі є такі недоліки: ризик «галюцинацій» синтетичних зразків; можливе перенавчання на артефактах, внесених генератором; відсутні результати проти цілеспрямованих атак-обходів, коли зловмисник оптимізує URL під конкретний дискримінатор.

У статті [222] досліджується гібридна модель виявлення для фішингу в контексті цифрової криміналістики; заявлене підвищення точності на кількох наборах. Моделі бракує детального порівняння з сучасними сильнобазовими моделями (BERT-сімейство, ViT-конкатенації з DOM-графами) на однакових часових сплітах; не показана пояснюваність для форензики; невідомі вимоги до обчислювальних ресурсів у хмарі та на ресурсообмежених платформах.

У роботі [223] запропоновано метод «identity-based» детекції: поєднання візуальних брендових елементів (логотип/стиль) і текстових ключових слів; повідомлено про точність ~98.6% на кількох тестах. Попри це, можна виділити наступні недоліки роботи: візуальні брендові сигнали крихкі до невеликих трансформацій (мінімальні правки лого, темна тема, адаптивний CSS); метод потребує якісного рендерингу сторінки (це ускладнює застосування на ресурсообмежених платформах); можливий «dataset bias» на конкретні бренди; не показано стійкість до homograph-доменів та контенту з lazy-loading.

У роботі [224] зроблено оцінку великих мовних моделей (та/або fine-tuned BERT-сім'ї) для класифікації фішингових/спам електронних листів, показано варіативні результати та потенціал до fine-тюнінгу (IPSDM). Недоліками є те, що: LLM дорогі обчислювально; без квантизації складно розгорнути на ресурсообмеженних платформах; ризик «data contamination» (попадання схожих прикладів у передтренувальні набори); відсутній глибокий аналіз стійкості до промпт-обфускацій та мультимодальних фішинг електронних листів (зображення, вкладення).

Автори статті [225] пропонують розробку браузерного розширення (для Chromium-браузерів), яке в режимі реального часу перевіряє URL-адреси, попереджає або блокує користувача при виявленні підозрілого посилання. Для класифікації використовується CNN-модель, натренована на великому наборі даних – понад 651 191 URL, яка досягає точності 98.4 %. До недоліків роботи слід віднести залежність від серверної обробки, обмежену реалізацію CNN, відсутність тестування проти адаптивних атак.

Усі розглянуті роботи демонструють значний прогрес у застосуванні методів глибокого навчання для виявлення фішингу, охоплюючи класифікацію URL, контенту вебсторінок, DOM-структури, візуальних ознак та гібридні підходи [98, 219…225]. Особливо критичною є ситуація для вебзастосунків, які в реальному часі мають контролювати переходи користувачів за посиланнями: наявні моделі переважно тестуються офлайн або на синтетичних даних, не враховують динамічний контент, lazy-loading, адаптивний CSS чи маніпуляції з піддоменами та шляхами, що робить їх практично непридатними для оперативного захисту користувачів у реальному середовищі. Таким чином, попри високі показники точності на окремих датасетах, сучасні підходи не забезпечують надійного та масштабованого виявлення фішингових атак у реальному часі.

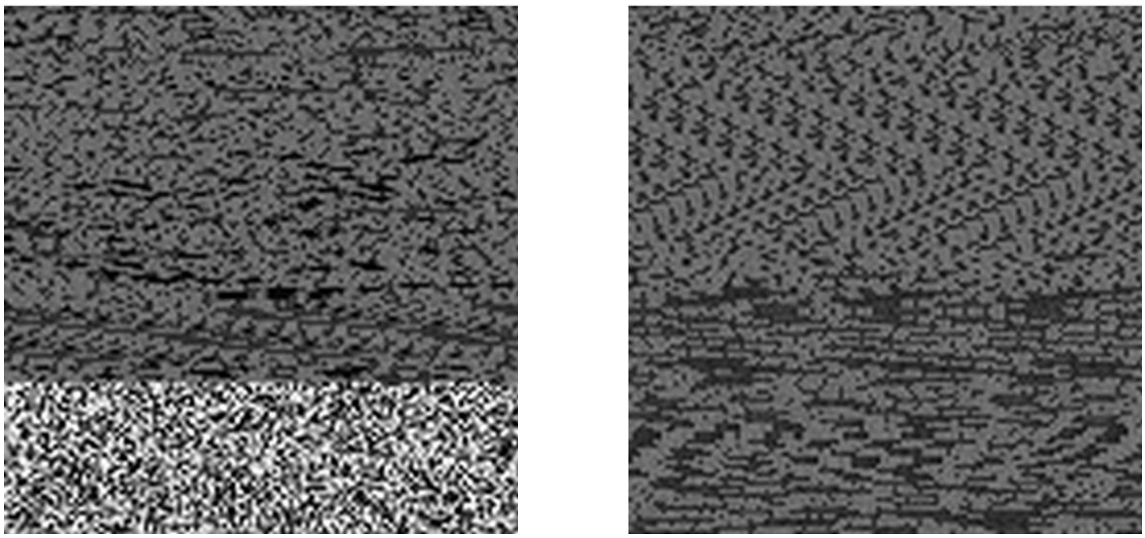
4.4. Запропонований підхід до подання даних під час виявлення фішингового контенту та його подальшого аналізу

Одним із сучасних та перспективних напрямів у галузі аналізу структурованих даних є подання вихідного коду або вебсторінок у формі зображень із подальшим застосуванням методів машинного навчання [226]. Такий підхід дозволяє інтерпретувати код як візуальну структуру, де просторові та текстурні закономірності можуть нести інформативні ознаки, релевантні для класифікації або виявлення аномалій. На відміну від традиційних текстових або синтаксичних аналізаторів, візуальне подання відкриває можливість виявлення шаблонів на основі глобального вигляду документа – його «візуального сліду». Це особливо актуально для завдань виявлення фішингових сайтів, де важливу роль відіграє зовнішня схожість зі справжніми ресурсами, а не лише семантичний вміст коду. Подання коду як зображення дозволяє ефективно використовувати перевірені архітектури нейронних мереж, зокрема згорткові нейронні мережі (CNN), а також традиційні алгоритми машинного навчання, які працюють з візуальними ознаками [15].

Для забезпечення можливості аналізу вебсторінок та подальшої візуалізації їхніх характеристик доцільно здійснити попереднє перетворення HTML-документів у бінарне подання. Попри те, що HTML-код має текстову природу, саме перехід до бітового або байтового рівня дозволяє усунути залежність від синтаксичних особливостей мови розмітки, а також створити уніфіковане подання вхідних даних. Робота з бінарною формою дає змогу виявляти низькорівневі аномалії, регулярності або приховані шаблони, які можуть залишитися непоміченими при аналізі на рівні символів або тегів. Крім того, бінарне подання є зручним для перетворення на матрицю або зображення, що відкриває можливості для застосування методів машинного навчання, орієнтованих на роботу з візуальними структурами.

На наступному етапі виконують візуалізацію HTML-файлу. Байтовий потік перетворюється у масив типу NumPy зі значеннями від 0 до 255, кожне з яких відповідає одному байту. Кожен байт коду подається як сірий піксель у форматі RGB (R = G = B), тобто без кольорової інформації. Отримані пікселі формують зображення розміром 128×128 пікселів, що містить 16 384 значення [11].

На рис. 4.2 наведено приклад подання фішингового та легітимного файлів у вигляді картинок.



Файл fishtest.html URL: https://www.instagram.com/

Рисунок 4.2 – Порівняння візуалізації фішингового та легітимного файлів

У разі перевищення кількості вихідних даних для формування зображення зазначеного розміру надлишок обрізається. Таким чином забезпечується єдиний розмір вхідних зображень для подачі на класифікатор.

Після побудови зображення відбувається процедура нормалізації: значення пікселів перетворюються так, щоб мати нульове математичне сподівання та одиничне стандартне відхилення. Це підвищує ефективність подальшої класифікації. Для аналізу використовуються методи машинного

навчання, зокрема SVM, який працює зі сформованими векторами ознак, отриманими із зображень [1].

Після отримання візуального подання HTML-файлу треба здійснити перетворення зображення на вектор ознак, придатний для подачі на вхід моделі машинного навчання [2]. З метою підвищення інформативності аналізу, зображення розміром 128×128 пікселів розбивається на чотири горизонтальні області: верхню, нижню та дві центральні. Така сегментація дозволяє локалізовано досліджувати візуальні патерни, зокрема в центральній частині зображення, де зазвичай концентруються важливі елементи сторінки, як-от: JavaScript-код, функціональні блоки та логіка взаємодії. Натомість початок і кінець HTML-файлу часто містять службові теги або метаінформацію, які не завжди є релевантними для класифікації. Розбиття також дозволяє моделі краще диференціювати регіональні особливості структури сторінки, що може суттєво впливати на якість розпізнавання фішингових шаблонів. Приклад візуального розмежування областей наведено на рис. 4.3.

Рисунок 4.3 – Розбиття зображення на області

Для кожної з виділених горизонтальних областей зображення обчислюється нормалізована гістограма розподілу яскравості пікселів. Нормалізація гістограми здійснюється шляхом ділення кількості пікселів у кожному біні на загальну кількість пікселів у відповідній області, що забезпечує масштабну незалежність ознак. Отримані гістограми з усіх

чотирьох областей об’єднуються в єдиний вектор ознак довжиною 1024 елементи, що дозволяє зберегти локальні особливості структури HTML-файлу в різних його сегментах. На наступному етапі вектор ознак піддається стандартизації з використанням відповідної функції нормалізації (скейлера) – трансформації, яка зводить кожну ознаку до нульового середнього значення та одиничного стандартного відхилення. Така стандартизація дозволяє усунути вплив масштабних розбіжностей між окремими компонентами вектора, покращує стабільність роботи моделі та забезпечує її узагальнюючу здатність під час обробки вхідних даних, отриманих з різних HTML-документів. Нормалізований вектор ознак є фінальним результатом етапу перетворення даних і подається на вхід моделі ШІ для подальшої класифікації чи прогнозування.

4.5. Запропонований метод виявлення фішингового контенту та навчання нейронної мережі

Ґрунтуючись на запропонованому підході до подання даних, сформуємо метод виявлення фішингового контенту у вигляді конкретних кроків.

Крок 1. Отримання вхідних даних. Користувач подає HTML-файл або посилання на вебсторінку. У разі посилання вебсторінка завантажується та зберігається у вигляді HTML-документа.

Крок 2. Визначення та уніфікація кодування. Визначається кодування HTML-файлу. Якщо кодування відмінне від UTF-8 (наприклад, Windows-1251 або KOI8-U), файл перекодується у UTF-8.

Крок 3. Перетворення HTML-коду у байтовий потік. HTML-файл конвертується у потік байтів (значення від 0 до 255). Цей потік буде базою для подальшої візуалізації.

Крок 4. Формування зображення. Кожному байту відповідає сірий піксель формату RGB (R=G=B). Пікселі розміщуються лінійно у зображенні розміром 128×128 пікселів (загалом 16 384 байти). Якщо байтів менше 16 384,

зображення доповнюється байтами із значенням 255. Якщо байтів більше 16 384, надлишок обрізається.

Крок 5. Нормалізація зображення. Яскравість пікселів (значення 0…255) нормалізується: середнє значення дорівнює 0; стандартне відхилення дорівнює 1.

Крок 6. Сегментація зображення. Зображення поділяється на 4 горизонтальні області: верхня, нижня, дві центральні.

Крок 7. Побудова гістограм. Для кожної з 4-х областей обчислюється нормалізована гістограма яскравості пікселів. Гістограма відображає частотний розподіл пікселів за яскравістю. Кожен бін нормалізується відносно загальної кількості пікселів в області.

Крок 8. Формування вектора ознак. Усі чотири гістограми об'єднуються в один вектор ознак довжиною 1024 елементи. Цей вектор відображає візуальну структуру HTML-документа у стислому числовому вигляді.

Крок 9. Стандартизація вектора ознак. Застосовується функція нормалізації (скейлер): кожна ознака трансформується до середнього, яке дорівнює 0, та стандартного відхилення, яке дорівнює 1. Це дозволяє моделі ефективно навчатися на однорідних ознаках.

Крок 10. Класифікація. Отриманий нормалізований вектор подається на вхід моделі машинного навчання (наприклад, SVM).

Відзначимо, що метод може бути використаний у практичних системах виявлення шкідливих вебсторінок, зокрема у вигляді вбудованого модуля безпеки в браузер, який у фоновому режимі візуалізує HTML-вміст і на основі його зображення оцінює потенційно шкідливу активність. Метод також може бути реалізований у вигляді модуля, вбудованого у серверну частину вебзастосунку, для аналізу всіх посилань, за якими переходить користувач. Завдяки перетворенню коду на зображення забезпечується незалежність від конкретної мови розмітки, що розширює можливості застосування моделі у гетерогенних середовищах.

Для побудови класифікатора була використана модель типу Support Vector Machine з ядром RBF. З відкритого джерела [227] отримано 3000

легітимних та 1700 фішингових HTML-файлів. Після попередньої обробки 4700 файлів було об’єднано в єдиний датасет з доданими мітками класів. Отриманий датасет розділено за пропорцією 80/20, де 80% – вектори, на яких модель буде навчатися виявляти патерни, і 20% – вектори, на яких буде проведено тестування точності моделі.

З метою вибору оптимальних гіперпараметрів моделі SVM було проведено пошук по сітці (Grid Search) з використанням 5-кратної стратифікованої крос-валідації (5-fold stratified cross-validation). Стратифікація забезпечує збереження частки кожного класу (легітимний/фішинговий) у кожному з фолдів, що є критичним для отримання достовірної оцінки якості моделі на незбалансованих даних.

Пошук здійснювався за наступною сіткою параметрів:

- тип ядра (kernel): RBF, лінійне (linear), поліноміальне (poly).
- параметр регуляризації (*C*): [0.1, 1, 10, 100, 200].
- коефіцієнт ядра (*gamma*): ['scale', 'auto', 0.1, 0.01, 0.001, 0.0001] (для ядер RBF та поліноміального).
- ступінь полінома (degree): [2, 3, 4] (лише для поліноміального ядра).

Для кожної комбінації параметрів обчислювалася середня точність класифікації (accuracy) на 5 фолдах. Після завершення пошуку було відібрано набір параметрів, який максимізував середню точність на валідаційних фолдах. Оптимальна конфігурація моделі має такий вигляд:

$$P = \{'C':\ 100,\ 'gamma':\ 'scale',\ 'kernel':\ 'rbf'\}.$$

Реалізацію та результат пошуку найкращих налаштувань наведено на рис. 4.4.

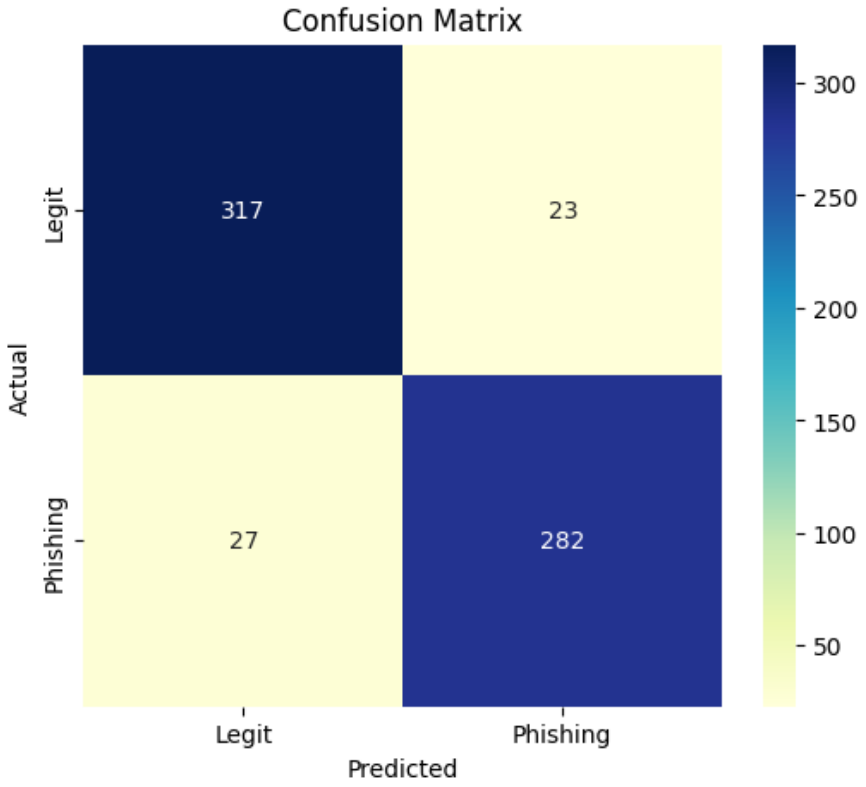


Рисунок 4.4 – Матриця невідповідностей

Після остаточного навчання моделі SVM, використовуючи оптимальні параметри, було досягнуто точності класифікації у 92.45%. Обидва класи даних демонструють схожі значення метрик precision, recall та F1-score зі значенням ~0.92, що свідчить про коректно налаштований баланс класів, а саме:

1) recall (повнота) – відображає частку фішингових сайтів, які модель змогла успішно виявити та віднести до класу «1»;

2) precision (точність) – відображає, яка частка прикладів, класифікованих як фішингові, дійсно є такою;

3) F1-score – зважений показник для оцінки балансу між помилками першого та другого роду.

Навчена модель коректно виявляє більшість випадків обох класів, при цьому показник False Positive Rate становить 0.0676, а False Negative Rate – 0.0841.

Для розуміння структури отриманих векторів ознак та перевірки наявності розділюваності між класами (наприклад, шкідливими та легітимними HTML-файлами), доцільно провести їхню візуалізацію у двовимірному просторі. Оскільки початкові вектори мають високу розмірність (1024 ознаки), перед візуалізацією доцільно зменшити розмірність до 2D-простору. Це дозволяє зберегти ключові характеристики розподілу даних та виявити можливі кластери чи закономірності. Для цього використовують методи зменшення розмірності, наприклад, t-Distributed Stochastic Neighbor Embedding (t-SNE), які зберігають глобальну або локальну структуру даних відповідно.

На рис. 4.5 наведено приклад такої візуалізації: кожна точка відповідає одному HTML-документу, кольором позначено клас (шкідливий / легітимний). Ця ілюстрація дозволяє наочно оцінити якість отриманих ознак і потенціал моделі до класифікації. Візуалізація показує розподіл фішингових і легітимних векторів на окремі кластери, що підтверджує здатність класифікатора розпізнавати відмінності між класами.

У табл. 4.1 наведено порівняльний аналіз запропонованого методу з сучасними підходами до виявлення фішингу. Важливо зазначити, що пряме кількісне порівняння методів машинного навчання є коректним лише за умови їх тестування на ідентичних наборах даних з використанням однакових метрик. Оскільки у відкритому доступі відсутній єдиний стандартизований датасет, який би використовувався всіма дослідниками, наведені в таблиці показники точності для наявних методів наведено згідно з результатами, опублікованими авторами у відповідних статтях [228,229,230]. У цих роботах використовувалися набори даних, подібні за обсягом (від 3 до 5 тисяч вебсторінок) до вибірки, застосованої в нашому дослідженні (4700 HTML-файлів), що дозволяє зробити якісне порівняння підходів, хоча абсолютні значення точності слід інтерпретувати з певним застереженням.

Запропонований у цій роботі метод тестувався на об'єднаному датасеті з 3000 легітимних та 1700 фішингових HTML-файлів [227]. Як основна метрика

для порівняння обрано Accuracy (загальна точність класифікації), оскільки вона є найбільш поширеною в проаналізованих наукових джерелах.

У порівнянні враховувалися такі характеристики методів: точність класифікації, вимоги до ресурсів, чутливість до обходу захисту та придатність до інтеграції в реальні системи (наприклад, у браузери або у пошукові проксі).

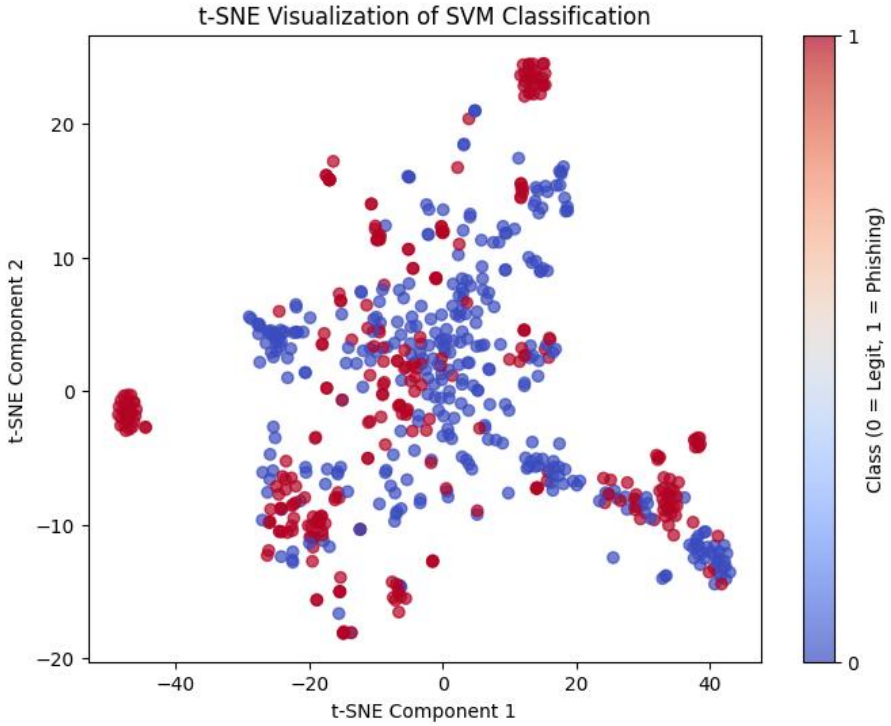


Рисунок 4.5 – Візуалізація розподілу векторів ознак

Таблиця 4.1 – Порівняльний аналіз запропонованого методу із сучасними аналогами

Підхід	Точність	Примітки
Запропонований підхід SVM + візуальне подання байт-коду HTML	92.45 %	Метод базується на перетворенні HTML-коду у байтовий потік з подальшою візуалізацією у графічне зображення розміру 128x128 пікселів. Аналізуються нормалізовані гістограми яскравості з чотирьох сегментів зображення, що дозволяє виявляти структурні патерни фішингових сторінок незалежно від текстового наповнення.

Гібридне машинне навчання на основі URL-адреси [228]	98.12 %	Метод аналізує тільки URL-адресу (довжину, наявність символів, домен, піддомен тощо). Це робить його вразливим до фішингових сайтів, які використовують легітимну або замасковану URL-адресу. Не аналізує вміст сторінки. Не можна виявляти клон-сайти й атаки через візуальну схожість. Має меншу стійкість до адаптивних атак
Функція на основі URL та HTML з використанням моделі штучних нейронних мереж (ANN) з MobileBERT [228]	86… 96%	Метод вимагає значні обчислювальні потужності (часто задіюється GPU), що утруднює його реальне застосування. Структура сторінки може враховуватися не повністю через особливості реалізації методу
Машинне навчання [229]	82.7…98.3%	Метод ґрунтується на аналізі тексту, що залишає зловмисникові простір для можливих маніпуляцій, структура файлу не враховується
Машинне навчання [230]	89.2 %	Не застосовується аналіз структури сторінки

Це дозволяє об’єктивно оцінити переваги запропонованої моделі та обґрунтувати її практичне застосування як ефективного компонента інфраструктури кіберзахисту.

Запропонований метод на основі SVM та візуалізації HTML демонструє конкурентну точність проти сучасних аналогів, при цьому вирізняється низкою ключових переваг: швидкістю обробки, незалежністю від текстового контенту та можливістю виявляти візуально схожі підроблені сайти. На відміну від методів, які аналізують лише URL [228] або текстовий вміст [229], такий підхід забезпечує комплексний аналіз структури сторінки, що робить його менш вразливим до адаптивних атак. Хоча окремі алгоритми демонструють вищу точність (до 98.12%, як в [231]), вони мають суттєві обмеження – залежність від ознак URL, високу ресурсоємність або нездатність виявляти клоновані

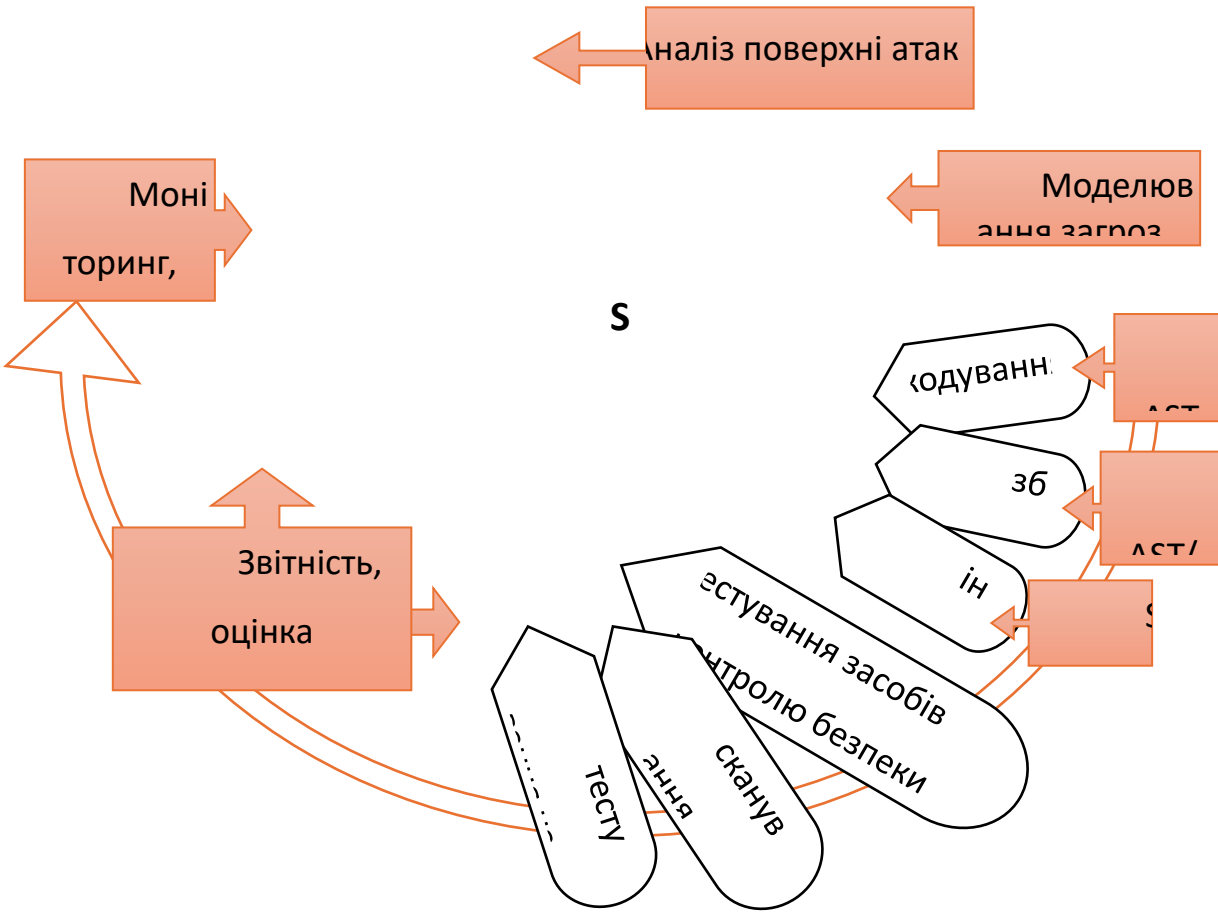
сайти. Тому запропоноване рішення є оптимальним балансом між точністю, продуктивністю та універсальністю для реальних умов застосування.

4.6. Інтеграція розроблених методів у процеси тестування та забезпечення безпеки вебзастосунків

Розвиток сучасних вебзастосунків супроводжується зростанням кількості та складності кіберзагроз, що вимагає не лише розробки нових методів їх виявлення і протидії, а й перевірки ефективності таких підходів у реальних умовах функціонування інформаційних систем. У попередніх розділах дисертаційної роботи представлено методи стеганоаналізу на основі машинного навчання і математичних перетворень, спрямовані на виявлення прихованих каналів та запобігання фішинговим атакам. Проте для інтеграції цих результатів у практику забезпечення безпеки вебзастосунків необхідне їх поєднання із сучасними процедурами тестування та оцінки захищеності.

Проведений аналіз сучасних методів та інструментів тестування вебзастосунків виявив наявність різних підходів і розмаїття засобів для захисту програмних продуктів. З такою кількістю підходів до тестування безпеки вебзастосунків доволі не просто зорієнтуватися і зрозуміти, які саме методи використовувати або коли та в якій послідовності застосовувати ті чи інші засоби тестування. Нерідко команди з тестування безпеки на практиці зіштовхуються з високими вимогами, але застарілими інструментами й методами безпеки, розробленими для «дохмарної епохи». Досвід показує, що за умов зростання ризиків та небезпек через зацікавленість зловмисників у зламі клієнтських систем не існує єдиної методики, яка б могла ефективно розв’язати всі проблеми захисту. Тестування безпеки традиційно проводять після розгортання коду [232, 233]. Однак вже на початкових етапах SDLC, принаймні під час розробки коду, варто планувати, організовувати та долучати тестування безпеки. На рис. 4.6 показано місце використання різних

інструментів тестування безпеки для виконання тестування безпеки у життєвому циклі ПЗ.



.

Рисунок 4.6 – Інтеграція тестування безпеки у життєвий цикл розробки ПЗ

Під час планування та аналізу вимог до ПЗ паралельно командою тестування безпеки визначаються обсяги, вимоги, передумови та терміни тестування безпеки. Визначення вимог до продукту зводиться до компромісу між бізнес-потребами клієнта та рівнем безпеки, необхідним для захисту його або її активів. На цьому етапі проводять активну (через контактування із

замовниками, сканування їх мережі) та пасивну (через соціальні мережі, Google Hacking) розвідку для збору інформації про організацію [9, 10].

На наступному етапі SDLC разом з моделюванням та проєктуванням ПЗ виконують аналіз поверхні атак та моделювання можливих загроз до створюваного ПЗ. На цьому етапі аналізують здобуту на попередніх етапах інформацію та визначають можливі ефективні методи та способи атак на організацію.

Під час розробки ПЗ за допомогою інструментів SAST та SCA аналізується вихідний код задля виявлення можливих недоліків. Тести безпеки можуть проводити під час збирання та фіксації гілок і версій, перевіряючи відомі небезпечні шаблони у вихідному коді перед додаванням їх у репозиторій коду або об’єднанням з основною гілкою. На рис. 4.7 наведено інструменти тестування безпеки відповідно до фаз SDLC за рекомендацією OWASP [234].

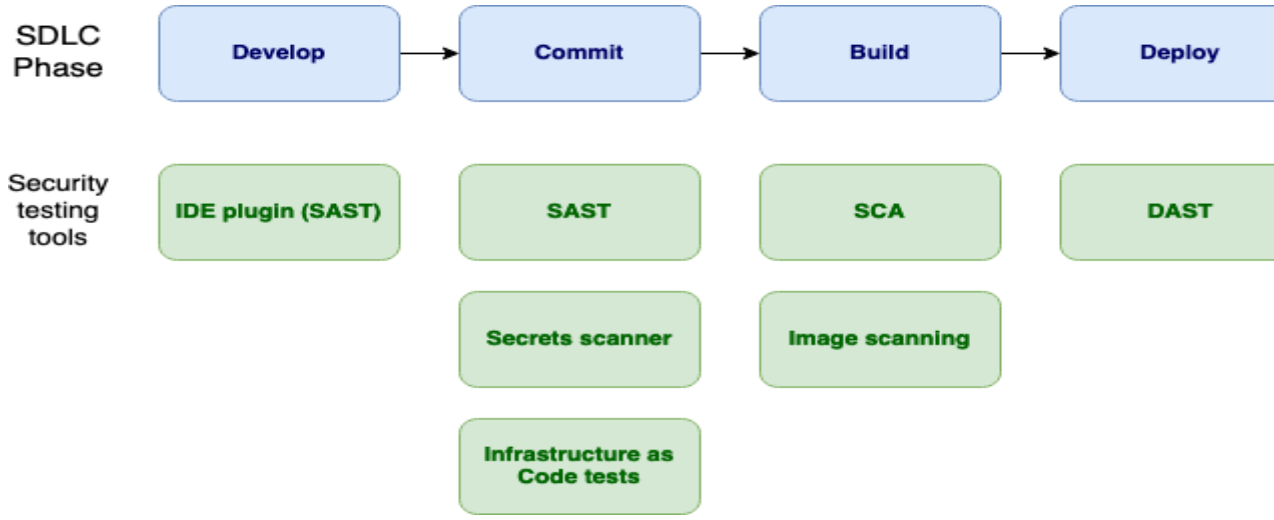


Рисунок 4.7 – Інструменти тестування безпеки за фазовою діаграмою SDLC відповідно до рекомендації фундації OWASP

Аналіз захищеності коду, який проводять експерти з безпеки, та однорангові перевірки коду призначені для усунення помилок на початкових етапах розробки. Вони доволі ефективні для виявлення логічних помилок, незалежно від того, чи є ці помилки навмисними, чи ні. Крім того, на цьому

етапі має відбуватися контроль, налаштування та оновлення засобів контролю безпеки, щоб переконатися, що вони розроблені належним чином і фактично забезпечують необхідний рівень безпеки. Також на цьому етапі періодично виконують сканування вразливостей та тестування на проникнення (penetration test, pen test). Мета тестування на проникнення полягає в тому, щоб пентестер (тестувальник безпеки на проникнення або етичний хакер) отримав несанкціонований доступ до ІТ-системи шляхом використання вразливостей ПЗ для імітації намірів зловмисного хакера [9, 10].

Грамотна програма управління вразливостями передбачає періодичне тестування на проникнення за такою логікою: сканування вразливостей виявляє громіздкий список вразливостей ПЗ, оцінка вразливості розміщує вразливості в пріоритетному списку, програма керування вразливістю виправляє ці вразливості за пріоритетом, а тест на проникнення зрештою перевіряє ці виправлення. Penetration test є обов'язковим або рекомендованим відповідно до певних положень деяких законодавчих актів, зокрема PCI-DSS, HIPAA, Sarbanes-Oxley та інших. По суті, тестування на проникнення є складовою повного аудиту безпеки. Наприклад, стандарт безпеки даних індустрії платіжних карток (PCI-DSS) вимагає тестування на проникнення щорічно або разом із будь-якими серйозними змінами в мережі чи то оновленням застосунків [233]. Адже зміни в ІТ-інфраструктурі часто створюють прогалини в безпеці. Тому система безпеки середовища потребує застосування брандмауерів, фільтрації вмісту, моніторингу журналів тощо. Крім того, обов'язково тестування на проникнення виконують після виявлених порушень для того, щоб впевнитися, що вразливість нейтралізована, а також щоб виявити будь-які додаткові наявні вразливості та діри у коді.

Інструменти DAST використовують переважно під час тестування та розгортання ПЗ, що дозволяє автоматизувати тестування безпеки на цих етапах. DAST здатне виявляти уразливості перевірки вхідних даних, ін'єкції SQL та XSS, проблеми автентифікації, помилки конфігурації, слабкі шифри

[16]. Також автоматизоване тестування безпеки може бути побудоване за допомогою інструментів IAST. Адже інтерактивні інструменти безпеки рекомендовано розгортати у середовищі контролю якості, яке запускає автоматизовані функціональні тести, що таким чином може допомогти уникнути людської помилки та переконатися, що увесь функціонал програми перевірено, а виявлені вразливості виправлено [83].

На етапі верифікації ПЗ має виконуватись перевірка відповідності продукту вимогам специфікації та необхідному рівню безпеки. Хоча інтеграція, регресія та модульне тестування корисні для функціональної оцінки, важливими складовими тестування безпеки є: 1) тестування засобів контролю безпеки, 2) сканування вразливостей і 3) тестування на проникнення [232]. Засоби контролю безпеки необхідно протестувати, щоб переконатися, що вони розроблені належним чином і фактично забезпечують необхідний рівень безпеки. Автоматизоване сканування вразливостей виконують періодично задля виявлення та усунення вразливостей до випуску продукту. Крім того, на цьому етапі рекомендується провести фазинг тестування (fuzzing, fuzz testing), щоб перевірити, чи правильно програма реагує як на очікувані, так і на неочікувані (випадкові, fuzz) вхідні значення з метою виявлення переповнення буфера, помилок перевірки введення, а також інших недоліків безпеки [88]. Тестування на проникнення, яке проводять експерти з безпеки, є ефективним для виявлення складних уразливостей і, як наслідок, плідно доповнює сканування вразливостей, яке виконується автоматизованими інструментами.

Під час розгортання протестованого ПЗ (pre-prod) виконують остаточну перевірку безпеки і, за потреби, сертифікацію ПЗ щодо рівня його безпеки; приймається рішення щодо виведення продукту на ринок. Доцільно на цьому етапі розробити план реагування на інциденти та передбачити сценарії подальшого керування змінами. Як би не намагалась компанія зробити створюване нею програмне забезпечення ідеальним, перевірити і вичистити від негараздів, помилки в ньому можуть траплятися, як і упущення та інші

небажані речі, а з часом можуть виявлятися нові вразливості. Тому готовий план реагування на інциденти має важливе значення для економії часу та грошей. З іншого боку, працездатність та продуктивність кожного готового програмного продукту у виробничому середовищі необхідно відстежувати в часі та за потреби оновлювати, виправляти виявлені помилки кодування та вразливості з урахуванням рекомендацій NIST (наприклад, NIST SP 800-160 [235], SP 800-100 [236]), матеріалів OWASP [234] і вимог серії ISO/IEC 27000 [237]. Проблеми з безпекою виникають і будуть виникати попри всі профілактичні зусилля, наявні інструменти та процеси. Тому важливо підготувати спеціальну робочу групу зі встановленими ролями та обов'язками, щоб визначити план пом'якшення загроз та виконати його якнайшвидше. Проведення імітації надзвичайних ситуацій і випробування процедури допомагає підготувати таку команду до аварійного відновлення і спланувати найгірший сценарій. Наразі все більше організацій розуміють важливість впровадження плану аварійного відновлення для захисту від втрати даних або руйнування IT-інфраструктури.

RASP та WAF використовуються після випуску продукту, що робить їх більш орієнтованими на безпеку, як порівняти з іншими відомими для тестування інструментами. Важливими складовими на етапі підтримки та обслуговування ПЗ є звітування про виявлені вразливості та вторгнення, а також моніторинг безпеки. Звіти про тестування на проникнення допоможуть оцінити потенційний вплив на організацію та запропонувати контрзаходи для пом'якшення ризиків. Організаціям слід пам'ятати, що передача продукту – це не кінець процесу. Має бути певна форма моніторингу. Пентестери мають регулярно перевіряти ПЗ, особливо після його оновлень. Безпека ПЗ та його розробка наразі не сумісна з підходом «встановив і забув».

Щоб адаптуватися до динамічної та неоднорідної природи інтернету та якнайкраще забезпечити захист вебзастосунків, ефективним є комплексний і збалансований підхід до тестування їхньої безпеки та вибору відповідних засобів. На архітектурному рівні розроблені у цій роботі методи виявлення

фішингу та стеганографії найкраще інтегрувати у рішення класу RASP та WAF. На рис. 4.8 показано, як запропоновані методи вбудовуються в єдиний контур безпеки.

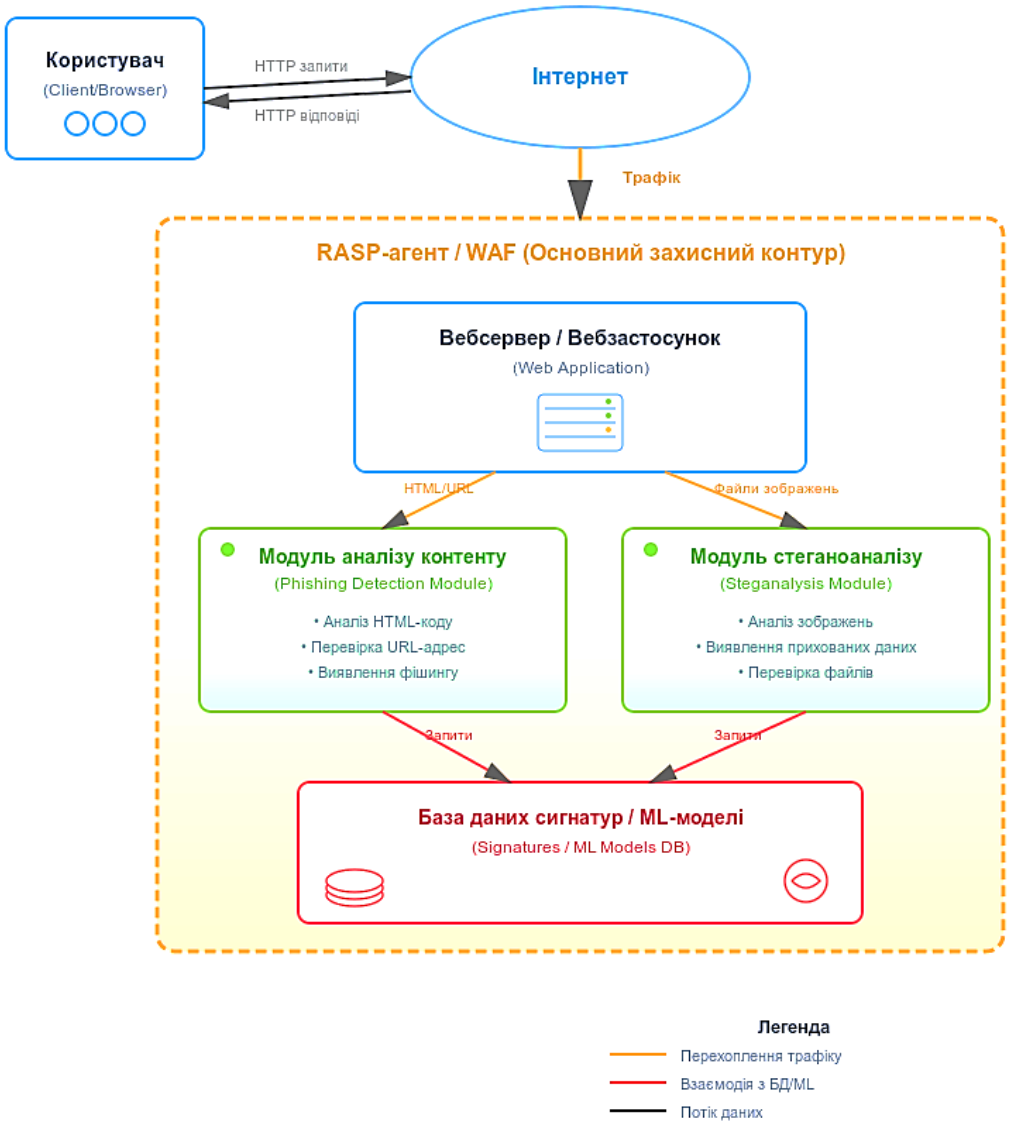


Рисунок 4.8 – Архітектурна модель інтеграції модулів стеганоаналізу та виявлення фішингу в контур Runtime Security

На схемі зображено архітектуру захисту, де весь трафік між користувачем та вебзастосунком проходить через захисний екран (WAF або агент RASP), який не лише виконує стандартні функції безпеки, а й містить два спеціалізовані модулі, розроблені в рамках дисертації:

1. **Модуль аналізу контенту (Phishing Detection Module** отримує на аналіз вихідний трафік (HTML-сторінки, URL-адреси) та через використання методів машинного навчання виявляє фішингові загрози в реальному часі;

2. **Модуль стеганоаналізу (Steganalysis Module)** аналізує файли зображень, які передаються у вхідному чи вихідному трафіку, на наявність прихованих каналів передавання даних.

Така інтеграція дозволяє доповнити традиційні засоби захисту інтелектуальними механізмами, здатними виявляти складні, замасковані атаки безпосередньо під час експлуатації вебзастосунку.

Динаміка роботи розробленої системи аналізу загроз у формі UML-діаграм послідовності для виявлення замаскованих загроз інтегрованими засобами захисту для різних сценаріїв показана на рис. 4.9 та 4.10.

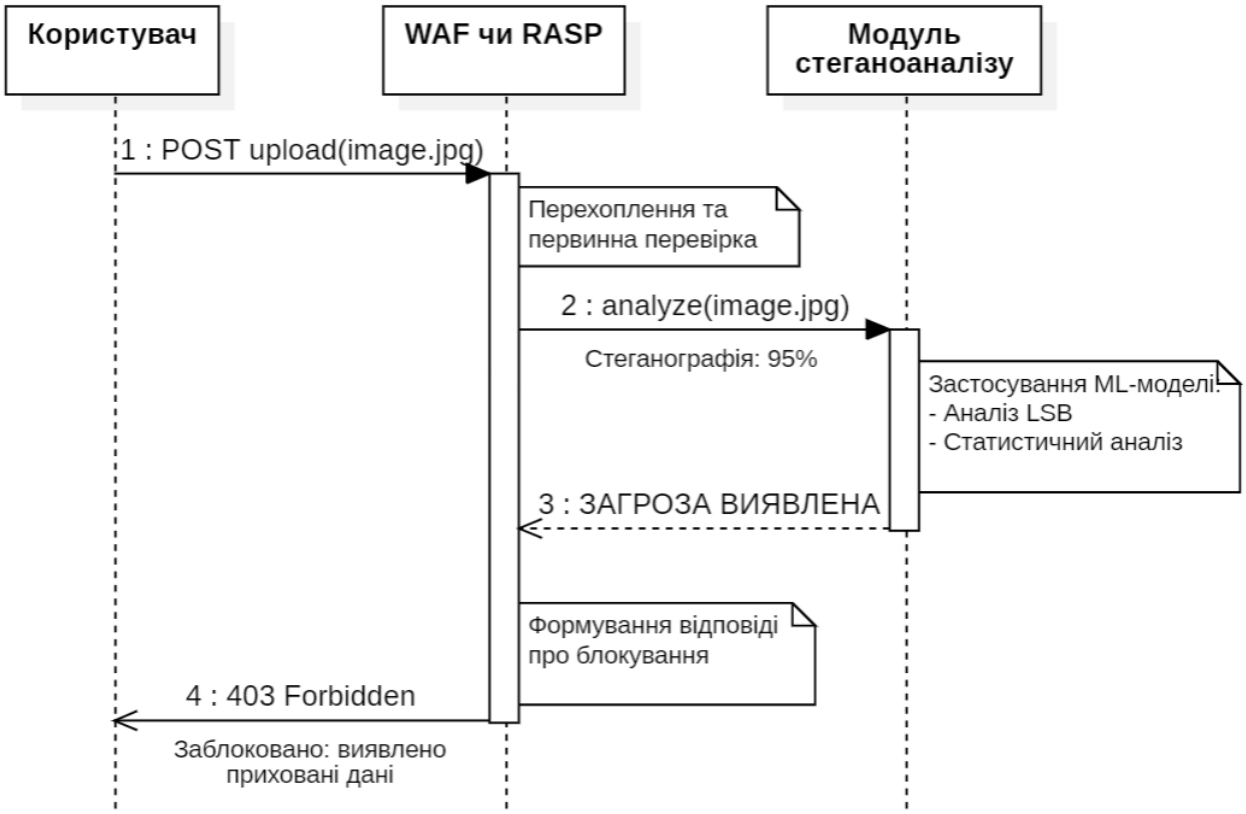


Рисунок 4.9 – UML-діаграма послідовності виявлення замаскованих загроз інтегрованими засобами захисту для сценарію завантаження та аналізу зображення

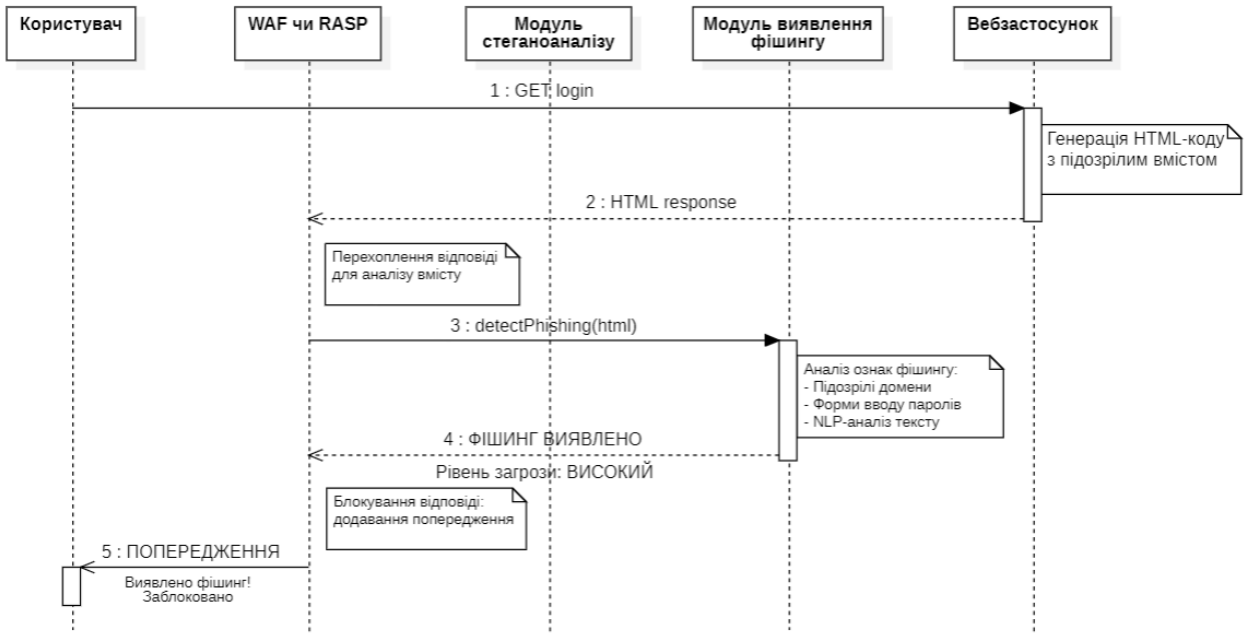


Рисунок 4.10 – UML-діаграма послідовності виявлення замаскованих загроз інтегрованими засобами захисту для сценарію запиту підозрілої сторінки

На рис. 4.9 зображено сценарій виявлення замаскованих загроз під час завантаження зображення, що ілюструє взаємодію між користувачем, інтегрованими засобами захисту (WAF або RASP) та спеціалізованим модулем стеганоаналізу. Центральну роль у цьому процесі відіграє модуль стеганоаналізу, який виконує глибоку перевірку вмісту зображення на предмет прихованих даних, що можуть становити загрозу безпеці. Після того як користувач надсилає POST-запит із файлом зображення, WAF або RASP-система перехоплює запит та ініціює первинну перевірку. На цьому етапі файл передається до модуля стеганоаналізу, який застосовує машинне навчання, аналіз найменш значущих бітів (LSB) та статистичні методи для виявлення ознак стеганографії. Якщо результат аналізу свідчить про наявність прихованих даних, модуль повертає висновок про виявлену загрозу. На основі цього висновку захисна система блокує запит і надсилає користувачеві повідомлення про помилку 403, що сигналізує про недопустимість передачі

потенційно небезпечного контенту. Тут модуль стеганоаналізу виступає як критичний елемент у ланцюгу кіберзахисту, забезпечуючи інтелектуальне виявлення прихованих загроз у мультимедійному трафіку ще до того, як вони можуть бути використані для шкідливих цілей.

На рис. 4.10 показано сценарій реагування на запит користувача до потенційно фішингової вебсторінки, в якому інтегровані засоби захисту взаємодіють із модулями стеганоаналізу та виявлення фішингу для виявлення замаскованих загроз. Ключову роль у цьому процесі відіграють два аналітичні компоненти, які працюють послідовно та доповнюють один одного. Після того як користувач ініціює запит до вебзастосунку, останній формує HTML-відповідь, яка може містити приховані елементи або ознаки фішингової активності. WAF або RASP-система перехоплює цей вихідний трафік ще до його передачі користувачеві, забезпечуючи точку контролю для подальшого аналізу. HTML-код передається до модуля виявлення фішингу, який виконує комплексну перевірку з використанням ознак фішингу, виявлення форм введення паролів та лінгвістичного аналізу тексту. У разі виявлення високого рівня загрози, система може модифікувати відповідь або повністю її заблокувати, попередивши користувача про ризик. Модуль стеганоаналізу в цьому сценарії виконує додаткову функцію – він аналізує вміст HTML на предмет прихованих даних, які можуть бути вбудовані у зображення або інші мультимедійні елементи сторінки. Його роль полягає у виявленні стеганографічних технік, що використовуються для маскування шкідливого контенту, який не може бути виявлений традиційними засобами. Завдяки цьому модуль забезпечує глибший рівень перевірки, дозволяючи системі захисту реагувати не лише на явні ознаки фішингу, а й на приховані загрози, які можуть бути частиною складних атак. Таким чином, ця UML-діаграма демонструє узгоджену роботу захисної інфраструктури, де модулі стеганоаналізу та виявлення фішингу функціонують як інтелектуальні фільтри, що дозволяють виявити та нейтралізувати замасковані загрози ще до того, як вони досягнуть користувача.

Запропонований у роботі підхід доповнює SAST/DAST/IAST і створює єдиний контур безпеки, який охоплює весь життєвий цикл (рис. 4.11).

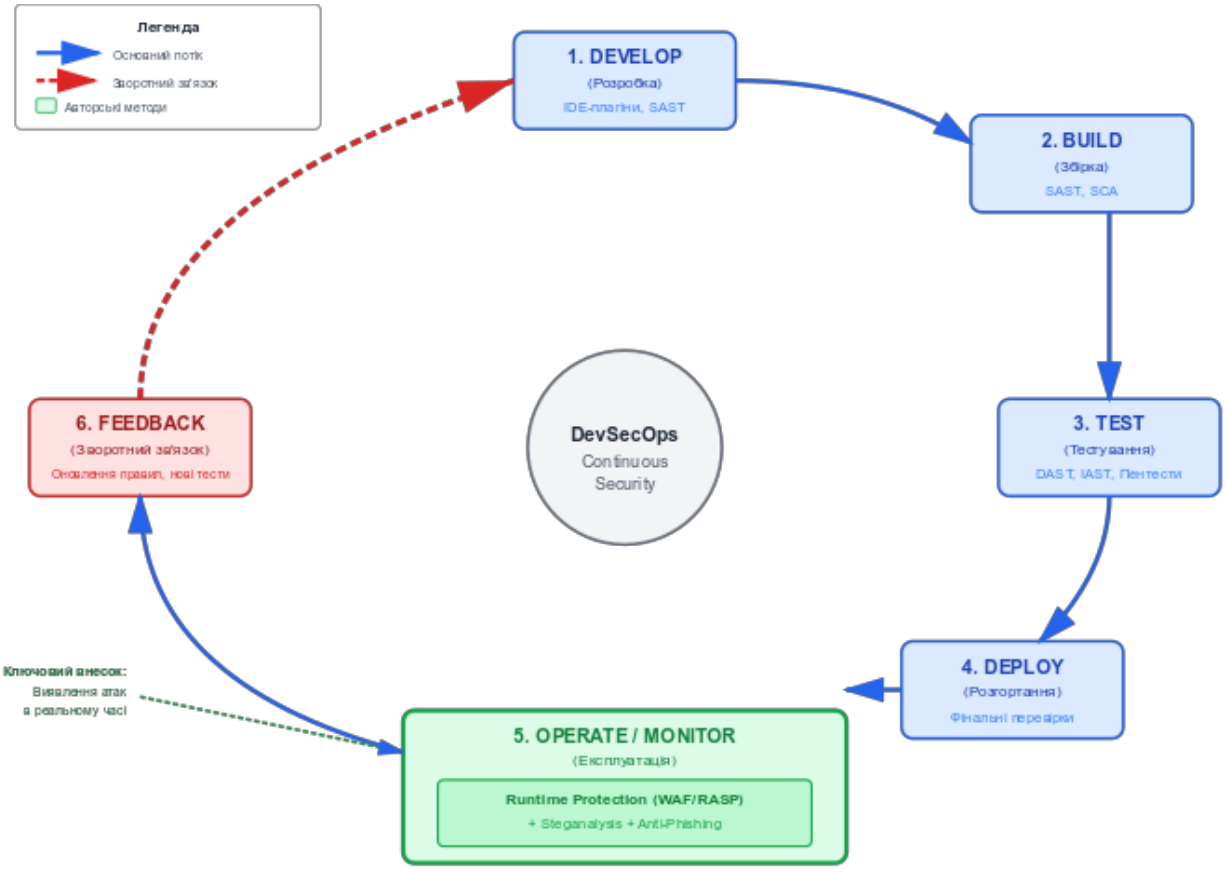


Рисунок 4.11 – Концептуальна модель єдиного контуру безпеки в парадигмі DevSecOps

Запропонована концептуальна модель DevSecOps як неперервного циклу охоплює всі фази життєвого циклу ПЗ – від розробки до експлуатації – з інтеграцією безпеки на кожному етапі. Ліва частина циклу репрезентує традиційні підходи до забезпечення безпеки до моменту релізу: на етапах Develop і Build застосовуються статичний аналіз коду (SAST) та перевірка сторонніх компонентів (SCA), а на фазах Test і Deploy – динамічне тестування (DAST), інтерактивний аналіз (IAST) і пентестування. Права частина циклу – Runtime Security – є критичною для сучасних вебзастосунків, оскільки саме тут реалізуються методи, розроблені в межах дисертаційного дослідження. Вони інтегруються в механізми захисту WAF/RASP і забезпечують активне виявлення загроз у реальному часі, зокрема фішингових атак і прихованих

даних, які маскуються за допомогою стеганографії. Ці методи не лише блокують шкідливу активність, а й формують аналітичну основу для подальшого вдосконалення системи. На етапі експлуатації як ключовий захисний механізм впроваджено блок Runtime Protection з модулями стеганоаналізу та виявлення фішингу. Ці модулі функціонують у реальному часі, виявляючи приховані загрози та ознаки соціальної інженерії, що дозволяє оперативно реагувати на атаки, не виявлені на попередніх етапах.

Інноваційність моделі полягає в тому, що дані, отримані під час моніторингу, не залишаються ізольованими – вони повертаються у вигляді зворотного зв'язку до фази розробки. Механізм зворотного зв'язку (Feedback) перетворює однонапрямлений процес у самопідсилювану систему. Дані про атаки, виявлені на етапі експлуатації, не залишаються локальними – вони повертаються до команди розробників, де використовуються для оновлення правил аналізу (SAST/DAST), створення нових тестів та архітектурних покращень. Це дозволяє кожній наступній ітерації продукту бути більш захищеною, адаптованою до нових загроз і технологічно зрілою. Таким чином, забезпечується адаптивність системи: нові виявлені вектори атак використовуються для вдосконалення правил безпеки, оновлення тестових сценаріїв та покращення коду. Такий підхід принципово відрізняється від традиційних моделей, де безпека часто реалізується як окремий етап або зовнішній процес.

У DevSecOps безпека є невіддільною частиною кожної фази, а зворотний зв'язок перетворює її з реактивної на проактивну. Завдяки цьому концептуальному рішенню DevSecOps стає динамічною екосистемою, здатною до самонавчання та еволюції. Інтеграція модулів стеганоаналізу та фішинг-детекції у фазу експлуатації – це стратегічний крок до побудови стійкої, самокоригувальної архітектури безпеки. Запропонована схема демонструє побудову цілісного, еволюційного контуру захисту, де DevSecOps перетворюється на інтелектуальну екосистему, здатну до самонавчання, адаптації та превентивного реагування. Інтеграція авторських методів у фазу

Operate/Monitor закладає основу для формування нової парадигми безпеки – динамічної, контекстно чутливої та орієнтованої на стійкість.

Висновки до розділу 4

Узагальнимо основні, отримані у розділі результати:

1. Виконана класифікація фішингових атак та аналіз сучасних тенденцій у фішингових атаках свідчить про їхню еволюцію та збільшення складності. Застосування ШІ та новітніх технологій дозволяє зловмисникам створювати переконливі, важкі для виявлення фішингові схеми. Надалі організаціям доцільно впроваджувати комплексні стратегії кібербезпеки, включно з навчанням користувачів, використанням багатофакторної автентифікації та постійним оновленням систем безпеки, щоб ефективно протистояти загрозам.

2. Запропонований метод поєднує машинне навчання на основі SVM з інноваційним графічним поданням байтів вебсторінки, демонструє високу ефективність із точністю класифікації до 92.45%. Його ключова особливість полягає у перетворенні HTML-коду на візуальне подання, що дозволяє аналізувати структуру сторінки через призму графічних шаблонів. Такий підхід усуває обмеження традиційних текстових методів аналізу та забезпечує адаптивність до різноманітних типів фішингових атак, включно зі складними випадками імітації легальних ресурсів. Метод відрізняється практичною спрямованістю – він може бути ефективно інтегрований у сучасні браузери або серверні системи для оперативної перевірки вебконтенту. Його універсальність забезпечується аналізом на рівні байтів, що дозволяє виявляти аномалії незалежно від специфіки кодування чи структури документа. Важливою перевагою є збалансованість результатів, де високі показники точності та повноти свідчать про здатність системи мінімізувати кількість помилкових спрацьовувань та пропусків загроз. Цей підхід відкриває нові можливості для боротьби з сучасними фішинговими атаками, особливо за умов

стрімкого розвитку генеративних технологій, які ускладнюють виявлення штучно створених шкідливих ресурсів. Його практична цінність полягає у поєднанні високої точності, адаптивності та можливості впровадження в реальні системи кібербезпеки.

3. Розробка ефективного методу виявлення фішингових HTML-сторінок може розглядатися як один із компонентів загальної інфраструктури боротьби з фішингом. У реальних умовах подібні методи інтегруються у більш складні системи – зокрема, вони можуть бути частиною веббраузера, розширенням до нього або системою серверного захисту вебзастосунків. Така інтеграція дозволяє забезпечити своєчасну перевірку підозрілих сторінок під час перегляду користувачем та запобігти переходу на потенційно небезпечні ресурси. Крім безпосереднього блокування шкідливих сторінок, подібна система може автоматично формувати чорні списки URL-адрес, виявляти шаблони фішингових атак, а також надсилати сповіщення розробникам чи відповідним правоохоронним або аналітичним структурам. Це дозволяє не лише захищати окремих користувачів, а й оперативно реагувати на масові кампанії фішингу, оновлювати захисні механізми та сприяти координації зусиль у сфері кібербезпеки на міжорганізаційному рівні.

4. Сучасний захист вебзастосунків вимагає двоєдиного підходу, який поєднує безпеку розробки (DevSec) та безпеку експлуатації (Runtime Security). Інструменти SAST, DAST та IAST формують фундамент безпеки на етапі створення ПЗ, виявляючи та усуваючи вразливості до виходу продукту на ринок. Однак цього недостатньо для протидії динамічним атакам, які виникають безпосередньо під час роботи застосунку. Запропоновані в дисертації методи не конкурують з традиційними підходами, а органічно доповнюють їх, формуючи повний життєвий цикл захисту. Розроблені методи функціонують в рамках Runtime Security, тобто працюють вже на розгорнутому, функціонуючому застосунку, аналізують його вхідні та вихідні дані (трафік, файли, контент) в реальному часі для протидії конкретним загрозам: організації прихованих каналів передачі даних та фішинговим

атакам на кінцевих користувачів. Таким чином, якщо SAST/DAST/IAST – це захист застосунку, що формується під час розробки, то запропоновані нами методи – це захист, що постійно працює після його релізу, відловлюючи специфічні загрози, які уникли первинних засобів захисту.

5. Інтеграція запропонованих методів у сучасні практики DevSecOps є не лише можливою, а й необхідною. Їх основна цінність проявляється саме на етапі експлуатації: методи виявлення фішингових атак та стеганографічних повідомлень можуть бути інтегровані у рішення класу RASP (Runtime Application Self-Protection) та WAF (Web Application Firewall). Це дозволить створювати «розумні» захисні екрани, здатні не лише блокувати відомі шаблони атак, а й виявляти складні замасковані загрози безпосередньо у робочому трафіку. Такий комбінований підхід створює єдиний контур безпеки. Він охоплює весь життєвий цикл застосунку: від написання першого рядка коду (де працює SAST) до його неперервної експлуатації (де працюють RASP та наші методи). Це дозволяє не лише значно зменшити кількість уразливостей, які виникають на етапі розробки, а й ефективно захищатися від нових загроз, які з'являються вже після релізу.

Загалом механізми автоматичного виявлення фішингових сторінок є невіддільною частиною загальної безпеки вебзастосунків і цифрового простору в цілому. Їх використання дозволяє створювати багаторівневий захист, який охоплює як кінцевого користувача, так і серверну інфраструктуру вебресурсів.

ВИСНОВКИ

Удосконалення безпеки є неперервним процесом, необхідним для забезпечення високої продуктивності тестування безпеки. Моніторинг безпеки, реагування на інциденти та виявлення вразливостей – це дії, спрямовані на підвищення якості ПЗ з позиції безпеки. Дослідження, проведені в дисертаційній роботі, розв'язали одну з актуальних науково-практичних задач у сфері тестування кібербезпеки вебзастосунків. У роботі сформовано теоретичний базис і розроблено нові методи виявлення сучасних загроз, які охоплюють фішингові атаки та приховані канали передавання інформації, зокрема організовані на основі стеганографічних методів. Запропоновані підходи відрізняються від наявних аналогів високою ефективністю, адаптивністю та можливістю інтеграції у життєвий цикл розробки й експлуатації вебсистем. Це забезпечило підвищення рівня захищеності вебзастосунків від сучасних кіберзагроз і створило передумови для подальшого розвитку напрямку.

В рамках досягнення мети роботи були отримані такі результати:

1. Здійснено комплексний аналіз сучасного стану проблеми забезпечення безпеки вебзастосунків у контексті життєвого циклу розробки ПЗ (CI/CD). Виявлено, що ключовими загрозами є фішингові атаки та приховані канали передачі інформації, організовані засобами стеганографії. Показано, що наявні підходи до тестування безпеки не забезпечують достатнього рівня захисту від зазначених загроз, що зумовлює потребу розроблення нових методів і засобів їх виявлення та нейтралізації.

2. Розроблено теоретичний базис для виявлення прихованих каналів, побудованих на основі концепції кодового управління. Запропоновано підхід, що ґрунтується на аналізі статистичних властивостей трансформант перетворення Уолша-Адамара, який забезпечує зниження обчислювальної складності та придатність до інтеграції у вебзастосунки. Це створює

можливості для ефективного виявлення стеганографічних каналів у реальному часі та підвищення рівня захисту інформаційних систем.

3. Узагальнено розроблений теоретичний базис виявлення прихованих каналів у просторі перетворення Уолша-Адамара на випадок каналів, створених за допомогою SVD UV-методів. Показано, що аналіз характерних зсувів у спектрі трансформант дозволяє ефективно виявляти стеганографічні вбудовування при суттєвому зниженні обчислювальної складності. Проведені експерименти підтвердили потенційне скорочення обчислювальних витрат у 16-21 раз (до 16 разів – для несліпого аналізу, до 21 разу – для сліпого), що робить запропонований підхід придатним для реалізації у вебсередовищах з обмеженими ресурсами та відкриває перспективи створення високоефективних систем захисту мультимедійних даних у реальному часі.

4. Розроблено та експериментально обґрунтовано комплекс методів стеганоаналізу на основі машинного навчання для виявлення сучасних типів стеганографічних повідомлень із кодовим управлінням та на основі SVD UV, орієнтованих на роботу в реальному часі та інтеграцію у Runtime Security вебзастосунків. Для каналів із кодовим управлінням запропоновано підхід на основі аналізу огинаючої гістограм трансформант Уолша-Адамара з використанням мережі Multilayer Perceptron, який забезпечив Accuracy ≈ 93.3%, Precision = 0.95, Recall = 0.98 та F1-score = 0.96. При цьому час класифікації одного зображення за наявності попередньо обчислених ознак становить лише ~0.0044 с (>200 зобр./с), що гарантує можливість застосування в умовах, наближених до реального часу. Додатково розроблено метод стеганоаналізу вбудовувань, створених із застосуванням SVD UV-методу на основі трансформант перетворення Уолша-Адамара та ансамблю SVM з апостеріорними ймовірностями. Експерименти підтвердили його високу ефективність: Accuracy = 90.57%, Precision = 0.96, Recall = 0.94, F1-score = 0.95. Разом ці результати доводять, що запропоновані методи перевершують більшість відомих аналогів як за рівнем точності, так і за продуктивністю,

забезпечують надійність і практичну придатність для сучасних систем інформаційної безпеки.

5. Сформовано ансамблевий підхід до стеганоаналізу сучасних стеганоалгоритмів шляхом інтеграції розроблених теоретичних засад із сучасними алгоритмами машинного навчання. Проведені експерименти засвідчили, що поєднання ансамблю з трьох згорткових нейронних мереж на основі ознак із трансформант перетворення Уолша-Адамара забезпечує підвищення точності детектування прихованих каналів для методів з кодовим управлінням до 94.3% (Precision = 0.98; Recall = 0.96; F1-score = 0.97).

6. Інтегровано алгоритми штучного інтелекту та машинного навчання в процедури автоматизованого тестування безпеки вебзастосунків, що дозволило створити адаптивні методи виявлення фішингу з високою точністю детектування та практичною придатністю для вбудування у CI/CD-конвеєри й Runtime-контури захисту. Запропонований підхід, який перетворює HTML-код у візуальне (байтове) подання і класифікує його за допомогою SVM (RBF), продемонстрував: точність класифікації до 92.45%; метрики precision/recall/F1 для обох класів знаходяться близько ~0.92, що вказує на збалансовану роботу моделі. Запропонований метод показує конкурентну точність при помітно нижчих вимогах до інтерпретованості та хорошій швидкодії, як порівняти з важкими моделями (наприклад, MobileBERT або повноцінним рендерингом + CNN), що робить його кращим компромісом точність / продуктивність для застосування в реальному середовищі.

7. Проведено порівняльний аналіз ефективності запропонованих методів стеганоаналізу з відомими аналогами за базовими критеріями (точність, чутливість/специфічність, стійкість до перетворень і швидкодія) у контексті тестування безпеки вебзастосунків, який дозволив встановити, що:

– єдиний відомий у відкритих джерелах метод стеганоаналізу стеганоповідомлень з кодовим управлінням на основі аналізу трансформант Уолша-Адамара (традиційний підхід) показує точність 87.7%, що на 6.6% менше від запропонованого методу. При цьому класичні статистичні методи

майже неефективні проти сучасних вбудовувань. RS-аналіз на вибірці з 530 зображень виявив лише 50/530 (9.43%) при порозі 5% і 18/530 (3.4%) при порозі 10%; χ^2-аналіз – 1/530 (0.18%); аналіз пар і утиліта StegExpose майже не детектують вбудовування, виконане методом з кодовим управлінням. Це підкреслює принципову слабкість класичних тестів проти адаптивних/структурованих стеганопроцедур;

– порівняльний аналіз підтвердив, що запропонований метод виявлення фішингових атак демонструє одну з найвищих серед сучасних підходів точність класифікації 92.45%, що на 6.45–9.75% більше аналогів. При цьому метод характеризується низькою обчислювальною складністю та універсальністю застосування. Зазначимо, що, хоча окремі URL-орієнтовані методи заявляють точність до 98.12%, вони є концептуально вразливими до маскування адреси, не здатні аналізувати вміст сторінки й виявляти клоновані ресурси, що суттєво знижує їх практичну надійність.

8. Розробка підходів до інтеграції запропонованих стеганоаналітичних методів і методу захисту від фішингу у життєвий цикл вебзастосунків у межах концепції DevSecOps є логічним завершенням та практичною реалізацією теоретичних розробок, представлених у дисертації. Проведений аналіз сучасних інструментів та практик тестування безпеки підтверджує, що:

– інтеграція є необхідною та можливою. Запропоновані методи не конкурують з традиційними підходами (SAST, DAST, IAST), а органічно доповнюють їх, формуючи повний життєвий цикл захисту. Вони призначені для роботи на етапі експлуатації, аналізуючи робочий трафік та контент для виявлення складних, замаскованих загроз;

– оптимальним механізмом інтеграції є впровадження у рішення класу RASP (Runtime Application Self-Protection) та WAF (Web Application Firewall). Це дозволить створити «розумні» захисні екрани, здатні виявляти фішингові атаки та стеганографічні повідомлення безпосередньо під час роботи застосунку;

– така інтеграція формує єдиний контур безпеки в межах DevSecOps, охоплюючи всі етапи життєвого циклу: від розробки (за допомогою SAST/DAST) до експлуатації (за допомогою RASP, WAF та запропонованих методів). Це забезпечує не лише проактивний захист на етапі створення ПЗ, але й реактивний захист від нових загроз, що виникають після релізу.

Загалом дисертаційна робота комплексно вирішує актуальну задачу підвищення кіберзахищеності вебзастосунків у контексті сучасних загроз, зокрема фішингових атак і прихованих каналів передавання інформації. Запропоновані теоретичні засади, алгоритмічні рішення та методи стеганоаналізу демонструють високу точність, продуктивність й адаптивність, що підтверджено експериментально. Вони не лише перевершують відомі аналоги за ключовими метриками, а й забезпечують практичну придатність для інтеграції у CI/CD-конвеєри, Runtime-контури захисту та системи класу RASP і WAF. Робота формує науково обґрунтований базис для побудови інтелектуальних засобів тестування безпеки, здатних ефективно виявляти складні, замасковані загрози в реальному часі. Це відкриває перспективи подальшого розвитку напрямку в межах концепції DevSecOps та сприяє формуванню єдиного контурного захисту вебзастосунків на всіх етапах їх життєвого циклу.

СПИСОК ВИКОРИСТАННИХ ДЖЕРЕЛ

1. Концептуальні основи захисту інформаційного суверенітету України : монографія / Задерейко О. В., Троянський О. В., Чанишев Р. І., Дика А. І. Одеса : Фенікс, 2022. 220 с. DOI: https://doi.org/10.32837/11300.22733

2. Трофименко О. Г., Дика А. І. Проблеми тестування вебсайтів е-комерції // Інформаційні технології та менеджмент у вищій освіті та науці : матеріали міжнар. наук. конф. (м. Фергана, 28 листоп. 2022 р.). Izdevnieciba «Baltija Publishing», 2022. С. 195-199.

3. Трофименко О. Г., Дика А. І., Лобода Ю. Г. Аналіз уразливостей та проблем безпеки вебзастосунків. *Системні технології*. 2023. № 3(146). С. 25-37. DOI: https://doi.org/10.34185/1562-9945-3-146-2023-03

4. Логінова Н. І., Дика А. І., Янковський О. Г. Аналіз вразливостей систем керування базами даних // Європейський вибір України, розвиток науки та національна безпека в реаліях масштабної військової агресії та глобальних викликів XXI століття (до 25-річчя Національного університету «Одеська юридична академія» та 175-річчя Одеської школи права) : матеріали міжнар. наук.-практ. конф. (м. Одеса, 17 черв. 2022 р.). Одеса: Видавничий дім «Гельветика», 2022. Т. 1. С. 682-685. URL: https://hdl.handle.net/11300/19760

5. Манаков С. Ю., Дика А. І. Вибір сучасних засобів розробки мобільних застосунків // Інформаційне суспільство: проблеми та перспективи *:* матеріали VIII всеукр. наук.-практ. конф. (Одеса, 2 червня 2023 р.). Одеса: НУ «ОЮА», 2023. С. 86-89. DOI: https://doi.org/10.32837/11300.25263

6. Дика А. І. Сучасні інтерактивні інструменти тестування безпеки застосунків // Інформаційні технології: моделі, алгоритми, системи (ITMAS-2024) *:* матеріали V Всеукр. наук.-практ. інтернет-конф. (м. Миколаїв, 30-31 жовт. 2024 р.). Миколаїв : НУК імені адмірала Макарова, 2024. С. 188-190. URL: https://itconf.nuos.edu.ua/2024/publications/analysis-of-project-management-software/

7. Трофименко О. Г., Дика А. І., Лобода Ю. Г. Аналіз інструментів тестування вебзастосунків. *Кібербезпека: освіта, наука, техніка.* 2023. № 4(20). С. 62-71. DOI: https://doi.org/10.28925/2663-4023.2023.20.6271

8. Дика А. І. Регресійне тестування в Agile. *Кібербезпека в сучасному світі: актуальні виклики* : матеріали IV всеукр. наук.-практ. конф. (м. Одеса, 25 листоп. 2022 р.). Одеса : НУ «ОЮА», 2022. С. 44-48. URL: https://hdl.handle.net/11300/28986 ; DOI: 10.32837/11300.28986

9. Тестування та забезпечення якості програмних систем : навч.-метод. посіб. для підгот. здобув. вищої освіти галузі знань 12 «Інформаційні технології» / Нац. ун-т «Одес. юрид. академія»; уклад.: О. Г. Трофименко, А. І. Дика ; Одеса : Фенікс, 2024. 140 с. URL: https://hdl.handle.net/11300/27246 ; DOI: https://doi.org/10.32837/11300.27246

10. Тестування та забезпечення якості програмних систем : навч. посіб. для підгот. здобув. вищої освіти галузі знань 12 «Інформаційні технології» / О. Г. Трофименко, А. І. Дика; Нац. ун-т «Одес. юрид. академія». Одеса: Фенікс, 2024. 195 с. URL: https://hdl.handle.net/11300/27717. ISBN 978-617-8395-88-9.

11. Дика А. І., Трофименко О. Г., Ковальжі А. С. Інтелектуальне виявлення фішингових сторінок у середовищі вебзастосунків із застосуванням візуального подання. *Вчені записки ТНУ імені В.І. Вернадського*. Серія: Технічні науки. 2025. Т. 36(75), № 1. С. 298-308. DOI: https://doi.org/10.32782/2663-5941/2025.1.2/43

12. Дика А. І. Застосування контекстуальних ембеддінгів для виявлення ознак шкідливих введень // Інформаційне суспільство: проблеми та перспективи *:* матеріали X Всеукр. наук.-практ. конф. (м. Одеса, 23 трав. 2025 р.). Одеса : НУ «ОЮА», 2025. С. 158-160. DOI: https://doi.org/10.32837/11300.29842

13. Дика А. І., Лобода Ю. Г. Розвиток застосування WebAssembly у веброзробці // Європейські орієнтири розвитку України в умовах війни та глобальних викликів XXI століття: синергія наукових, освітніх та технологічних рішень: у 2 т. : матеріали міжнар. наук.-практ. конф. (м. Одеса,

19 трав. 2023 р.). Одеса : Юридика, 2023. Т. 1. С. 594-596. URL: https://hdl.handle.net/11300/27641

14. Dyka A. I. Intelligent methods for steganalysis of code-controlled covert messages in web applications. *Scientific notes of V. I. Vernadsky Taurida National University*. 2025. Vol. 36(75), № 2. P. 324-333. DOI: https://doi.org/10.32782/2663-5941/2025.2.2/44

15. Трофименко О., Дика А. І., Логінова Н., Задерейко О., Струк Н. Штучний інтелект у військових навчальних симуляторах. *Інформаційні технології та суспільство*. 2024. № 2(13). 2024. С. 89-95. DOI: https://doi.org/10.32689/maup.it.2024.2.13

16. Дика А. І. Застосування NLP-методів для виявлення XSS-атак // Європейські орієнтири розвитку України: нові виміри у воєнний час : матеріали міжнар. наук.-практ. конф. у 2 т. (Одеса, 16 травня 2025 р.). Одеса : Фенікс, 2025. Т. 2. С. 520-522. URL: https://hdl.handle.net/11300/30820

17. Трофименко О. Г., Дика А. І., Лобода Ю. В., Толокнов А. А., Бондаренко М. Є. Штучний інтелект у розробці ігор. *Кібербезпека: освіта, наука, техніка*. 2025. № 3(27). С. 109-119. DOI: https://doi.org/10.28925/2663-4023.2025.27.701

18. Трофименко О. Г., Лобода Ю. Г., Гура В. І., Дика А. І., Стрілець М. І. Інструменти штучного інтелекту для системного аналізу. *Вісник Херсонського національного технічного університету.* 2024. № 4. С. 349-357. DOI: https://doi.org/10.35546/kntu2078-4481.2024.4.46

19. Dyka A. I. Detection of SVD-based hidden data in web applications using Walsh-Hadamard transformant analysis. *Scientific notes of V. I. Vernadsky Taurida National University*. 2025. Vol. 36 (75), № 3. P. 110-118. URL: https://www.tech.vernadskyjournals.in.ua/journals/2025/4_2025/part_2/17.pdf

20. Dyka A. I. Detection of covert channels in web applications based on unimodality violation in the Walsh-Hadamard spectrum. *Informatics and Mathematical Methods in Simulation*. 2025. Vol.15, № 1. P. 5-14. DOI: https://doi.org/10.15276/imms.v15.no1.5

21. Трофименко О. Г., Дика А. І., Задерейко О. В., Шевченко О. В. Сфери застосування штучного інтелекту в розробці та тестуванні ігор. *Кібербезпека: освіта, наука, техніка*. 2025. № 1(29). С. 41-59. DOI: https://doi.org/10.28925/2663-4023.2025.29.832

22. Дика А. І. Роль штучного інтелекту у тестуванні безпеки вебзастосунків // Інформаційне суспільство: проблеми та перспективи : матер. IX всеукр. наук.-практ. конф. (м. Одеса, 24 трав. 2024 р.). Одеса : НУ «ОЮА», 2024. С. 153-156. DOI: https://doi.org/10.32837/11300.27842

23. Дика А. І. Тестування безпеки вебзастосунків за допомогою ChatGPT // Актуальні тенденції розвитку освіти, науки та технологій : матеріали VII міжнар. наук.-практ. конф. (м. Харків-Бахмут, 15 трав. 2024 р.) Т. 2. С. 28-30. URL: https://www.nnppi.in.ua/index.php/abit/2-uncategorised/270-naukovi-konferentsiyi

24. Дика А. І., Трофименко О. Г. Аналіз потенційних загроз засобами ChatGPT для тестування безпеки вебзастосунків // Європейські орієнтири розвитку України: науково-практичний вимір в умовах воєнних викликів : матеріали міжнар. наук.-практ. конф. (м. Одеса, 26 квіт. 2024 р.) Одеса : НУ «ОЮА», 2024. С. 876-878. URL: https://hdl.handle.net/11300/28232

25. Дика А. І., Максименко А. Ж. Штучний інтелект у веброзробці: революція персоналізації користувача // Кіберпростір в умовах війни та глобальних викликів XXI століття: теорія та практика : матеріали міжнар. наук.-практ. конф. (м. Одеса, 24 листоп. 2023 р.). Одеса : НУ «ОЮА», 2023. С. 174-176. DOI: https://doi.org/10.32837/11300.27179

26. Дика А. І. Тестування штучного інтелекту: ключові виклики, стратегії вдосконалення // *Електронні інформаційні ресурси: створення, використання, доступ :* матеріали міжнар. наук.-практ. інтернет-конф. (м. Вінниця, 20-21 листоп. 2023 р.). Вінниця : КЗВО «Вінницька академія безперервної освіти». С. 93-95.

27. Трофименко О. Г., Савєльєва О. С., Прокоп Ю. В., Логінова Н. І., Дика А. І. Soft skills для розробників програмного забезпечення. *Кібербезпека:*

освіта, наука, техніка. 2023. № 3(19). С. 6-19. DOI: https://doi.org/10.28925/2663-4023.2023.19.619

28. Dzhangarov A. I., Pakhaev K. K., Potapova N. V. Modern web application development technologies. *IOP Conference Series: Materials Science and Engineering*. 2021. Vol. 1155, №. 1. P. 012100. DOI: 10.1088/1757-899x/1155/1/012100

29. Saravanos A., Curinga M. X. Simulating the software development lifecycle: The waterfall model. *Applied System Innovation*. 2023. Vol. 6, No. 6. P. 108. DOI: 10.3390/asi6060108

30. Kuhrmann M. et al. What makes agile software development agile? *IEEE transactions on software engineering*. 2021. Vol. 48, №. 9. P. 3523-3539. DOI: 10.1109/TSE.2021.3099532

31. Mazrae R. P. et al. On the usage, co-usage and migration of CI/CD tools: A qualitative analysis. *Empirical Software Engineering*. 2023. Vol. 28, №. 2. P. 52. DOI: 10.1007/s10664-022-10285-5

32. Mahida A. A Review on Continuous Integration and Continuous Deployment (CI/CD) for Machine Learning. *International journal of science and research*. 2021. Vol. 10, №. 3. P. 1967-1970. DOI : https://dx.doi.org/10.21275/SR24314131827

33. Fluri J., Fornari F., Pustulka E. On the importance of CI/CD practices for database applications. *Journal of Software: Evolution and Process*. 2024. Vol. 36, No. 12. P. e2720. DOI: 10.1002/smr.2720

34. Kachanov V. V. et al. Technical debt in the software development lifecycle: code smells. *Proceedings of the Institute for System Programming of the RAS*. 2022. Vol. 33, No. 6. P. 95-110. DOI: 10.15514/ispras-2021-33(6)-7

35. Randhawa T. S. Software Life Cycle. *Mobile applications: Design, development and optimization*. Cham : Springer International Publishing. 2022. P. 1-55. DOI: https://doi.org/10.1007/978-3-030-02391-1

36. Jain N., Jain A. Software Development Life Cycle: A Detailed Study. *International journal of advanced research in computer science*. 2011. Vol. 2, No. 3. P. 261-264. DOI: 10.26483/ijarcs.v2i3.512

37. Bellini P. et al. Rapid prototyping & development life cycle for smart applications of internet of entities. *27th International Conference on Engineering of Complex Computer Systems (ICECCS). IEEE*, 2023. P. 142-151. DOI: 10.1109/iceccs59891.2023.00026

38. Prates L., Pereira R. DevSecOps practices and tools. *International Journal of Information Security*. 2025. Vol. 24, № 1. P. 11. DOI: 10.1007/s10207-024-00914-z

39. Sinan M., Shahin M., Gondal I. Integrating Security Controls in DevSecOps: Challenges, Solutions, and Future Research Directions. *Journal of Software: Evolution and Process*. 2025. Vol. 37, №. 6. P. e70029. DOI: 10.1002/smr.70029

40. Anjaria D., Kulkarni M. Effective DevSecOps Implementation: A Systematic Literature Review. *Cardiometry*. 2022. № 24. P. 410-417. DOI: 10.47059/revistageintec.v11i4.2514

41. Ramaj X. et al. Unveiling the safety aspects of DevSecOps: evolution, gaps and trends. *Recent Advances in Computer Science and Communications*. 2023. Vol. 16, № 3. P. 61-69. DOI: 10.2174/2666255816666220804143918

42. Alanda A. et al. Web application penetration testing using SQL injection attack. *International Journal on Informatics Visualization*. 2021. Vol. 5, №. 3. P. 320-326. DOI: 10.30630/joiv.5.3.470

43. Arcuri A. et al. Emb: A curated corpus of web/enterprise applications and library support for software testing research. *IEEE Conference on Software Testing, Verification and Validation.* 2023. P. 433-442. DOI: 10.1109/icst57152.2023.00047

44. Rooij O. et al. webfuzz: Grey-box fuzzing for web applications. *European Symposium on Research in Computer Security*. 2021. P. 152-172. DOI: 10.1007/978-3-030-88418-5_8

45. Толкачова А., Піскозуб А. Методи для тестування безпеки вебзастосунків. *Кібербезпека: освіта, наука, техніка*. 2024. Т. 2, № 26. С. 115-122. DOI: https://doi.org/10.28925/2663-4023.2024.26.668

46. Муляр І. та ін. Метод пошуку вразливостей вебзастосунків з використанням API ChatGPT. *Підводні технології*. 2024. Т. 2, № 15. С. 46-55. DOI: https://doi.org/10.32347/uwt.2024.15.1203

47. Скідан В. В., Пилипенко В. І., Каленський Б. В. Автоматизація тестування вебзастосунків // Мехатронні системи: інновації та інжиніринг : матеріали VII Міжнар. наук.-практ. конф. 2023. С. 228-229. URL: https://er.knutd.edu.ua/bitstream/123456789/26070/1/MSIE_2023_P228-229.pdf

48. Feio C. et al. An empirical study of devsecops focused on continuous security testing // IEEE European Symposium on Security and Privacy Workshops. 2024. P. 610-617. DOI: eurospw61312.2024.00074

49. Baig A. F., Eskeland S. Security, privacy, and usability in continuous authentication: A survey. *Sensors*. 2021. Vol. 21, № 17. P. 5967. DOI: 10.3390/s21175967

50. Rindell K. et al. Security in agile software development: A practitioner survey. *Information and Software Technology*. 2021. Vol. 131. P. 106488. DOI: 10.1016/j.infsof.2020.106488

51. Croft R. et al. An empirical study of rule-based and learning-based approaches for static application security testing // Proceedings of the 15th ACM/IEEE international symposium on empirical software engineering and measurement (ESEM). 2021. P. 1-12. DOI: 10.1145/3475716.3475781

52. Li K. et al. Comparison and evaluation on static application security testing (SAST) tools for java // Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering. 2023. P. 921-933. DOI: 10.1145/3611643.3616262

53. Albahar M., Alansari D., Jurcut A. An empirical comparison of pen-testing tools for detecting web app vulnerabilities. *Electronics*. 2022. Vol. 11, № 19. P. 2991. DOI: 10.3390/electronics11192991

54. Software Testing Help. Differences between SAST, DAST, IAST, and RASP. URL: https://www.softwaretestinghelp.com/differences-between-sast-dast-iast-and-rasp/.

55. Seth A. et al. Comparing effectiveness and efficiency of interactive application security testing (IAST) and runtime application self-protection (RASP) tools in a large java-based system. *Empirical Software Engineering*. 2025. Vol. 30, № 3. P. 67. DOI: 10.2139/ssrn.4306114

56. Imtiaz N., Thorn S., Williams L. A comparative study of vulnerability reporting by software composition analysis tools // Proceedings of the 15th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM). 2021. P. 1-11. DOI: 10.1145/3475716.3475769

57. Zhao L. et al. Software composition analysis for vulnerability detection: An empirical study on Java projects // Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering. 2023. P. 960-972. DOI: 10.1145/3611643.3616299

58. Jiang L. et al. Binaryai: Binary software composition analysis via intelligent binary source code matching // Proceedings of the IEEE/ACM 46th International Conference on Software Engineering. 2024. P. 1-13. DOI: 10.1145/3597503.3639100

59. Jeong B. et al. Utopia: Automatic generation of fuzz driver using unit tests // IEEE Symposium on Security and Privacy (SP). 2023. P. 2676-2692. DOI: 10.1109/sp46215.2023.10179394

60. Xiong W. et al. Cyber security threat modeling based on the MITRE Enterprise ATT&CK Matrix. *Software and Systems Modeling*. 2022. Vol. 21, № 1. P. 157-177. DOI:10.1007/s10270-021-00898-7

61. Crothers E. N., Japkowicz N., Viktor H. L. Machine-generated text: A comprehensive survey of threat models and detection methods. *IEEE Access*. 2023. Vol. 11. P. 70977-71002. DOI: 10.1007/s10270-021-00898-7

62. Loncar K., Redzepagic J., Dakic V. Secure coding guidelines and standards. *Annals of DAAAM & Proceedings*. 2024. Vol. 35. P. 0191-0198. DOI: 10.2507/35th.daaam.proceedings.025

63. Singh R. et al. Analysis of web application vulnerabilities using dynamic application security testing // *IEEE 9th International Conference for Convergence in Technology (I2CT).* 2024. P. 1-6. DOI: 10.1109/i2ct61223.2024.10543484

64. Gupta N., Jindal V., Bedi P. A survey on intrusion detection and prevention systems. *SN Computer Science*. 2023. Vol. 4. P. 439. DOI: 10.1007/s42979-023-01926-7

65. Kumar A. et al. Intrusion detection and prevention system for an IoT environment. *Digital Communications and Networks*. 2022. Vol. 8, Is. 4. P. 540-551. DOI: 10.1016/j.dcan.2022.05.027

66. Kizza J. M. System intrusion detection and prevention // Kizza J. M. (eds) A Guide to computer network security. The Computer Communications and Networks., London : Springer, 2009. P. 273-298. DOI: 10.1007/978-1-84800-917-2_13

67. Shaheed A., Kurdy M. H. D. B. Web application firewall using machine learning and features engineering. *Security and Communication Networks*. 2022. Is. 1. P. 5280158. DOI: 10.1155/2022/5280158

68. Seng L., Ithnin N., Said S. The approaches to quantify web application security scanners quality: a review. *International Journal of Advanced Computer Research.* 2018. Vol. 8, Iss. 38. P. 285-312. DOI: http://dx.doi.org/10.19101/IJACR.2018.838012

69. Shahid J., Hameed M., Javed I., Qureshi K., Ali M., Crespi N. A Comparative Study of Web Application Security Parameters: Current Trends and Future Directions. *Applied Sciences*. 2022. Vol. 12. P. 4077. DOI: https://doi.org/10.3390/app12084077

70. Mothanna Y., ElMedany W., Hammad M., Ksantini R., Sharif M. Adopting security practices in software development process: Security testing

framework for sustainable smart cities. *Computers & Security*. 2024. Vol. 144. № 103985. P. 1-9. DOI: https://doi.org/10.1016/j.cose.2024.103985

71. Dukes L., Yuan X., Akowuah F. A case study on web application security testing with tools and manual testing. *Proceedings of IEEE Southeastcon.* 2013. P. 1-6. DOI: https://doi.org/10.1109/SECON.2013.6567420

72. Aydos M., Aldan Ç., Coşkun E., Soydan A. Security testing of web applications: A systematic mapping of the literature. *Journal of King Saud University - Computer and Information Sciences*. 2022. Vol. 34, Iss. 9. P. 6775-6792. DOI: https://doi.org/10.1016/j.jksuci.2021.09.018

73. The complete guide to developer-first application security. *GitHub : website.* URL: https://github.com/resources/whitepapers/developer-first-application-security

74. Interactive Application Security Testing. ContrastSecurity : website. URL: https://www.contrastsecurity.com/glossary/interactive-application-security-testing

75. What is RASP: Runtime Application Self Protection. URL: https://www.softwaretestinghelp.com/rasp-tutorial/

76. Інструменти тестування безпеки: SAST / DAST / IAST / RAPS. URL: https://qagroup.com.ua/publications/instrumenty-testuvannia-bezpeky-sast-dast-iast-raps/

77. Top 28 Cloud Security Posture Management (CSPM) Tools. URL: https://startupstash.com/cloud-security-posture-management-tools/

78. Tauqeer O. B., Jan S., Khadidos A. O., Khan F. Q. et al. Analysis of security testing techniques. *Intelligent Automation & Soft Computing*. 2021. *Vol. 29(1)*. P. 291-306. DOI: https://doi.org/10.32604/iasc.2021.017260

79. Fernandes A. A. Evaluating the top application security tools: from static analysis to runtime protection. *Asian Journal of Research in Computer Science*. 2024. Vol. 17(7). P. 119-27. DOI: https://doi.org/10.9734/ajrcos/2024/v17i7483.

80. 12 Best Code Analysis Tools in 2024. URL: https://thectoclub.com/tools/best-code-analysis-tools/

81. The Top 10 Dynamic Application Security Testing (DAST) Tools. ExpertInsight : website. URL: https://expertinsights.com/insights/the-top-dynamic-application-security-testing-dast-tools/

82. Tudela M.F., Higuera J.B., Higuera J.B., Montalvo J.S., Argyros M.I. On Combining Static, Dynamic and Interactive Analysis Security Testing Tools to Improve OWASP Top Ten Security Vulnerability Detection in Web Applications. *Applied Sciences*. 2020. Vol. 10, № 24: 9119. P. 1-24. DOI: https://doi.org/10.3390/app10249119

83. The Top 7 Interactive Application Security Testing (IAST) Tools. URL: https://expertinsights.com/insights/the-top-interactive-application-security-testing-iast-tools/

84. The Top 8 Runtime Application Self-Protection (RASP) Software. URL: https://expertinsights.com/insights/the-top-runtime-application-self-protection-rasp-software/

85. Average duration of downtime after a ransomware attack at organizations in the United States from 1st quarter 2020 to 2nd quarter 2022. URL: https://www.statista.com/statistics/1275029/length-of-downtime-after-ransomware-attack-us/

86. A Roundup of the 23 Best DAST Tools of 2024. URL: https://thectoclub.com/tools/best-dast-tools/

87. May M., Zelalem B., Eunkyoung J., Doo-Hwan B. Cybersecurity Vulnerability Identification in System-of-Systems using Model-based Testing // 17th Annual System of Systems Engineering Conference (SOSE). 2022. P. 317-322. DOI: https://doi.org/10.1109/SOSE55472.2022.9812676

88. Marksteiner S., Priller P., Wolf M. Approaches for Automating Cybersecurity Testing of Connected Vehicles // Karner M., Peltola J., Jerne M., Kulas L., Priller P. (eds). Intelligent Secure Trustable Things. Studies in Computational Intelligence. 2024. Vol. 1147. P. 219-234. DOI: https://doi.org/10.1007/978-3-031-54049-3_13

89. Krishna A. Automated Pentesting 101 (How It Works + ROI You Can Expect). URL: https://www.getastra.com/blog/security-audit/automated-penetration-testing-software/

90. Keshav M. Automated VS Manual Security Testing – Which One to Choose? URL: https://www.getastra.com/blog/security-audit/manual-security-testing/

91. Saumick B. 17 Best Penetration Testing Tools / Software of 2023 [Reviewed]. URL: https://www.getastra.com/blog/security-audit/best-penetration-testing-tools/

92. Nguyen T. C. H. et al. Improving web application firewalls with automatic language detection. *SN Computer Science*. 2022. Vol. 3, № 6. P. 446. DOI: 10.1007/s42979-022-01327-2

93. Pingale S. V., Sutar S. R. Analysis of web application firewalls, challenges, and research opportunities // Proceedings of the 2nd International Conference on Data Science, Machine Learning and Applications. Singapore : Springer, 2021. P. 239-248. DOI: 10.1007/978-981-16-3690-5_21

94. Hashim A., Medani R., Attia T. A. Defences against web application attacks and detecting phishing links using machine learning // International conference on computer, control, electrical, and electronics engineering (ICCCEEE). IEEE, 2021. P. 1-6. DOI: 10.1109/iccceee49695.2021.9429609

95. Nirmal K., Janet B., Kumar R. Analyzing and eliminating phishing threats in IoT, network and other Web applications using iterative intersection. *Peer-to-Peer Networking and Applications*. 2021. Vol. 14, № 4. P. 2327-2339. DOI: 10.1007/s12083-020-00944-z

96. Sadiq A. et al. A review of phishing attacks and countermeasures for internet of things-based smart business applications in industry 4.0. *Human behavior and emerging technologies*. 2021. Vol. 3, No. 5. P. 854-864. DOI: https://doi.org/10.1002/HBE2.301

97. Taha D. M. A., Jabar H. D. A., Mohammed W. K. A machine learning algorithms for detecting phishing websites: A comparative study. *Iraqi Journal for*

Computer Science and Mathematics. 2024. Vol. 5, Iss. 3. P. 13. DOI: 10.52866/ijcsm.2024.05.03.015

98. Catal C. et al. Applications of deep learning for phishing detection: a systematic literature review. *Knowledge and Information Systems*. 2022. Vol. 64, Iss. 6. P. 1457-1500. DOI: 10.1007/s10115-022-01672-x

99. Shombot E. S. et al. An application for predicting phishing attacks: A case of implementing a support vector machine learning model. *Cyber Security and Applications*. 2024. Vol. 2. P. 100036. DOI: 10.1016/j.csa.2024.100036

100. Sushma K., Jayalakshmi M., Guha T. Deep learning for phishing website detection. *IEEE 2nd Mysore Sub Section International Conference (MysuruCon)*. 2022. P. 1-6. DOI: 10.1109/mysurucon55714.2022.9972621

101. Jha A. K., Muthalagu R., Pawar P. M. Intelligent phishing website detection using machine learning. *Multimedia Tools and Applications*. 2023. Vol. 82, Iss. 19. P. 29431-29456. DOI: 10.1007/s11042-023-14731-4

102. Zaimi R., Hafidi M., Lamia M. A deep learning mechanism to detect phishing URLs using the permutation importance method and SMOTE-Tomek link. *The Journal of Supercomputing*. 2024. Vol. 80, Iss. 12. P. 17159-17191. DOI: 10.1007/s11227-024-06124-7

103. Jain A. K., Gupta B. B. A survey of phishing attack techniques, defence mechanisms and open research challenges. *Enterprise Information Systems*. 2022. Vol. 16, No. 4. P. 527-565. DOI: 10.1080/17517575.2021.1896786

104. Alghenaim M. F. et al. Phishing attack types and mitigation: A survey. *The International Conference on Data Science and Emerging Technologies*. Singapore: Springer, 2023. P. 131-153. DOI: https://doi.org/10.1007/978-981-99-0741-0_10

105. Asiri S. et al. A survey of intelligent detection designs of HTML URL phishing attacks. *IEEE Access*. 2023. Vol. 11. P. 6421-6443. DOI: 10.1109/access.2023.3237798

106. Ali M. M., Mohd Zaharon N. F. Phishing – A cyber fraud: The types, implications and governance. *International Journal of Educational Reform*. 2024. Vol. 33, Iss. 1. P. 101-121. DOI: 10.1177/10567879221082966

107. Kavya S., Sumathi D. Staying ahead of phishers: a review of recent advances and emerging methodologies in phishing detection. *Artificial Intelligence Review*. 2024. Vol. 58, № 2. P. 50. DOI: 10.1007/s10462-024-11055-z

108. Justindhas Y. et al. A comprehensive review on an ensemble-based machine learning approach for phishing website detection // 2nd International Conference on Computer, Communication and Control (IC4). IEEE, 2024. P. 1-6. DOI: 10.1109/ic457434.2024.10486561

109. Yarphel T., Rani D. Cross-Site Scripting (XSS) in Web Applications: A systematic literature review. *International Journal of Science and Research Archive*. 2025. Vol. 15, № 2. P. 1658-1667. DOI: 10.30574/ijsra.2025.15.2.1521

110. Majeed M. A. et al. A review on text steganography techniques. *Mathematics*. 2021. Vol. 9, № 21. P. 2829. DOI: 10.3390/math9212829

111. Gurunath R., Samanta D. A novel approach for semantic web application in online education based on steganography. *International Journal of Web-Based Learning and Teaching Technologies (IJWLTT)*. 2022. Vol. 17, № 4. P. 1-13. DOI: 10.4018/ijwltt.285569

112. Azam M. H. et al. A systematic review on cover selection methods for steganography: Trend analysis, novel classification and analysis of the elements. *Computer Science Review*. 2025. Vol. 56. P. 100726. DOI: 10.1016/j.cosrev.2025.100726

113. Subramanian N. et al. Image steganography: A review of the recent advances. *IEEE Access*. 2021. Vol. 9. P. 23409-23423. DOI: 10.1109/access.2021.3053998

114. Setiadi D. R. I. M. et al. Digital image steganography survey and investigation (goal, assessment, method, development, and dataset). *Signal processing*. 2023. Vol. 206. P. 108908. DOI: 10.1016/j.sigpro.2022.108908

115. Xu Y. et al. Robust invertible image steganography // Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022. P. 7875-7884. DOI: 10.1109/cvpr52688.2022.00772

116. Кальчук І. В., Лаптєва Т. О., Лукова-Чуйко Н. В., Харкевич Ю. І. Метод побудови захищених каналів передачі даних з використанням модифікованої нейронної мережі. *Information Technology and Security*. 2021. Vol. 9(2). P. 232-243. DOI: https://doi.org/10.20535/2411-1031.2021.9.2.250077

117. Kobozeva A., Lebedeva E., Kostyrka O. Stego transformation of spatial domain of cover image robust against attacks on embedded message. *Problemele energeticii regionale. Sistemele informaţionale şi robotica*. 2014. №1(24). P. 71-81. URL: https://journal.ie.asm.md/assets/files/07_01_24_2014.pdf

118. Рудницький В. М., Костирка О. В. Стійке стеганоперетворення в просторовій області зображення-контейнера. *Інформатика та математичні методи в моделюванні*. 2013. Т. 3, № 4. С. 353-360. URL : http://nbuv.gov.ua/UJRN/Itmm_2013_3_4_9

119. Костирка О. В. Модифікація стійкого до збурних дій стеганоперетворення просторової області зображення-контейнера. *Інформатика та математичні методи в моделюванні*. 2016. Т. 6, № 1. С. 85-93. URL: https://repositsc.nuczu.edu.ua/bitstream/123456789/21579/1/110402.pdf

120. Varghese F., Sasikala P. A detailed review based on secure data transmission using cryptography and steganography. *Wireless Personal Communications*. 2023. Vol. 129, № 4. P. 2291-2318. DOI: 10.1007/s11277-023-10183-z

121. Yao Y. et al. High invisibility image steganography with wavelet transform and generative adversarial network. *Expert Systems with Applications*. 2024. Vol. 249, Part A, P. 123540. DOI: 10.1016/j.eswa.2024.123540

122. Apau R. et al. Image steganography techniques for resisting statistical steganalysis attacks: A systematic literature review. *PloS one*. 2024. Vol. 19, Iss. 9. P. e0308807. DOI: 10.1371/journal.pone.0308807

123. Shehab D. A., Alhaddad M. J. Comprehensive survey of multimedia steganalysis: Techniques, evaluations, and trends in future research. *Symmetry*. 2022. Vol. 14, Iss. 1. P. 117. DOI: 10.3390/sym14010117

124. Шелест М. Є. та ін. Стеганографічний метод, стійкий до збурних дій. *Інформатика та математичні методи в моделюванні*. 2019. Т. 9, №. 4. С. 324-332.

125. Dalal M., Juneja M. Steganalysis of DWT Based Steganography Technique for SD and HD Videos. *Wireless Personal Communications*. 2023. Vol. 128, № 4. P. 2441-2452. DOI: 10.1007/s11277-022-10050-3

126. Sun Y. Enhancing image steganalysis via integrated reinforcement learning and dilated convolution techniques. *Signal, Image and Video Processing*. 2024. Vol. 18., № Suppl 1. P. 1-16. DOI: 10.1007/s11760-024-03113-4

127. Zhou H. et al. Three-dimensional mesh steganography and steganalysis: A review. *IEEE Transactions on Visualization and Computer Graphics*. 2021. Vol. 28, No. 12. P. 5006-5025. DOI: 10.1109/tvcg.2021.3075136

128. De La Croix N. J. et al. Steganalysis in Spatial Domain Images for Securing Data Transmission in IoT Environments. *IEEE 21st International Conference on Mobile Ad-Hoc and Smart Systems (MASS)*. 2024. P. 586-591. DOI: 10.1109/mass62177.2024.00093

129. Bravo-Ortiz M. A. et al. CVTStego-Net: A convolutional vision transformer architecture for spatial image steganalysis. *Journal of Information Security and Applications*. 2024. Vol. 81. P. 103695. DOI: 10.1016/j.jisa.2023.103695

130. Duan X. et al. Preprocessing enhancement method for spatial domain steganalysis. *Mathematics*. 2022. Vol. 10, Iss. 21. P. 3936. DOI: 10.3390/math10213936

131. Putra Y. H. et al. Sensitivity of a Convolutional Neural Network for Different Pooling Layers in Spatial Domain Steganalysis. *Ingenierie des Systemes d'Information*. 2024. Vol. 29, No. 4. P. 1653-1665. DOI: 10.18280/isi.290438

132. Ahmed N., Natarajan T., Rao K. R. Discrete cosine transform. *IEEE transactions on Computers*. 2006. Vol. 100, Iss. 1. P. 90-93. DOI: 10.1109/t-c.1974.223784

133. Lu Leng, Jiashu Zhang, Jing Xu et al. Dynamic weighted discrimination power analysis: A novel approach for face and palmprint recognition in DCT domain. *International Journal of Physical Sciences*. 2010. Iss. 5(17) P. 467-471.

134. Jia-Fa M. et al. A steganalysis method in the DCT domain. *Multimedia Tools and Applications*. 2016. Vol. 75, No. 10. P. 5999-6019. DOI: 10.1007/s11042-015-2708-0

135. Holub V., Fridrich J. Low-complexity features for JPEG steganalysis using undecimated DCT. *IEEE Transactions on Information forensics and security*. 2014. Vol. 10, No. 2. P. 219-228. DOI: 10.1109/tifs.2014.2364918

136. Wang Y., Moulin P. Steganalysis of block-DCT image steganography. *IEEE Workshop on Statistical Signal Processing*, 2003. 2003. P. 339-342. DOI: 10.1109/ssp.2003.1289414

137. Abdi H. Singular value decomposition (SVD) and generalized singular value decomposition. *Encyclopedia of measurement and statistics*. 2007. Vol. 907, Iss. 912. P. 44.

138. Akritas A. G., Malaschonok G. I. Applications of singular-value decomposition (SVD). *Mathematics and computers in simulation*. 2004. Vol. 67, Iss. 1-2. P. 15-31.

139. Klema V., Laub A. The singular value decomposition: Its computation and some applications. *IEEE Transactions on automatic control*. 1980. Vol. 25, Iss. 2. P. 164-176. DOI: 10.1109/tac.1980.1102314

140. Ahmed N., Rao K. R. Walsh-hadamard transform. Orthogonal transforms for digital signal processing. Berlin, Heidelberg : Springer Berlin Heidelberg, 1975. P. 99-152. DOI: 10.1007/978-3-642-45450-9_6

141. Jayathilake A., Perera A. A. I., Chamikara M. A. P. Discrete Walsh-Hadamard transform in signal processing. *IJRIT Int. J. Res. Inf. Technol*. 2013. Vol. 1. P. 80-89.

142. Pan H., Badawi D., Cetin A. E. Fast Walsh-Hadamard transform and smooth-thresholding based binary layers in deep neural networks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2021. P. 4650-4659. DOI: 10.1109/cvprw53098.2021.00523

143. Lanovska O. O., Sokolov A. V. Steganalysis of a method with code control of information embedding in the Walsh-Hadamard transform domain. *Informatics & Mathematical Methods in Simulation*. 2024. Vol. 14, Iss. 4. P. 273-283. DOI: 10.15276/imms.v14.no4.273

144. Zhang D. Wavelet transform. *Fundamentals of image data mining: Analysis, Features, Classification and Retrieval*. Cham: Springer International Publishing, 2019. P. 35-44

145. Bentley P. M., McDonnell J. T. E. Wavelet transforms: an introduction. *Electronics & Communication Engineering Journal*. 1994. Vol. 6, Iss. 4. P. 175-186. DOI: 10.1049/ecej:19940401

146. Burrus C. S., Gopinath R. A., Guo H. Wavelets and wavelet transforms. rice university, houston edition. 1998. Vol. 98. 311 p.

147. Lilly J. M., Olhede S. C. Higher-order properties of analytic wavelets. *IEEE Transactions on Signal Processing*. 2008. Vol. 57, No. 1. P. 146-160. DOI: 10.1109/tsp.2008.2007607

148. Shi Y. Q. et al. Effective steganalysis based on statistical moments of wavelet characteristic function // International Conference on Information Technology: Coding and Computing (ITCC'05). 2005. Vol. 1. P. 768-773. DOI: 10.1109/ITCC.2005.138

149. Luo X. et al. Image universal steganalysis based on wavelet packet transform // IEEE 10th Workshop on Multimedia Signal Processing. 2008. P. 780-784. DOI: 10.1109/mmsp.2008.4665180

150. Liu S., Yao H., Gao W. Steganalysis of data hiding techniques in wavelet domain // International Conference on Information Technology: Coding and Computing. 2004. Vol. 1. P. 751-754. DOI: 10.1109/itcc.2004.1286558

151. Winograd S. On computing the discrete Fourier transform. *Mathematics of computation*. 1978. Vol. 32, Iss. 141. P. 175-199. DOI: 10.2307/2006266

152. Selesnick I. W. et al. The discrete Fourier transform. *The transform and data compression handbook.* 2001. P. 37-74. DOI: 10.1201/9781420037388.ch2

153. Lanari R., Hensley S., Rosen P. A. Chirp z-transform based SPECAN approach for phase-preserving ScanSAR image generation. *IEEE Proceedings-Radar, Sonar and Navigation*. 1998. Vol. 145 No. 5. P. 254-261. DOI: 10.1049/ip-rsn:19982218

154. Rabiner L. R., Schafer R. W., Rader C. M. The chirp z-transform algorithm and its application. *Bell System Technical Journal*. 1969. Vol. 48, Iss. 5. P. 1249-1292. DOI: 10.1002/j.1538-7305.1969.tb04268.x

155. Horai M., Kobayashi H., Nitta T. G. Chirp signal transform and its properties. *Journal of Applied Mathematics*. 2014. Vol. 2014, Iss. 1. P. 161989. DOI: 10.1155/2014/161989

156. Xia X. G. Discrete chirp-Fourier transform and its application to chirp rate estimation. *IEEE Transactions on Signal processing*. 2000. Vol. 48, No. 11. P. 3122-3133. DOI: 10.1109/78.875469

157. Muralidharan T. et al. The infinite race between steganography and steganalysis in images. *Signal Processing*. 2022. Vol. 201. P. 108711. DOI: 10.1016/j.sigpro.2022.108711

158. Kranthi A. G. et al. Securing web apps: Analysis to understand common vulnerabilities, attack scenarios, and protective measures. *ICCDE* 2024. P. 64. DOI: https://doi.org/10.1145/3641181.3641187

159. Mohammed A. et al. Security of web applications: Threats, vulnerabilities, and protection methods. *International Journal of Computer Science & Network Security*. 2021. Vol. 21, № 8. P. 167-176. DOI: https://doi.org/10.22937/IJCSNS.2021.21.8.22

160. Othman N. A. et al. Image Steganography Using Web Application. *Journal of Computing Research and Innovation.* 2023. Vol. 8, № 2. P. 1-11. DOI: https://doi.org/10.24191/jcrinn.v8i2.373

161. Evsutin O., Melman A., Meshcheryakov R. Digital Steganography and Watermarking for Digital Images: A Review of Current Research Directions. IE*EE Access*. 2020. No. 8. P. 166589-166611. DOI:10.1109/ACCESS.2020.3022779

162. Kobozeva A.A., Sokolov A.V. Robust Steganographic Method with Code-Controlled Information Embedding. *Problemele energeticii regionale*. 2021. No. 4 (52). P. 115-130. DOI: https://doi.org/10.52254/1857-0070.2021.4-52.11

163. Kobozeva A.A., Sokolov A.V. Steganographic Method with Code Control of Information Embedding Based on Multi-level Code Words. *Radioelectronics and Communications Systems*. 2023. Vol. 66, № 4. P. 173-189. DOI: https://doi.org/10.3103/S0735272723040052

164. Beer T. Walsh transforms. *American Journal of Physics*. 1981. Vol. 49, No. 5. P. 466-472.

165. Kobozeva A. A., Sokolov A. V. The Sufficient Condition for Ensuring the Reliability of Perception of the Steganographic Message in the Walsh-Hadamard Transform Domain. *Problemele Energeticii Regionale.* 2022. Vol. 54 (2). P. 84-100. DOI: https://doi.org/10.52254/1857-0070.2022.2-54.08

166. Natural Resources Conservation Service (NRCS). United States Department of Agriculture. URL: https://www.nrcs.usda.gov

167. Heidari M., Gaemmaghami S. Universal image steganalysis using singular values of DCT coefficients. *10th International ISC Conference on Information Security and Cryptology (ISCISC).* IEEE, 2013. P. 1-5. DOI: 10.1109/iscisc.2013.6767340

168. Khoroshko V., Kobozeva A., Bobok I. Improving the method of detecting block processing of digital images. *Information control systems and technologies*. *Problems and solutions*. 2019. Vol. 4. P. 47-59. DOI: https://doi.org/10.30837/rt.2019.4.199.16

169. Khalilollahi S. M. S., Mansouri A. JPEG steganalysis using the relations between DCT coefficients // International Conference on Machine Vision and Image Processing (MVIP). IEEE. 2022. P. 1-4. DOI: 10.1109/mvip53647.2022.9738785

170. Kuznetsov A. et al. Image steganalysis using deep learning models. *Multimedia Tools and Applications*. 2024. Vol. 83, № 16. P. 48607-48630. DOI: 10.1007/s11042-023-17591-0

171. Jung K. H. A study on machine learning for steganalysis // Proceedings of the 3rd International Conference on Machine Learning and Soft Computing. 2019. P. 12-15. DOI: 10.1145/3310986.3311000

172. Iskanderani A. I. et al. Artificial Intelligence-Based Digital Image Steganalysis. *Security and Communication Networks.* 2021. Vol. 2021, № 1. P. 9923389. DOI: 10.1155/2021/9923389

173. Mandal P. C., Mukherjee I., Paul G., Chatterji B. N. Digital image steganography: A literature survey. *Inf. Sci.* 2022. Vol. 609. P. 1451-1488. DOI: 10.1016/j.ins.2022.07.120

174. Gupta A. et al. Machine learning and deep learning in steganography and steganalysis. *Multidisciplinary Approach to Modern Digital Steganography.* IGI Global Scientific Publishing. 2021. P. 75-98. DOI:10.4018/978-1-7998-7160-6.ch004

175. Kobozeva A. A., Melnyk M. A. Steganographic algorithm based on sign-insensitivity of singular vectors of the image matrix. *Systems of information processing*. 2013.Vol. 3, No. 110. P. 90-94.

176. Sokolov A. Artificial intelligence from a technical perspective // Digitalization, metaverse, artificial intelligence in the context of human and individual rights protection in Ukraine and the world : monograph ed. by Kostenko O., Kharytonova O., Kharytonov Y., SciFormat, 2025. P. 90-121.

177. Azizan N. et al. File hiding web application (fhwa) using image steganography. *European Proceedings of Multidisciplinary Sciences*. 2022. P. 599-609. DOI: 10.15405/epms.2022.10.56

178. Ediriweera S., Dilhara B. A. S., Disanayake C. Web-Based Data Hiding: A Hybrid Approach Using Steganography and Visual Cryptography // *International Research Conference on Smart Computing and Systems Engineering (SCSE)*. IEEE, 2023. Vol. 6. P. 1-7. DOI: 10.1109/scse59836.2023.10214994

179. Kheddar H. et al. Deep learning for steganalysis of diverse data types: A review of methods, taxonomy, challenges and future directions. *Neurocomputing*. 2024. Vol. 581. P. 127528. DOI: 10.1016/j.neucom.2024.127528

180. De La Croix N. J., Ahmad T., Han F. Comprehensive survey on image steganalysis using deep learning. *Array*. 2024. Vol. 22. P. 100353. DOI: 10.1016/j.array.2024.100353

181. Eid W. M. et al. Digital image steganalysis: current methodologies and future challenges. *IEEE Access*. 2022. Vol. 10. P. 92321-92336. DOI: 10.1109/access.2022.3202905

182. Farooq N., Selwal A. Image steganalysis using deep learning: a systematic review and open research challenges. *Journal of Ambient Intelligence and Humanized Computing*. 2023. Vol. 14, Iss. 6. P. 7761-7793. DOI: 10.1007/s12652-023-04591-z

183. Weng S. et al. Lightweight and effective deep image steganalysis network. *IEEE Signal Processing Letters*. 2022. Vol. 29. P. 1888-1892. DOI: 10.1109/lsp.2022.3201727

184. Hammad B. T., Ahmed I. T., Jamil N. A steganalysis classification algorithm based on distinctive texture features. *Symmetry*. 2022. Vol. 14, Iss. 2. P. 236. DOI: 10.3390/sym14020236

185. Ray B. et al. Image steganography using deep learning based edge detection. *Multimedia Tools and Applications*. 2021. Vol. 80. Iss. 24. P. 33475-33503. DOI:10.1007/s11042-021-11177-4

186. Ker A. D. Quantitative evaluation of pairs and RS steganalysis. *Security, Steganography, and Watermarking of Multimedia Contents VI*. SPIE, 2004. DOI: 10.1117/12.526720

187. Westfeld A. Pfitzmann A. Attacks on Steganographic Systems. *International Workshop on Information Hiding*. 1999. Vol. 1768. P. 61-76. DOI: 10.1007/10719724_5

188. Pan I. H., Liu K. C., Liu C. L. Chi-square detection for PVD steganography // International Symposium on Computer, Consumer and Control. IEEE, 2020. P. 30-33. DOI: 10.1109/is3c50286.2020.00015

189. StegExpose. URL: https://github.com/b3dk7/StegExpose

190. Abdullah D. M., Abdulazeez A. M. Machine learning applications based on SVM classification a review. *Qubahan Academic Journal*. 2021. Vol. 1, № 2. P. 81-90. DOI: 10.48161/qaj.v1n2a50

191. Chandra M. A., Bedi S. S. Survey on SVM and their application in image classification. *International Journal of Information Technology*. 2021. Vol. 13. P. 1-11. DOI: 10.1007/s41870-017-0080-1

192. Goel A., Srivastava S. K. Role of kernel parameters in performance evaluation of SVM // *Second international conference on computational intelligence & communication technology (CICT)*. IEEE, 2016. P. 166-169. DOI: 10.1109/cict.2016.40

193. Jakkula V. Tutorial on support vector machine (svm). *School of EECS, Washington State University*. 2006. Vol. 37, No. 2.5. P. 3.

194. Sánchez A V. D. Advanced support vector machines and kernel methods. *Neurocomputing*. 2003. Vol. 55, Iss. 1-2. P. 5-20. DOI: 10.1016/s0925-2312(03)00373-4

195. Murty M. N., Raghava R. Kernel-based SVM. Support vector machines and perceptrons. *Learning, optimization, classification, and application to social networks*. Cham : Springer International Publishing. 2016. P. 57-67

196. Phishing. Statistics & Facts. URL: https://www.statista.com/topics/8385/phishing/

197. Top 5 Phishing Statistics 2025. URL: https://www.broadband4europe.com/phishing-statistics/

198. Phishing Statistics by Demographic, Healthcare, Industry and Country. URL: https://www.sci-tech-today.com/stats/phishing-statistics-updated/

199. Habrylchuk A. V., Susukailo V. A., Kurii Y. O., Vasylyshyn S. I. Analysis of Cyber Attacks Using Machine Learning on the Information Security Management Systems. *Computer systems and networks.* 2025. Vol. 7, № 1. P. 68-78. DOI: https://doi.org/10.23939/csn2025.01.068

200. Денисюк Д., Сорочинський О., Гнатчук Є., & Дрозд А. Інформаційна технологія виявлення зловмисних кодів в інформаційних системах на основі аналізу паралельних процесів. *Вісник Хмельницького національного університету. Серія: Технічні науки.* 2025. Vol 347. № 1. P. 576-583. DOI: https://doi.org/10.31891/2307-5732-2025-347-80

201. Журавчак А., Піскозуб А. Аналіз методів машинного навчання для автоматизації тестування на проникнення. *Кібербезпека: освіта, наука, техніка*. 2025. Vol. 3, № 27. P. 54–62. DOI: https://doi.org/10.28925/2663-4023.2025.27.711

202. Bhinge Pr. Threat Guard AI: A Multi-Domain System for Intelligent Fraud Detection Using Machine Learning. *International Journal for Research in Applied Science and Engineering Technology*. 2025. Vol. 13. Iss. V. P. 2985-2998. DOI: https://doi.org/10.22214/ijraset.2025.70866

203. Albahadili A., Akbas A., Rahebi J. Detection of phishing URLs with deep learning based on GAN-CNN-LSTM network and swarm intelligence algorithms. *Signal, Image and Video Processing.* 2024. Vol. 18. P. 4979-4995. DOI: https://doi.org/10.1007/s11760-024-03204-2

204. Burova N., Oprysk R., Kurii Y., Lakh Y., Susukailo V. (2024). Machine learning as a key tool in defensive cyber operations: the effectiveness of phishing threat detection. *Social Development and Security*. 2024. Vol. 14, № 5. P. 113-123. DOI: https://doi.org/10.33445/sds.2024.14.5.11

205. Adane K., Beyene B., Yimer M. Intelligent Phishing Website Detection before and after Multiple Informative Feature Selection Techniques: Machine

Learning Approach. *International Journal of Information Science and Management*. 2024. Issue 22. P. 31-62. DOI: https://doi.org/10.22034/ijism.2023.1977974.0.

206. Prokopov V., Meleshko Ye., Yakymenko M., Reznichenko V., Shymko S. The methods of data storing of a recommendation system based on linked lists. *Control, Navigation and Communication Systems. Academic Journal.* 2022. Vol. 2, № 68. P. 79-84. DOI: https://doi.org/10.26906/SUNZ.2022.2.079

207. Phishing Statistics. TRUELIST : website. 2024. URL: https://truelist.co/blog/phishing-statistics/

208. Most impactful stats from the 2024 Email Security Risk Report. URL: https://www.egress.com/blog/company-news/stats-from-the-email-security-risk-report

209. Williams W. AI is making phishing emails far more convincing with fewer typos and better formatting: Here's how to stay safe. *TechRadar : website*. URL: https://www.techradar.com/pro/security/ai-is-making-phishing-emails-far-more-convincing-with-fewer-typos-and-better-formatting-heres-how-to-stay-safe

210. Business Email Compromise Statistics 2025. *Hoxhunt : website*. URL: https://hoxhunt.com/blog/business-email-compromise-statistics

211. Top 15 Phishing Stats to Know in 2024. URL: https://news.trendmicro.com/2024/07/22/phishing-stats-2024

212. Mascellino A. Phishing Attacks Double in 2024. *Infosecurity Magazine : website*. URL: https://www.infosecurity-magazine.com/news/2024-phishing-attacks-double/

213. Arad R. 6 Ways to Prevent Man-in-the-Middle (MitM) Attacks. *MEMCYCO : website*. 20.11.2024. URL: https://www.memcyco.com/6-ways-to-prevent-man-in-the-middle-mitm-attacks/

214. The Latest 2025 Phishing Statistics (updated June 2025). *AAG : website.* URL: https://aag-it.com/the-latest-phishing-statistics/

215. Phishing Trends Report (Updated for 2025). URL: https://hoxhunt.com/guide/phishing-trends-report

216. Exploring Q4 2024 Brand Phishing Trends: Microsoft Remains the Top Target as LinkedIn Makes a Comeback. URL: https://blog.checkpoint.com/research/exploring-q4-2024-brand-phishing-trends-microsoft-remains-the-top-target-as-linkedin-makes-a-comeback/

217. OWASP Top Ten. URL: https://owasp.org/www-project-top-ten/

218. Product Security and Certification. *Enisa*. URL: https://www.enisa.europa.eu/topics/product-security-and-certification

219. Karim A. et al. Phishing detection system through hybrid machine learning based on URL. *IEEE Access*. 2023. Vol. 11. P. 36805-36822. DOI: 10.1109/access.2023.3252366

220. Alshingiti Z. et al. A deep learning-based phishing detection system using CNN, LSTM, and LSTM-CNN. *Electronics*. 2023. Vol. 12, № 1. P. 232. DOI: 10.3390/electronics12010232

221. Al-Ahmadi S., Alotaibi A., Alsaleh O. PDGAN: Phishing detection with generative adversarial networks. *IEEE Access*. 2022. Vol. 10. P. 42459-42468. DOI: 10.1109/access.2022.3168235

222. Alsubaei F. S., Almazroi A. A., Ayub N. Enhancing phishing detection: A novel hybrid deep learning framework for cybercrime forensics. *IEEE Access*. 2024. Vol. 12. P. 8373-8389. DOI: 10.1109/access.2024.3351946

223. Tan C. C. L. et al. Hybrid phishing detection using joint visual and textual identity. *Expert systems with applications*. 2023. Vol. 220. P. 119723. DOI: 10.1016/j.eswa.2023.119723

224. Chataut R., Gyawali P. K., Usman Y. Can AI keep you safe? A study of large language models for phishing detection // IEEE 14th Annual Computing and Communication Workshop and Conference (CCWC). IEEE, 2024. P. 0548-0554. DOI: 10.1109/ccwc60891.2024.10427626

225. Linh D. M. et al. Real-time phishing detection using deep learning methods by extensions. *International Journal of Electrical and Computer Engineering (IJECE)*. 2024. Vol. 14, № 3. P. 3021-3035. DOI: 10.11591/ijece.v14i3.pp3021-3035

226. Baptista I., Shiaeles S., Kolokotronis N. A Novel Malware Detection System Based on Machine Learning and Binary Visualization // IEEE International Conference on Communications Workshops (ICC Workshops). 20-24 May 2019, Shanghai, China. P. 1-6. DOI: 10.1109/ICCW.2019.8757060

227. VirusShare.com, Because Sharing is Caring. URL: https://virusshare.com/about

228. Abdul K., Mobeen Sh., Khabib M., et al. Phishing Detection System Through Hybrid Machine Learning Based on URL. *IEEE Access*. 2023. Vol. 11. P. 36805-36822. DOI: 10.1109/access.2023.3252366

229. Shirazia H., Haynesb K., Raya I. Towards performance of NLP transformers on URL-based phishing detection for mobile devices. *Journal of Ubiquitous Systems & Pervasive Networks*. 2022. Vol. 17, № 1. P. 35-42. DOI: 10.5383/JUSPN.17.01.005

230. Sahingoz O. K, Buber E., Demir O., Diri B. Machine learning based phishing detection from URLs. *Expert Syst. Appl*. 2019. Vol. 117. P. 345-357. DOI: 10.1016/j.eswa.2018.09.029

231. Shahrivari V., Darabi M., Izadi M. Phishing detection using machine learning techniques. *arXiv* : *website.* 2020. Vol. 11116. P. 1-9. DOI: 10.48550/arXiv.2009.11116

232. Security Development Lifecycle. PROTELION: website. URL: https://protelion.com/support/sdl/

233. Basireddy R. M. Investigations into Security Testing Techniques, Tools, and Methodologies for Identifying and Mitigating Security Vulnerabilities. *Journal of Artificial Intelligence, Machine Learning and Data Science (JAIMLD).* 2024. Vol. 2(2). P. 1-5. DOI: https://doi.org/10.51219/JAIMLD/maheswara-reddy-basireddy/161

234. OWASP Security Culture. URL: https://owasp.org/www-project-security-culture/v10/7-Security_Testing/

235. NIST SP 800-160. Engineering Trustworthy Secure Systems. URL: https://csrc.nist.gov/pubs/sp/800/160/v1/r1/final

236. NIST SP 800-100. Information Security Handbook: A Guide for Managers. URL: https://csrc.nist.gov/pubs/sp/800/100/upd1/final

237. ISO/IEC 27000 family. Information security management. URL: https://www.iso.org/standard/iso-iec-27000-family

ДОДАТКИ

Додаток А.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Наукові праці, в яких опубліковані основні наукові результати дисертації:

1. Dyka A. I. Detection of SVD-based hidden data in web applications using Walsh-Hadamard transformant analysis. Scientific notes of V. I. Vernadsky Taurida National University. 2025. Vol. 36 (75), № 4. P. 110-118. DOI: https://doi.org/10.32782/2663-5941/2025.4.2/15

2. Dyka A. I. Detection of covert channels in web applications based on unimodality violation in the Walsh-Hadamard spectrum. Informatics and Mathematical Methods in Simulation. 2025. Vol.15, № 1. P. 5-14. DOI: https://doi.org/10.15276/imms.v15.no1.5

3. Dyka A. I. Intelligent methods for steganalysis of code-controlled covert messages in web applications. Scientific notes of V. I. Vernadsky Taurida National University. 2025. Vol. 36(75), № 2. P. 324-333. DOI: https://doi.org/10.32782/2663-5941/2025.2.2/44

4. Дика А. І., Трофименко О. Г., Ковальжі А. С. Інтелектуальне виявлення фішингових сторінок у середовищі вебзастосунків із застосуванням візуального подання. Вчені записки ТНУ імені В.І. Вернадського. Серія: Технічні науки. 2025. Т. 36(75), № 1. С. 298-308. DOI: https://doi.org/10.32782/2663-5941/2025.1.2/43

5. Трофименко О. Г., Дика А. І., Задерейко О. В., Шевченко О. В. Сфери застосування штучного інтелекту в розробці та тестуванні ігор. Кібербезпека: освіта, наука, техніка. 2025. № 1(29). С. 41-59. DOI: https://doi.org/10.28925/2663-4023.2025.29.832

6. Трофименко О. Г., Дика А. І., Лобода Ю. В., Толокнов А. А., Бондаренко М. Є. Штучний інтелект у розробці ігор. Кібербезпека: освіта,

наука, техніка. 2025. № 3(27). С. 109-119. DOI: https://doi.org/10.28925/2663-4023.2025.27.701

7. Трофименко О., Дика А., Логінова Н., Задерейко О., Струк Н. Штучний інтелект у військових навчальних симуляторах. Інформаційні технології та суспільство. 2024. № 2(13). 2024. С. 89-95. DOI: https://doi.org/10.32689/maup.it.2024.2.13

8. Трофименко О. Г., Лобода Ю. Г., Гура В. І., Дика А. І., Стрілець М. І. Інструменти штучного інтелекту для системного аналізу. Вісник Херсонського національного технічного університету. 2024. № 4. С. 349-357. DOI: https://doi.org/10.35546/kntu2078-4481.2024.4.46

9. Трофименко О. Г., Дика А. І., Лобода Ю. Г. Аналіз уразливостей та проблем безпеки вебзастосунків. Системні технології. 2023. № 3(146). С. 25-37. DOI: https://doi.org/10.34185/1562-9945-3-146-2023-03

10. Трофименко О. Г., Дика А. І., Лобода Ю. Г. Аналіз інструментів тестування вебзастосунків. Кібербезпека: освіта, наука, техніка. 2023. № 4(20). С. 62-71. DOI: https://doi.org/10.28925/2663-4023.2023.20.6271

11. Трофименко О. Г., Савєльєва О. С., Прокоп Ю. В., Логінова Н. І., Дика А. І. Soft skills для розробників програмного забезпечення. Кібербезпека: освіта, наука, техніка. 2023. № 3(19). С. 6-19. DOI: https://doi.org/10.28925/2663-4023.2023.19.619

Наукові праці, які засвідчують апробацію матеріалів дисертації:

12. Дика А. І. Застосування контекстуальних ембеддінгів для виявлення ознак шкідливих введень. Інформаційне суспільство: проблеми та перспективи: матер. X Всеукр. наук.-практ. конф. (Одеса, 23 травня 2025 р.). Одеса. 2025. С. 158-160. DOI: https://doi.org/10.32837/11300.29842

13. Дика А. І. Застосування NLP-методів для виявлення XSS-атак // Європейські орієнтири розвитку України: нові виміри у воєнний час :

матеріали міжнар. наук.-практ. конф. у 2 т. (Одеса, 16 травня 2025 р.). Одеса : Фенікс, 2025. Т. 2. С. 520-522. URL: https://hdl.handle.net/11300/30820

14. Дика А. І. Сучасні інтерактивні інструменти тестування безпеки застосунків // Інформаційні технології: моделі, алгоритми, системи (ITMAS-2024) : матеріали V Всеукр. наук.-практ. інтернет-конф. (м. Миколаїв, 30-31 жовт. 2024 р.). Миколаїв : НУК імені адмірала Макарова, 2024. С. 188-190. URL: https://itconf.nuos.edu.ua/2024/publications/analysis-of-project-management-software/

15. Дика А. І. Роль штучного інтелекту у тестуванні безпеки вебзастосунків // Інформаційне суспільство: проблеми та перспективи : матер. IX всеукр. наук.-практ. конф. (м. Одеса, 24 трав. 2024 р.). Одеса : НУ «ОЮА», 2024. С. 153-156. DOI: https://doi.org/10.32837/11300.27842

16. Дика А. І. Тестування безпеки вебзастосунків за допомогою ChatGPT // Актуальні тенденції розвитку освіти, науки та технологій : матеріали VII міжнар. наук.-практ. конф. (м. Харків-Бахмут, 15 трав. 2024 р.) Т. 2. С. 28-30. URL: https://www.nnppi.in.ua/index.php/abit/2-uncategorised/270-naukovi-konferentsiyi

17. Дика А. І., Трофименко О. Г. Аналіз потенційних загроз засобами ChatGPT для тестування безпеки вебзастосунків // Європейські орієнтири розвитку України: науково-практичний вимір в умовах воєнних викликів : матеріали міжнар. наук.-практ. конф. (м. Одеса, 26 квіт. 2024 р.) Одеса : НУ «ОЮА», 2024. С. 876-878. URL: https://hdl.handle.net/11300/28232

18. Дика А. І., Максименко А. Ж. Штучний інтелект у веброзробці: революція персоналізації користувача // Кіберпростір в умовах війни та глобальних викликів XXI століття: теорія та практика : матеріали міжнар. наук.-практ. конф. (м. Одеса, 24 листоп. 2023 р.). Одеса : НУ «ОЮА», 2023. С. 174-176. DOI: https://doi.org/10.32837/11300.27179

19. Дика А. І. Тестування штучного інтелекту: ключові виклики, стратегії вдосконалення // Електронні інформаційні ресурси: створення, використання, доступ : матеріали міжнар. наук.-практ. інтернет-конф. (м. Вінниця, 20-21

листоп. 2023 р.). Вінниця : КЗВО «Вінницька академія безперервної освіти». С. 93-95.

20. Манаков С. Ю., Дика А. І. Вибір сучасних засобів розробки мобільних застосунків // Інформаційне суспільство: проблеми та перспективи : матеріали VIII всеукр. наук.-практ. конф. (Одеса, 2 червня 2023 р.). Одеса: НУ «ОЮА», 2023. С. 86-89. DOI: https://doi.org/10.32837/11300.25263

21. Дика А. І., Лобода Ю. Г. Розвиток застосування WebAssembly у веброзробці // Європейські орієнтири розвитку України в умовах війни та глобальних викликів XXI століття: синергія наукових, освітніх та технологічних рішень: у 2 т. : матеріали міжнар. наук.-практ. конф. (м. Одеса, 19 трав. 2023 р.). Одеса : Юридика, 2023. Т. 1. С. 594-596. URL: https://hdl.handle.net/11300/27641

22. Трофименко О. Г., Дика А. І. Проблеми тестування вебсайтів е-комерції // Інформаційні технології та менеджмент у вищій освіті та науці : матеріали міжнар. наук. конф. (м. Фергана, 28 листоп. 2022 р.). Izdevnieciba «Baltija Publishing», 2022. С. 195-199.

23. Дика А. І. Регресійне тестування в Agile // Кібербезпека в сучасному світі: актуальні виклики : матеріали IV всеукр. наук.-практ. конф. (м. Одеса, 25 листоп. 2022 р.). Одеса : НУ «ОЮА», 2022. С. 44-48. URL: https://hdl.handle.net/11300/28986 ; DOI: 10.32837/11300.28986.

24. Логінова Н. І., Дика А. І., Янковський О. Г. Аналіз вразливостей систем керування базами даних // Європейський вибір України, розвиток науки та національна безпека в реаліях масштабної військової агресії та глобальних викликів XXI століття (до 25-річчя Національного університету «Одеська юридична академія» та 175-річчя Одеської школи права) : матеріали міжнар. наук.-практ. конф. (м. Одеса, 17 черв. 2022 р.). Одеса: Видавничий дім «Гельветика», 2022. Т. 1. С. 682-685. URL: https://hdl.handle.net/11300/19760

Наукові праці, які додатково відображають наукові результати дисертації:

25. Концептуальні основи захисту інформаційного суверенітету України : монографія / Задерейко О. В., Троянський О. В., Чанишев Р. І., Дика А. І. Одеса : Фенікс, 2022. 220 с. DOI: https://doi.org/10.32837/11300.22733

26. Тестування та забезпечення якості програмних систем : навч.-метод. посіб. для підгот. здобув. вищої освіти галузі знань 12 «Інформаційні технології» / Нац. ун-т «Одес. юрид. академія»; уклад.: О. Г. Трофименко, А. І. Дика ; Одеса : Фенікс, 2024. 140 с. URL: https://hdl.handle.net/11300/27246 ; DOI: https://doi.org/10.32837/11300.27246

27. Тестування та забезпечення якості програмних систем : навч. посіб. для підгот. здобув. вищої освіти галузі знань 12 «Інформаційні технології» / О. Г. Трофименко, А. І. Дика; Нац. ун-т «Одес. юрид. академія». Одеса: Фенікс, 2024. 195 с. URL: https://hdl.handle.net/11300/27717. ISBN 978-617-8395-88-9.

28. Комп’ютерна програма «Black Sea Hanter»: авторське свідоцтво № 129198 від 21.08.2024 / Струк Н. О., Дика А. І., Задерейко О. В., Логінова Н. І., Трофименко О. Г. https://sis.nipo.gov.ua/uk/search/detail/1821799/

ВІДОМОСТІ ПРО АПРОБАЦІЮ МАТЕРІАЛІВ ДИСЕРТАЦІЇ

Положення, висновки та рекомендації, викладені у дисертації, доповідалися на: міжнародній науково-практичній конференції «Європейський вибір України, розвиток науки та національна безпека в реаліях масштабної військової агресії та глобальних викликів XXI століття» (м. Одеса, 17 червня 2022 р.) – особиста участь; всеукраїнській науково-практичній конференції «Кібербезпека в сучасному світі: актуальні виклики» (м. Одеса, 25 листопада 2022 р.) – особиста участь; міжнародній науковій конференції «Інформаційні технології та менеджмент у вищій освіті та науці» (м. Фергана, Республіка Узбекистан, 28 листопада 2022 р.) – заочна участь; міжнародній науково-практичній конференції «Європейські орієнтири розвитку України в умовах війни та глобальних викликів XXI століття: синергія наукових, освітніх та технологічних рішень» (м. Одеса, 19 травня 2023 р.) – особиста участь; всеукраїнській науково-практичній конференції «Інформаційне суспільство: проблеми та перспективи» (м. Одеса, 2 червня 2023 р.) – особиста участь; міжнародній науково-практичній конференції «Електронні інформаційні ресурси: створення, використання, доступ» (м. Вінниця, 20-21 листопада 2023 р.) – заочна участь; міжнародній науково-практичній конференції «Кіберпростір в умовах війни та глобальних викликів XXI століття: теорія та практика» (м. Одеса, 24 листопада 2023 р.) – особиста участь; міжнародній науково-практичній конференції «Європейські орієнтири розвитку України: науково-практичний вимір в умовах воєнних викликів» (м. Одеса, 26 квітня 2024 р.) – особиста участь; міжнародній науково-практичній конференції «Актуальні тенденції розвитку освіти, науки та технологій» (м. Харків – Бахмут, 15 травня 2024 р.) – заочна участь; всеукраїнській науково-практичній конференції «Інформаційне суспільство: проблеми та перспективи» (м. Одеса, 24 травня 2024 р.) – особиста участь; всеукраїнській науково-практичній інтернет-конференції «Інформаційні технології: моделі, алгоритми, системи» (м. Миколаїв, 30-31 жовтня 2024 р.) – заочна участь;

міжнародній науково-практичній конференції «Європейські орієнтири розвитку України: нові виміри у воєнний час» (м. Одеса, 16 травня 2025 р.) – особиста участь; всеукраїнській науково-практичній конференції «Інформаційне суспільство: проблеми та перспективи» (м. Одеса, 23 травня 2025 р.) – особиста участь.

Додаток Б.

ДОКУМЕНТИ ЩОДО ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕНЬ

«ЗАТВЕРДЖУЮ»
Ректор
Національного університету
«Одеська юридична академія»
_____ проф. Олег ТОДОЩАК
«_1_» _____ 2025 р.

АКТ ВПРОВАДЖЕННЯ

результатів дисертаційного дослідження Дикої Анастасії Іванівни «Методи машинного навчання в тестуванні безпеки веборієнтованого програмного забезпечення»

Комісія у складі: проректора з навчальної роботи, доктора юридичних наук, професора Галини УЛЬЯНОВОЇ, завідувача кафедри інформаційних технологій, кандидата педагогічних наук, доцента Наталії ЛОГІНОВОЇ, доцента кафедри інформаційних технологій, кандидата технічних наук, доцента Олени ТРОФИМЕНКО засвідчує, що дисертаційне дослідження аспіранта Дикої Анастасії Іванівни, представлене на здобуття наукового ступеня доктора філософії за спеціальністю 122 «Комп'ютерні науки», відповідає напрямку наукових робіт кафедри інформаційних технологій та тематиці досліджень її наукового керівника. Результати дослідження Дикої Анастасії Іванівни, викладені у численних наукових та навчальних публікаціях, використовуються в навчальному процесі Національного університету «Одеська юридична академія» при викладанні навчальних дисциплін «Тестування та забезпечення якості програмних систем», «Вебтехнології та вебдизайн», «Проєктування інформаційних систем», «Захист програм і даних», «DevOps». Матеріали дисертації використовуються також під час дипломного проєктування та в науково-дослідній роботі студентів кафедри.

Необхідність впровадження результатів дослідження зумовлена актуальними викликами цифрового середовища, зокрема:

- зростанням кіберзагроз (фішингові атаки, приховані канали передавання даних);
- потребою у підвищенні надійності цифрових сервісів;
- забезпеченні конфіденційності даних користувачів;
- стійкості критично важливої інформаційної інфраструктури.

Запропоновані підходи дозволяють:

- вдосконалити методи тестування безпеки вебзастосунків;
- інтегрувати сучасні механізми захисту в межах концепції DevSecOps;
- формувати комплексну модель кіберзахисту цифрової інфраструктури.

Впровадження результатів дисертації здійснюється через:

- видання навчального (https://hdl.handle.net/11300/27717, ISBN 978-617-8395-32-2) та навчально-методичного (https://hdl.handle.net/11300/27246, ISBN 978-617-8395-88-9) посібників з дисципліни «Тестування та забезпечення якості програмних систем»;
- використання теоретичного базису для виявлення прихованих каналів на основі концепції кодового управління, що відкриває нові можливості захисту мультимедійних даних у веборієнтованому програмному забезпеченні;
- дослідження комплексу методів стеганоаналізу на основі машинного навчання для виявлення сучасних типів стеганографічних повідомлень (зокрема з кодовим управлінням та на основі SVD-UV), орієнтованих на роботу в реальному часі та інтеграцію у Runtime Security вебзастосунків;
- інтеграцію алгоритмів штучного інтелекту та машинного навчання в процедури автоматизованого тестування безпеки, що дозволяє створити адаптивні методи виявлення фішингу з високою точністю та придатністю для CI/CD-конвеєрів і Runtime-контурів захисту.

Результати дослідження планується і надалі використовувати при підготовці магістрів та докторів філософії. Це сприятиме формуванню у студентів системного розуміння сучасних методів тестування безпеки вебзастосунків, а також розвитку навичок їх удосконалення та практичного застосування.

Члени комісії:

Проректор з навчальної роботи,
доктор юридичних наук, професор Галина УЛЬЯНОВА

Завідувач кафедри
інформаційних технологій,
кандидат педагогічних наук, доцент Наталія ЛОГІНОВА

Доцент кафедри
інформаційних технологій,
кандидат технічних наук, доцент Олена ТРОФИМЕНКО

Товариство з обмеженою відповідальністю «ЕПАМ СИСТЕМЗ»

вул. Прахових Сімї, 54 , Київ, 01033, Україна | Т: +38 044 390 54 57
e-mail: info.ua@epam.com | web: www.epam.com

ЄДРПОУ 33880213, п/р UA243003350000000260032198752 в АТ «Райффайзен Банк»

« 16 » жовтня 2025 року **№ 475**

АКТ

про впровадження у діяльність результатів дисертаційних дослідження Дикої Анастасії Іванівни на тему:

«Методи машинного навчання в тестуванні безпеки веборієнтованого програмного забезпечення»

Цим актом підтверджується, що дослідження, представлене в дисертаційній роботі А.І. Дикої на тему «Методи машинного навчання в тестуванні безпеки веборієнтованого програмного забезпечення», є актуальним з огляду на зростання складності та частоти кіберзагроз. У сучасних умовах, коли веб-застосунки залишаються однією з основних цілей для атак, критично важливим є впровадження інноваційних підходів до їх захисту. Особливу увагу привертають стеганографічні методи прихованої передачі даних, які дедалі частіше використовуються у шкідливих цілях. Це зумовлює потребу в розробці ефективних інструментів для виявлення подібних загроз із використанням технологій машинного навчання.

Метод автоматичного розпізнавання фішингових вебсторінок, розроблений на основі візуалізації HTML-коду та подальшої класифікації за допомогою методів машинного навчання, має високу практичну значущість. Для перевірки його працездатності та ефективності було проведено тестування в умовах реального використання корпоративної мережі. Протягом місяця система працювала в режимі дослідної експлуатації, інтегрована до веб-проксі підприємства, що дозволяло автоматично аналізувати сторінки, на які переходили працівники. Для оцінки ефективності було змодельовано кілька типових фішингових атак із використанням підроблених сторінок авторизації поштових сервісів, систем електронного документообігу та банківських порталів.

У результаті дослідної експлуатації було зафіксовано 147 спроб переходу співробітників на фішингові ресурси — усі вони були вчасно заблоковані системою. Середній час затримки під час обробки запитів не перевищував 0,08 секунди, що не вплинуло на комфорт користувачів. Частка хибнопозитивних спрацювань становила близько 7%, що є прийнятним показником для промислового використання. Порівняно з попереднім періодом, кількість звернень до служби підтримки з приводу підозрілих листів і посилань зменшилася на третину.

Важливим результатом впровадження став підтверджений економічний ефект. До початку експлуатації ризики, пов'язані з успішними фішинговими інцидентами, оцінювалися як суттєві та здатні завдати підприємству значних збитків. У ході дослідної експлуатації система забезпечила запобігання щонайменше 93% потенційних інцидентів, пов'язаних із фішингом, які могли призвести до втрати конфіденційних даних і фінансових витрат. Крім того, завдяки зменшенню навантаження на службу підтримки та скороченню часу реагування на інциденти, підприємство досягло зниження загальних витрат на забезпечення інформаційної безпеки приблизно на 12%.

Таким чином, метод розпізнавання фішингових вебсторінок підтвердив свою ефективність і практичну придатність, забезпечив підвищення рівня захищеності інформаційної інфраструктури та продемонстрував відчутний економічний ефект. Система рекомендована до подальшої постійної експлуатації та масштабування на інші цифрові сервіси підприємства.

Акт наданий для представлення за місцем захисту дисертаційної роботи.

Даний акт не є підставою для фінансових розрахунків.

Генеральний директор
ТОВ «ЕПАМ СИСТЕМЗ» **Надія БАБЕЙКО**

Прим. № 1

АДМІНІСТРАЦІЯ ДЕРЖСПЕЦЗВ'ЯЗКУ

УПРАВЛІННЯ
ДЕРЖАВНОЇ СЛУЖБИ СПЕЦІАЛЬНОГО ЗВ'ЯЗКУ ТА ЗАХИСТУ
ІНФОРМАЦІЇ УКРАЇНИ В ОДЕСЬКІЙ ОБЛАСТІ
Управління Держспецзв'язку в Одеській області

вул. Єврейська, 43, м. Одеса, 65045, тел. (048)722-71-57,
e-mail: odesa@cip.gov.ua, web: cip.gov.ua
Код ЄДРПОУ 34800034

від 02.03.2026 р. № 40-29 На № _____ від _____20___ р.

Довідка
про впровадження результатів дисертаційного дослідження на тему: «Методи машинного навчання в тестуванні безпеки веборієнтованого програмного забезпечення» аспірантки Дикої Анастасії Іванівни

Запропонований Дикою А.І. метод виявлення прихованих каналів передачі інформації ґрунтується на аналізі статистичних зсувів у гістограмах трансформант перетворення Уолша-Адамара, що відображають приховані зміни у сингулярних векторах U та V під час стеганографічного вбудовування. Для класифікації використано ансамбль моделей на основі методу опорних векторів (SVM), що дозволяє ефективно виявляти слабко виражені ознаки та забезпечує високу стійкість до перенавчання. Результати експериментального тестування при проведенні дослідження показали точність виявлення понад 92%, що перевищує показники наявних класичних статистичних методів аналізу та забезпечує надійне розпізнавання прихованих каналів навіть у складних умовах.

Інтеграція методу у роботу Управління Державної служби спеціального зв'язку та захисту інформації України в Одеській області сприятиме своєчасному виявленню каналів прихованої передачі даних, які використовували сучасні стійкі стеганографічні підходи. Використання SVD-UV методу дозволило автоматизувати процес контролю, підвищити рівень захисту інформаційної інфраструктури та знизити ризики витоку конфіденційних даних. Таким чином,

розроблений метод успішно пройшов апробацію, довів свою ефективність у реальному середовищі та став важливим елементом комплексної системи захисту вебзастосунків від прихованих загроз.

Враховуючи викладене вище, можна дійти висновку, що напрацювання автора заслуговують на практичне застосування у діяльності Управління Державної служби спеціального зв'язку та захисту інформації України в Одеській області.

Начальник
Управління Державної служби спеціального зв'язку
та захисту інформації України
в Одеській області **Володимир ДЯТЛОВСЬКИЙ**