МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Кваліфікаційна наукова
праця на правах рукопису

ГАМЗА ДМИТРО ЄВГЕНІЙОВИЧ

УДК 004.056

ДИСЕРТАЦІЯ

МЕТОДИ ВИЯВЛЕННЯ ШКІДЛИВОЇ АКТИВНОСТІ В ІНФОРМАЦІЙНІЙ СИСТЕМІ ОРГАНІЗАЦІЇ НА ОСНОВІ ГІБРИДНОЇ КЛАСИФІКАЦІЇ

Спеціальність 125 «Кібербезпека»

Подається на здобуття наукового ступеня доктора філософії

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

Науковий керівник:
ЗИБІН Сергій Вікторович
доктор технічних наук, професор

КИЇВ - 2026

АНОТАЦІЯ

Гамза Д.Є. Методи виявлення шкідливої активності в інформаційній системі організації на основі гібридної класифікації. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 125 «Кібербезпека». – Державний університет інформаційно-комунікаційних, Київ, 2026.

Дисертаційна робота присвячена вирішенню актуального наукового завдання щодо розробки методу виявлення шкідливої активності в інформаційній системі організації на основі гібридної класифікації, що забезпечує підвищення точності виявлення загроз при зниженні кількості хибних спрацювань та дотриманні вимог обробки в реальному часі.

Об'єктом дослідження дисертаційної роботи є процес виявлення шкідливої активності в інформаційних системах організацій.

Предметом дослідження дисертаційної роботи є методи виявлення шкідливої активності в інформаційних системах організацій.

Метою дослідження є підвищення точності виявлення шкідливої активності в інформаційній системі організації на основі гібридної класифікації.

Наукове завдання – розробка методів виявлення шкідливої активності в інформаційній системі організації на основі гібридної класифікації.

Для досягнення поставленої мети та вирішення наукового завдання, було отримано наступні наукові результати:

1. *Вперше* розроблено метод виявлення шкідливої активності в інформаційній системі організації на основі гібридної класифікації, який базується на використанні гетерогенного набору базових класифікаторів та мета-класифікаторі на основі XGBoost для дворівневої стекінг-архітектури, що дає можливість застосовувати різнорідні ансамблеві методи машинного навчання у системах виявлення шкідливої активності в режимі реального часу з можливістю динамічного переналаштування класифікаторів.

2. *Удосконалено* метод комплексної оптимізації вхідного набору даних методу виявлення шкідливої активності в інформаційній системі, який, на відміну від існуючих, поєднує балансування класів (SMOTE), нормалізацію (Min-Max) та зниження розмірності (PCA), що дозволило зменшити обчислювальне навантаження та підвищити точність методу гібридної класифікації.

3. *Набув подальшого розвитку* метод багатокритеріального вибору оптимальної архітектури системи виявлення шкідливої активності у режимі реального часу, в якому, за рахунок використання послідовного застосування стратегії фільтрації за середніми значеннями та побудови фронту Парето забезпечується баланс між максимальною точністю виявлення шкідливої активності та мінімальними витратами обчислювальних ресурсів.

У вступі обґрунтовано актуальність обраної теми дисертаційного дослідження, визначено основні положення, наукову та практичну цінність отриманих результатів.

У першому розділі проведено комплексний аналіз проблематики виявлення шкідливої активності в інформаційних системах організацій. Здійснено класифікацію та порівняльний аналіз існуючих методів виявлення шкідливої активності: сигнатурних, аномальних, поведінкових та гібридних. Встановлено, що сигнатурні підходи нездатні виявляти раніше невідомі атаки, а окремі класифікатори на основі машинного навчання мають обмежену узагальнюючу здатність та високий рівень хибних спрацювань. Проаналізовано ансамблеві методи та обґрунтовано перспективність стекінг-архітектури для задачі виявлення шкідливої активності.

У другому розділі розроблено метод виявлення шкідливої активності на основі гібридної класифікації. Обґрунтовано вибір гетерогенного набору базових класифікаторів (XGBoost, CatBoost, LightGBM, Random Forest, SVM, KNN) та мета-класифікаторів (логістична регресія, XGBoost) для дворівневої стекінг-архітектури. Виконано математичну формалізацію моделі як багатокритеріальної оптимізаційної задачі. Удосконалено модель простору ознак, що включає статистичні, поведінкові, контекстні та темпоральні характеристики мережевого

трафіку. Розроблено методику попередньої обробки даних, яка поєднує балансування класів за допомогою SMOTE, Min-Max нормалізацію та зниження розмірності методом головних компонент.

У третьому розділі проведено експериментальне дослідження ефективності розробленого методу на датасеті CSE-CIC-IDS2018. Розроблено комплексний метод оптимізації набору даних (SMOTE + Min-Max + PCA), який забезпечив редукцію ознакового простору з 80 до 18 компонент зі збереженням 95% дисперсії. Визначено критерії оцінювання ефективності Accuracy, F1-score, час прогнозу та проведено дослідження базових класифікаторів і гібридних стекінг-моделей. Експериментально встановлено, що оптимальна конфігурація ансамблю XGBoost + CatBoost + LightGBM з мета-класифікатором XGBoost досягає Accuracy 98,07%, F1-score 96,57% та часу прогнозування 7,16мс, що на 3,87% та 5,11% відповідно перевищує результати на необробленому датасеті, а середній час прогнозування скорочено на 76% (з 29,37 до 7,28 мс). Проведено багатокритеріальний відбір оптимальних архітектур із застосуванням стратегії above-average rule та побудови фронту Парето, який дозволив обрати оптимальну модель.

У четвертому розділі систематизовано результати дослідження та представлено їх у вигляді програмного рішення, готового до впровадження. Розроблено п'ятимодульну архітектуру програмного рішення розробленого методу, який складається з модулів збору даних, інженерії ознак, попередньої обробки, гібридної класифікації та реагування. Запропоновано методику оцінки ефективності в умовах реальної експлуатації з використанням операційних метрик (Accuracy, F1-score, FPR), метрик продуктивності та організаційних показників (MTTD, MTTR).

Дисертація виконувалась в Державному університеті інформаційно-комунікаційних технологій.

Результати наукових досліджень було вправаджено в навчальний процес на кафедрі систем та технологій кібербезпеки Навчально-наукового інституту кібербезпеки та захисту інформації для освітніх компонент «Організація проведення наукових досліджень в кібербезпеці» та «Штучний інтелект».

Результати наукових досліджень було використано на кафедрі систем та технолоій кібербезпеки Навчально-наукового інституту кібербезпеки та захисту інформації під час виконання науково-дослідних робіт «Методологія виявлення шкідливих процесів в інформаційних системах (№ 0121U113613, м. Київ), ДУІКТ та «Розробка науково-методичних рекомендацій виявлення шкідливих процесів в інформаційній системі організації» (ТОВ «АЛЬФА-МЕТАЛ», м. Київ).

Також результати наукових досліджень прийняті до впровадження в діяльність ТОВ «Євротелеком» та ТОВ «АРВІОМ».

Практичне значення отриманих результатів полягає у розробці програмного рішення, яке може бути інтегроване у сучасні системи моніторингу безпеки інформаційних систем організацій. За результатами моделювання, таке програмне рішення щодо виявлення шкідливої активності в інформаційній системі організації на основі гібридної класифікації дозволяє забезпечити приріст точності на 3,87 % та F1-score на 5,11 %, а також скоротити найменший час прогнозування на 76 %, порівняно з відповідними результатами на необробленому датасеті. При здісненні багатокритеріального відбору оптимальних архітектур із використанням стратегії фільтрації за середніми значеннями та побудови фронту Парето програмне рішення забезпечує максимальну точність на рівні 0,9807 та мінімальну затримку на рівні 7,16 мс, що задовольняє вимогам до систем виявлення вторгнень реального часу.

Ключові слова: кібербезпека, інформаційна безпека, кіберзагрози, інформаційна система організації, ризики, стан безпеки, виявлення шкідливої активності, вторгнення, штучний інтелект, машинне навчання, обробка даних, гібридна класифікація, аналіз, моделювання, Python.

ANOTATION

Hamza D.Ye. Methods for detecting malicious activity in an organization's information system based on hybrid classification. – Qualifying scientific work on the rights of a manuscript.

Dissertation for obtaining the academic degree of Doctor of Philosophy in specialty 125 "Cybersecurity". – State University of Information and Communication Technologies, Kyiv, 2026.

The dissertation is devoted to solving the relevant scientific problem of developing a method for detecting malicious activity in an organization's information system based on hybrid classification, which ensures the improvement of threat detection accuracy with a reduced number of false positives and compliance with real-time processing requirements.

The object of the research is the process of detecting malicious activity in organizations' information systems.

The subject of the research is methods of detecting malicious activity in organizations' information systems.

The aim of the research is to improve the accuracy of detecting malicious activity in an organization's information system based on hybrid classification.

The scientific task – is the development of methods for detecting malicious activity in an organization's information system based on hybrid classification.

To achieve the set goal and solve the scientific problem, the following scientific results were obtained:

1. For the first time, a method for detecting malicious activity in an organization's information system based on hybrid classification has been developed, which is based on the use of a heterogeneous set of basic classifiers and a meta-classifier based on XGBoost for a two-level stacking architecture, which makes it possible to apply heterogeneous ensemble machine learning methods in real-time malicious activity detection systems with the possibility of dynamic reconfiguration of classifiers.

2. A method for comprehensive optimization of the input data set of the method for detecting malicious activity in an information system has been improved, which, unlike existing ones, combines class balancing (SMOTE), normalization (Min-Max) and dimensionality reduction (PCA), which allowed to reduce the computational load and increase the accuracy of the hybrid classification method.

3. The method of multi-criteria selection of the optimal architecture of a real-time malicious activity detection system has been further developed, in which, through the use of consistent application of the filtering strategy by average values and

construction of the Pareto front, a balance between maximum accuracy of malicious activity detection and minimum consumption of computing resources is ensured.

The introduction justifies the relevance of the selected dissertation research topic, outlines the main tenets, and establishes the scientific and practical significance of the obtained results.

In the first chapter, a comprehensive analysis of the problem of detecting malicious activity in organizations' information systems is conducted. Classification and comparative analysis of existing methods of malicious activity detection are carried out: signature-based, anomaly-based, behavioral and hybrid. It is established that signature-based approaches are unable to detect previously unknown attacks, and individual classifiers based on machine learning have limited generalization ability and a high level of false positives. Ensemble methods have been analyzed and the perspective of the stacking architecture for the task of malicious activity detection has been substantiated.

In the second chapter, a method for detecting malicious activity based on hybrid classification has been developed. The choice of a heterogeneous set of base classifiers (XGBoost, CatBoost, LightGBM, Random Forest, SVM, KNN) and meta-classifiers (logistic regression, XGBoost) for a two-level stacking architecture is substantiated. A mathematical formalization of the model as a multi-criteria optimization problem has been performed. The feature space model has been improved to include statistical, behavioral, contextual and temporal characteristics of network traffic. A data preprocessing methodology has been developed that combines class balancing using SMOTE, Min-Max normalization and dimensionality reduction by the principal component method.

In the third chapter, an experimental study of the effectiveness of the developed method has been conducted on the CSE-CIC-IDS2018 dataset. A comprehensive method for optimizing the dataset (SMOTE + Min-Max + PCA) has been developed, which provided a reduction of the feature space from 80 to 18 components while preserving 95% of the variance. The criteria for evaluating the effectiveness (Accuracy, F1-score, prediction time) were determined, and a study of base classifiers and hybrid stacking models was conducted. It has been experimentally established that the optimal ensemble configuration XGBoost + CatBoost + LightGBM with the XGBoost meta-classifier achieves Accuracy of 98.07%, F1-score of 96.57% and prediction time of 7.16 ms, which is 3.87% and 5.11% respectively higher than the results on the unprocessed dataset, and the average prediction time has been reduced by 76% (from 29.37 to 7.28 ms). A multi-criteria selection of optimal architectures was carried out using the above-average rule strategy and the construction of the Pareto front, which allowed selecting the optimal model.

In the fourth chapter, the research results have been systematized and presented in the form of a software solution ready for implementation. A five-module architecture of

the software solution of the developed method has been developed, consisting of data collection, feature engineering, preprocessing, hybrid classification and response modules. A methodology for evaluating effectiveness in real operating conditions has been proposed using operational metrics (Accuracy, F1-score, FPR), performance metrics and organizational indicators (MTTD, MTTR).

The dissertation was carried out at the State University of Information and Communication Technologies.

The research results were implemented in the educational process at the Department of Cybersecurity Systems and Technologies of the Educational and Research Institute of Cybersecurity and Information Protection for the educational components "Organization of Scientific Research in Cybersecurity" and "Artificial Intelligence".

The research results were used at the Department of Cybersecurity Systems and Technologies of the Educational and Research Institute of Cybersecurity and Information Protection during the execution of the research and development works "Methodology for detecting malicious processes in information systems" (No. 0121U113613, Kyiv), SUICT, and "Development of scientific and methodological recommendations for detecting malicious processes in an organization's information system" (LLC "ALPHA-METAL", Kyiv).

Also, the research results have been accepted for implementation in the activities of LLC "Eurotelecom" and LLC "ARVIOM".

The practical significance of the obtained results lies in the development of a software solution that can be integrated into modern security monitoring systems of organizations' information systems. According to the modeling results, such a software solution for detecting malicious activity in an organization's information system based on hybrid classification allows an increase in accuracy by 3.87% and F1-score by 5.11%, as well as a reduction in the minimum prediction time by 76% compared to the corresponding results on the unprocessed dataset. By carrying out a multi-criteria selection of optimal architectures using the filtering strategy by average values and constructing the Pareto front, the software solution provides a maximum accuracy of 0.9807 and a minimum latency of 7.16 ms, which meets the requirements for real-time intrusion detection systems.

Keywords: cybersecurity, information security, cyber threats, organization information system, risks, security state, malicious activity detection, intrusion, artificial intelligence, machine learning, data processing, hybrid classification, analysis, modeling, Python.

СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Наукові праці, в яких опубліковані основні наукові результати дисертації:

1. Гайдур Г. І., Гахов С. О., Гамза Д. Є. Модель виявлення шкідливої активності в інформаційній системі організації на основі гібридної класифікації. *Сучасний захист інформації*. 2024. № 4(60). С. 30–38. https://doi.org/10.31673/2409-7292.2024.040003.

2. Haidur H., Gakhov S., Hamza D. Using support vectors to build a rule-based system for detecting malicious processes in an organisation's network traffic. *Informatyka, Automatyka, Pomiary W Gospodarce I Ochronie Środowiska*. 2024. Vol. 14, No. 4. P. 90–96. https://doi.org/10.35784/iapgos.6366 .

3. Гайдур Г. І., Гамза Д. Є. Гібридний метод виявлення шкідливої активності на основі стекінг-ансамблю класифікаторів. *Сучасний захист інформації*. 2025. № 3(63). С. 20–26. https://doi.org/10.31673/2409-7292.2025.030315.

4. Гамза, Д. (2025). Вплив оптимізації датасету CSE–CIC–IDS2018 на ефективність гібридної стекінгової моделі виявлення мережевих вторгнень. *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»*, 2(30), 766–777. https://doi.org/10.28925/2663-4023.2025.30.963.

Наукові праці, які засвідчують апробацію матеріалів дисертації:

1. Haydur, H., & Hamza, D. (2025, November 4–5). Hybrid method for malicious activity detection in information systems. In *Digital Transformation: Strengthening the Cybersecurity Capacities in the Modern World* (pp. 39–41).

2. Смолєв, Є. С., & Гамза, Д. Є. (2024, 26 квітня). Метод оптимізації даних для тренування моделі виявлення вторгнень на основі SVM. *Цифрова трансформація кібербезпеки: збірник тез наук.-практ. конф.* (с. 220–223). РВЦ ДУІКТ. https://duikt.edu.ua/uploads/n_12581_11703414.pdf

3. Haydur, H., & Hamza, D., Zybin S. (2026, 29 квітня) Multi-criteria selection of optimal hybrid stacking ensemble model for intrusion detection systems: pareto front and above-average rule filtering *Цифрова трансформація кібербезпеки:*

збірник тез наук.-практ. конф. (с. 27-30). РВЦ ДУІКТ. https://duikt.edu.ua/uploads/p_3086_30075970.pdf.

Наукові праці, які додатково відображають наукові результати дисертації:

1. Савченко В. А., Смолєв Є. С., Гамза Д. Є. Методика виявлення аномалій взаємодії користувачів з інформаційними ресурсами організації. *Сучасний захист інформації*. 2023. № 4. С. 6–12. https://doi.org/10.31673/2409-7292.2023.030101.

2. Волошко Д. С., Гамза Д. Є., Смолєв Є. С. Технологія виявлення простих вірусів у програмному коді. *Сучасний захист інформації*. 2023. № 2(54). С. 41–49. https://doi.org/10.31673/2409-7292.2023.020006.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

AUC-ROC – Area Under the Receiver Operating Characteristic Curve

APT – Advanced Persistent Threat

CEF – Common Event Format

CPU – Central Processing Unit

CSECICIDS2018 – Communications Security Establishment – Canadian Institute for Cybersecurity Intrusion Detection System 2018

CUDA – Compute Unified Device Architecture

DDoS – Distributed Denial of Service

DoS – Denial of Service

DPDK – Data Plane Development Kit

FN – False Negative

FP – False Positive

FPR – False Positive Rate

GPU – Graphics Processing Unit

HPA – Horizontal Pod Autoscaler

IAT – Inter-Arrival Time

IDS – Intrusion Detection System

IPS – Intrusion Prevention System

KNN – K-Nearest Neighbors

LEEF – Log Event Extended Format

MTTD – Mean Time to Detect

MTTR – Mean Time to Respond

ML – Machine Learning

MLP – Multilayer Perceptron

NIC – Network Interface Card

PCA – Principal Component Analysis

RAM – Random Access Memory

REST API	–Representational State Transfer Application Programming Interface
ROC	– Receiver Operating Characteristic
SIEM	– Security Information and Event Management
SMOTE	– Synthetic Minority Over-sampling Technique
SOAR	– Security Orchestration, Automation and Response
SOC	– Security Operations Center
SVM	– Support Vector Machine
TAP	– Test Access Point
TCP	– Transmission Control Protocol
TN	– True Negative
TP	– True Positive

ВСТУП

Актуальність теми. Стрімкий розвиток інформаційних технологій і глобальна цифровізація бізнес-процесів обумовили формування інформаційних систем організацій як критично важливого активу, що потребує комплексного захисту. За даними звіту ENISA Threat Landscape 2024, в Європейському Союзі за період липень 2024 – червень 2025 зафіксовано близько 4 900 верифікованих кіберінцидентів, понад 60 % яких розпочиналися з фішингу як первинного вектору атаки. Згідно з IBM Cost of a Data Breach Report 2024, середній час виявлення та стримування витоку даних становить 241 день, що засвідчує неспроможність значної частини організацій своєчасно реагувати на сучасні загрози [1, 2, 3, 7].

Традиційні засоби захисту інформаційних систем – сигнатурні системи виявлення шкідливої активності, поведінкові евристики та моно-методи машинного навчання – не забезпечують необхідного рівня точності класифікації загроз у реальному часі при допустимому рівні хибних спрацювань. Сигнатурні підходи нездатні виявляти раніше невідомі атаки та цілеспрямовані тривалі атаки, аномальні методи генерують надмірну кількість хибних спрацювань, а моно-класифікатори на основі машинного навчання мають обмежену узагальнюючу здатність на гетерогенному мережевому трафіку [4, 9, 10, 16].

Проблематику виявлення шкідливої активності розглядали такі дослідники, як A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman (фундаментальний огляд систем виявлення вторгнень), G. Karatas, O. Demir, O. K. Sahingoz (підвищення ефективності ML-IDS на дисбалансованих наборах даних), N. V. Chawla (метод SMOTE), I. T. Jolliffe (метод головних компонент), D. H. Wolpert (стекінгова генералізація). Однак, попри значний обсяг отриманих результатів, невирішеним залишається питання гібридизації різнорідних ансамблевих методів машинного навчання для виявлення шкідливої активності в інформаційних системах організацій у режимі реального часу, що поєднувала б високу точність класифікації, низький рівень хибних спрацювань і прийнятну обчислювальну складність [8, 10, 41, 42, 73].

Виявлені протиріччя – між необхідністю виявлення нових типів атак у режимі реальному часі та статичною природою існуючих засобів захисту (зовнішнє протиріччя), а також між точністю детектування і швидкістю прийняття рішень (внутрішнє протиріччя) – обумовлюють актуальність обраної теми дисертаційного дослідження.

Зв'язок з науковими програмами, планами, темами. Дисертаційну роботу виконано на кафедрі систем та технологій кібербезпеки Навчально-наукового інституту кібербезпеки та захисту інформації Державного університету інформаційно-комунікаційних технологій (ДУІКТ) у межах науково-дослідних робіт «Методологія виявлення шкідливих процесів в інформаційних системах» (№ 0121U113613, м. Київ, ДУІКТ) та «Розробка науково-методичних рекомендацій виявлення шкідливих процесів в інформаційній системі організації» (ТОВ «АЛЬФА-МЕТАЛ», м. Київ).

Тематика дисертаційного дослідження відповідає пріоритетним напрямам розвитку науки і техніки в Україні, узгоджується із Законом України «Про основні засади забезпечення кібербезпеки України», Стратегією кібербезпеки України, а також відповідає вимогам міжнародних стандартів ISO/IEC 27001 та NIST Cybersecurity Framework.

Мета і завдання дослідження. Метою дисертаційного дослідження є є підвищення точності виявлення шкідливої активності в інформаційній системі організації на основі гібридної класифікації.

У відповідності до поставленої мети дисертаційної роботи, для вирішення наукового завдання в роботі сформульовані наступні часткові завдання дослідження*:*

Проаналізувати існуючі методи та підходи до виявлення шкідливої активності в інформаційній системі організації.

Розробити гібридний метод виявлення шкідливої активності, визначити його архітектуру, специфікацію компонентів та обґрунтувати вибір мета-класифікатора.

Розробити метод попередньої обробки даних для методу виявлення шкідливої активності на основі гібридної класифікації в інформаційних системах

організацій класифікації з метою зниження обчислювального навантаження та підвищення точності виявлення.

Розробити метод багатокритеріального відбору оптимальних архітектур ансамблевого навчання для виявлення шкідливих процесів у реальному часі.

Провести експериментальне дослідження розробленого методу виявлення шкідливої активності в інформаційній системі організаціїна на основі гібридної класифікації.

Розробити рекомендації щодо створення та впровадження програмного рішення на базі розробленого методу виявлення шкідливої активності в інформаційній системі організації на основі гібридної класифікації.

Об’єкт дослідження – процес виявлення шкідливої активності в інформаційних системах організацій.

Предмет дослідження –методи виявлення шкідливої активності в інформаційних системах організацій.

Методи дослідження. Дослідження проведено на основі системного підходу із застосуванням: методів машинного навчання та ансамблевого навчання (стекінг-архітектура); методів попередньої обробки даних (SMOTE, Min-Max нормалізація, метод головних компонент PCA); методів багатокритеріальної оптимізації (фронт Парето, стратегія above-average rule); методів статистичного аналізу та експериментального дослідження..

Наукова новизна одержаних результатів. Найважливіші результати дисертаційного дослідження, які становлять його наукову новизну, полягають у такому:

1. *Вперше* розроблено метод виявлення шкідливої активності в інформаційній системі організації на основі гібридної класифікації, який базується на використанні гетерогенного набору базових класифікаторів та мета-класифікаторі на основі XGBoost для дворівневої стекінг-архітектури, що дає можливість застосовувати різнорідні ансамблеві методи машинного навчання у системах виявлення шкідливої активності в режимі реального часу з можливіст ю динамічного переналаштування класифікаторів.

2. *Удосконалено метод* комплексної оптимізації вхідного набору даних методу виявлення шкідливої активності в інформаційній системі, який, на відміну від існуючих, поєднує балансування класів (SMOTE), нормалізацію (Min-Max) та зниження розмірності (PCA), що дозволило зменшити обчислювальне навантаження та підвищити точність методу гібридної класифікації.

3. *Набув подальшого розвитку* метод багатокритеріального вибору оптимальної архітектури системи виявлення шкідливої активності у режимі реального часу, в якому, за рахунок використання послідовного застосування стратегії фільтрації за середніми значеннями та побудови фронту Парето забезпечується баланс між максимальною точністю виявлення шкідливої активності та мінімальними витратами обчислювальних ресурсів.

Практичне значення одержаних результатів. Практичне значення отриманих результатів полягає у розробці програмного рішення, яке може бути інтегроване у сучасні системи моніторингу безпеки інформаційних систем організацій. За результатами моделювання, таке програмне рішення щодо виявлення шкідливої активності в інформаційній системі організації на основі гібридної класифікації дозволяє забезпечити приріст точності на 3,87 % та F1-score на 5,11 %, а також скоротити найменший час прогнозування на 76 %, порівняно з відповідними результатами на необробленому датасеті. При здісненні багатокритеріального відбору оптимальних архітектур із використанням стратегії фільтрації за середніми значеннями та побудови фронту Парето програмне рішення забезпечує максимальну точність на рівні 0,9807 та мінімальну затримку на рівні 7,16 мс, що задовольняє вимогам до систем виявлення вторгнень реального часу. Результати впроваджено у навчальний процес ДУІКТ та прийняті до впровадження в діяльність ТОВ «Євротелеком» та ТОВ «АРВІОМ».

Особистий внесок здобувача. Усі основні наукові положення, висновки та рекомендації, винесені на захист, отримані здобувачем особисто. У наукових працях, опублікованих у співавторстві, особистий внесок здобувача полягає в розробці концептуальної моделі гібридної класифікації [113, 123], формалізації методу побудови правил на основі опорних векторів [115], розробці методики

виявлення аномалій взаємодії користувачів [114], підході до сигнатурного аналізу простих вірусів [116], дослідженні впливу оптимізації датасету [117] та формулюванні концепції гібридного методу виявлення [124, 125].

Апробація результатів дисертації. Основні наукові положення, результати та висновки дисертаційного дослідження доповідалися та обговорювалися на Міжнародній науково-практичній конференції «Digital Transformation: Strengthening the Cybersecurity Capacities in the Modern World» (4–5 листопада 2025 р.), на Науково-практичній конференції «Цифрова трансформація кібербезпеки» (Київ, ДУІКТ, 26 квітня 2024 р.) та Науково-практичній конференції «Цифрова трансформація кібербезпеки» (Київ, ДУІКТ, 29 квітня 2026 р.), а також на засіданнях кафедри систем та технологій кібербезпеки ДУІКТ.

Публікації. За результатами дисертаційних досліджень опубліковано 9 наукових праць. Основні наукові положення викладено у 6 наукових статтях [113–117, 123], серед яких [113, 117, 123] опубліковані у спеціалізованих фахових виданнях, затверджених наказом МОН України, [115] опубліковано у закордонному науковому виданні, що індексується в Scopus. За матеріалами виступів на науково-технічних конференціях опубліковано 3 тези доповідей [124, 125, 126].

Структура та обсяг дисертаційної роботи.

Дисертація складається зі вступу, чотирьох розділів, висновків, списку використаних джерел із 126 найменувань на 12 сторінках. Загальний обсяг роботи становить 168 сторінок серед яких 136 сторінок основного тексту, 18 рисунків, 14 таблиць.

РОЗДІЛ 1 АНАЛІЗ СТАНУ ТА ПОСТАНОВКА ЗАВДАННЯ ЩОДО РОЗРОБКИ МЕТОДУ ВИЯВЛЕННЯ ШКІДЛИВОЇ АКТИВНОСТІ В ІНФОРМАЦІЙНІЙ СИСТЕМІ ОРГАНІЗАЦІЇ НА ОСНОВІ ГІБРИДНОЇ КЛАСИФІКАЦІЇ

1.1. Аналіз проблеми виявлення шкідливої активності в інформаційній системі організації

Сучасні інформаційні системи організацій стикаються з постійно зростаючою кількістю та складністю кібератак. За даними аналітичних звітів, щороку фіксується збільшення кількості інцидентів інформаційної безпеки на 15-20%, що створює критичні ризики для функціонування бізнес-процесів, збереження конфіденційних даних та забезпечення безперервності діяльності підприємств.

Інформаційна система організації являє собою комплекс апаратних та програмних засобів, що забезпечують збирання, обробку, зберігання та передачу даних, необхідних для підтримки бізнес-процесів. Типова корпоративна система включає сервери додатків, бази даних, мережеву інфраструктуру, робочі станції користувачів та засоби захисту інформації (рис.1.1). Взаємодія між цими компонентами генерує значні обсяги мережевого трафіку та системних журналів, які є основним джерелом даних для виявлення шкідливої активності [9, 10].

Під шкідливими процесами в інформаційній системі організації, будемо розуміти атаки, які проводить зловмисник для використання уразливості інформаційної системи, що призводить до порушення доступності, цілісності та конфіденційності оброблюваної інформації [1].

Виявлення таких шкідливих процесів дозволяє усунення вразливості інформаційної системи, що призводить до зупинення реалізації атаки.

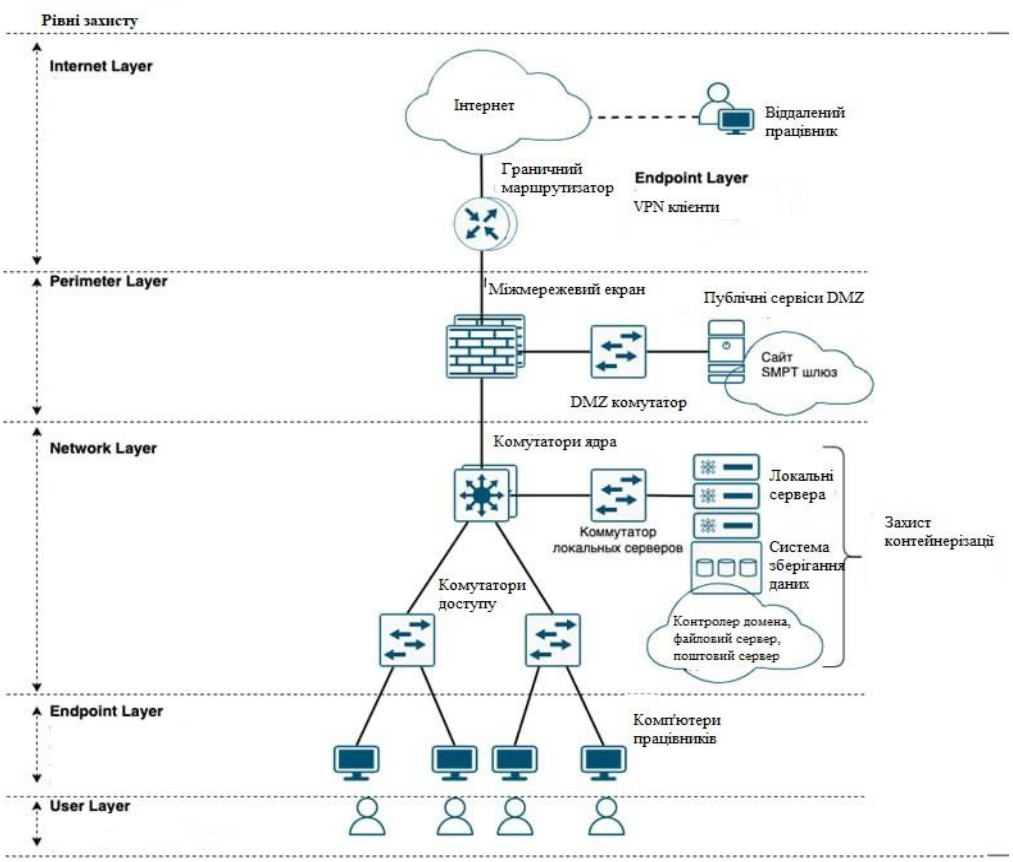

Рис.1.1. Приклад інформаціної системи організації

Існує три типи атак:

Розвідка. Ці атаки включають ping sweeps, передачу DNS зони, розвідку за допомогою e-mail, сканування TCP - або UDP портів і, можливо, аналіз загальнодоступних серверів з метою знаходження cgi-дір.

Експлойт (exploit - використовувати у своїх інтересах, зловживати). Це комп'ютерна програма, елемент програмного коду або послідовність команд, що використовують вразливості в програмному забезпеченні та застосовуються для проведення атаки на обчислювальну систему [9]. Метою атаки може бути як отримання контролю над системою (підвищення привілеїв користувача), так і порушення її функціонування (DoS-атака). Порушники можуть використовувати переваги прихованих можливостей або помилок для отримання несанкціонованого доступу до системи [11, 12].

Відмова в обслуговуванні (Denial of Service, DoS). При такій атаці порушник намагається завдати шкоди сервісу (або комп'ютеру), перевантажити мережу, перевантажити центральний процесор або переповнити диск.

Моделі атак. Традиційна модель атаки будується за принципом "один до одного" або "один до багатьох", тобто атака ведеться з одного джерела. Розробники мережевих засобів захисту (міжмережевих екранів, систем виявлення атак тощо) орієнтовані саме на традиційну модель атаки. У різних точках мережі, що захищається, встановлюються агенти (сенсори) системи захисту, які передають інформацію на центральну консоль управління. Це полегшує масштабування системи, забезпечує простоту віддаленого управління і т. д. Однак така модель не може впоратися із розподіленими атаками.

У моделі розподіленої атаки застосовуються інші принципи.

На відміну від традиційної моделі, в розподіленій моделі використовуються відношення «багато до одного» і «багато до багатьох».

Розподілені атаки засновані на «класичних» атаках типу «відмова в обслуговуванні», а точніше на їх підмножині, відомому як Flood- або Storm-атаки (зазначені терміни можна перекласти як «шторм», «повінь» або «лавина»). Суть цих атак полягає в надсиланні великої кількості пакетів на атакований вузол. Таким чином, вузол може вийти з ладу, оскільки він «захлинеться» в лавині надісланих пакетів і не зможе обробляти запити авторизованих користувачів. Однак у разі, якщо пропускна здатність каналу до атакованого вузла перевищує пропускну здатність атакуючого або атакований вузол некоректно налаштований, до «успіху» така атака не призведе. Але розподілена атака відбувається вже не з однієї точки інтернету, а відразу з декількох, що призводить до різкого зростання трафіку і виведення атакованого вузла з ладу.

Набули поширення такі типи атак:

– віддалене проникнення (remote penetration). Атаки, які дозволяють реалізувати віддалене керування комп'ютером через мережу. Наприклад, NetBus або BackOrifice;

– локальне проникнення (local penetration). Атака, яка призводить до отримання несанкціонованого доступу до вузла, на якому вона запущена. Наприклад, GetAdmin;

– віддалена відмова в обслуговуванні (remote denial of service). Атаки, які дозволяють порушити функціонування або перевантажити комп'ютер через інтернет. Наприклад, Teardrop або trin00;

– локальна відмова в обслуговуванні (local denial of service). Атаки, які дозволяють порушити функціонування або перевантажити комп'ютер, на якому вони реалізуються. Прикладом такої атаки є «ворожий» аплет, який навантажує центральний процесор нескінченним циклом, що призводить до неможливості обробки запитів інших додатків.

Існуючі актуальні шкідливі процеси (мережеві аномалії) можна класифікувати за типом джерела чи причини їх виникнення. Приклад такої класифікації наведений на рис. 1.2.

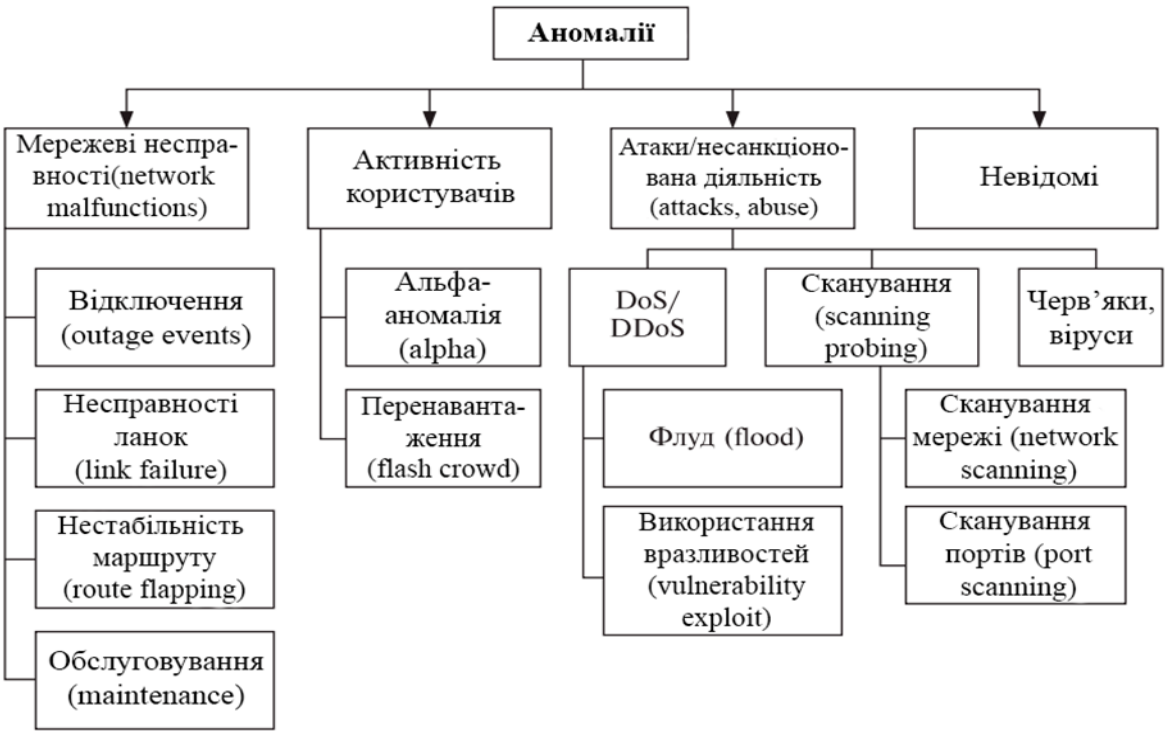

Рис.1.2. Класифікація мережевих аномалій в інформаційній системі організації

У таблиці 1.1 представлені основні типи мережевих аномалій, їх опис та основні характеристики. Наведена систематизація даних про атаки та етапи їх реалізації дає необхідний базис для розуміння технологій виявлення атак.

Таблиця 1.1.

Класифікація аномалій в інформаційній системі організації

Тип аномалії	Опис	Характеристики
Альфа-аномалія	Незвично високий рівень трафіку типу точка-точка	Викид в представленні трафіку байти/с, пакети/с, по одному домінуючому потоку джерело – призначення. Невелика тривалість (до 10 хвилин)
DoS-, DDoS-атака	Розподілена атака типу відмова в обслуговуванні на одну жертву	Викид у представленні трафіку пакети/с, потоки/с, від багатьох джерел до однієї адреси призначення
Перенавантаження	Надзвичайно високий попит на один мережевий ресурс або сервіс	Стрибок у трафіку з потоків/с до однієї домінуючої IP-адреси та домінуючого порту. Зазвичай короткочасна аномалія
Сканування мережі/портів	Сканування мережі за певними відкритими портами або сканування одного хоста по всіх портах з метою пошуку вразливості	Стрибок у трафіку по потоках/с, з декількома пакетами в потоках від однієї домінуючої IP-адреси
Діяльність черв’яків	Шкідлива програма, яка самостійно розповсюджується по мережі та використовує вразливості операційних систем	Викид у трафіку без домінуючої адреси призначення, але завжди з одним або декількома домінуючими портами призначення

Продовження таблиці 1.1.

Класифікація аномалій в інформаційній системі організації

Тип аномалії	Опис	Характеристики
Точка-мульти-точка	Розповсюдження контенту від одного сервера багатьом користувачам	Викид у пакетах, байтах від домінуючого джерела до декількох точок призначення, всі до одного добре відомого порта.
Відключення	Мережеві несправності, які спричиняють падіння в трафіку між однією парою джерело-призначення	Падіння в трафіку по пакетах, потоках і байтах зазвичай до нуля. Може бути довготривалим і включати всі потоки джерело-призначення від або до одного маршрутизатора
Переключення потоку	Незвичайне переключення потоків трафіку з одного вхідного маршрутизатора на інший	Падіння в байтах або пакетах в одному потоці трафіку і викид в іншому. Може зачіпати кілька потоків трафіку

Проблема виявлення шкідливої активності характеризується декількома ключовими аспектами. По-перше, сучасні атаки відзначаються високим рівнем складності та використанням передових технологій обходу традиційних засобів захисту. Зловмисники активно застосовують методи соціальної інженерії, експлуатацію вразливостей нульового дня, поліморфний та метаморфний шкідливий код, що ускладнює їх своєчасне виявлення.

Ця інформація має щорічне відображення в наступних аналітичних звітах [1, 2, 3] , що узагальнюють глобальні тенденції у сфері кібербезпеки: [1, 2, 3]

Verizon Data Breach Investigations Report (DBIR) показав змінену кількість видів інцидентів (рис.1.3) та класифікував типи інцидентів за часом (рис.1.4) [1].

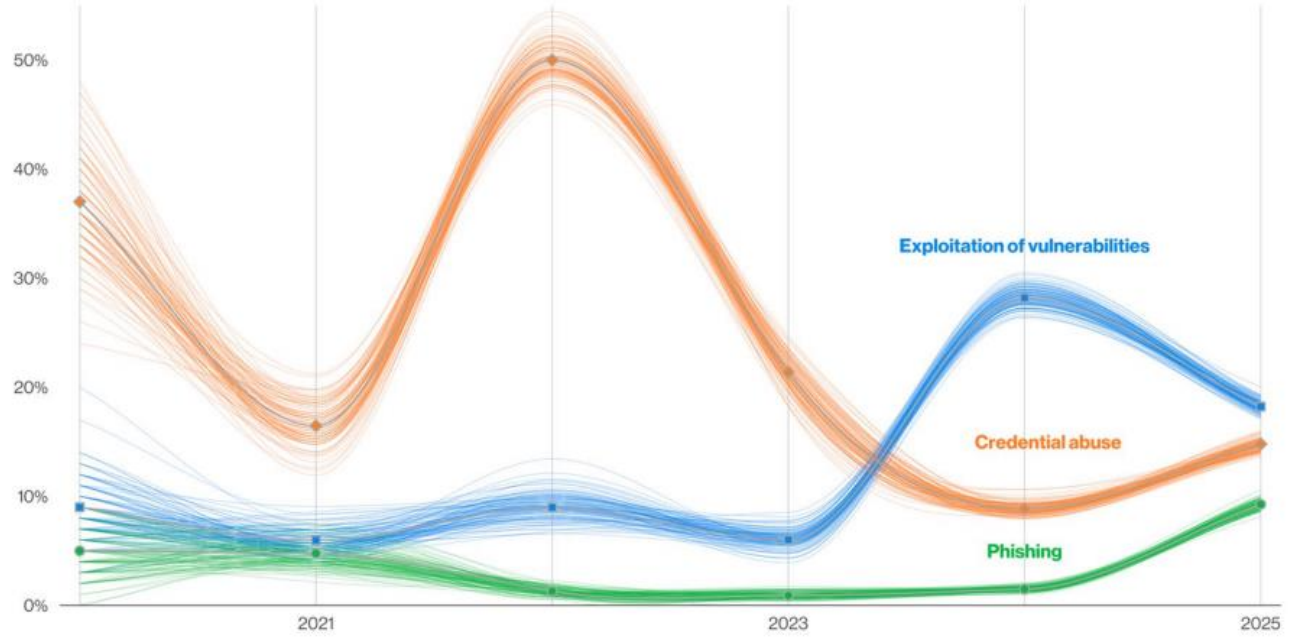

Рис. 1.3: Зміна пропорцій видів іницедентів з часом [1]

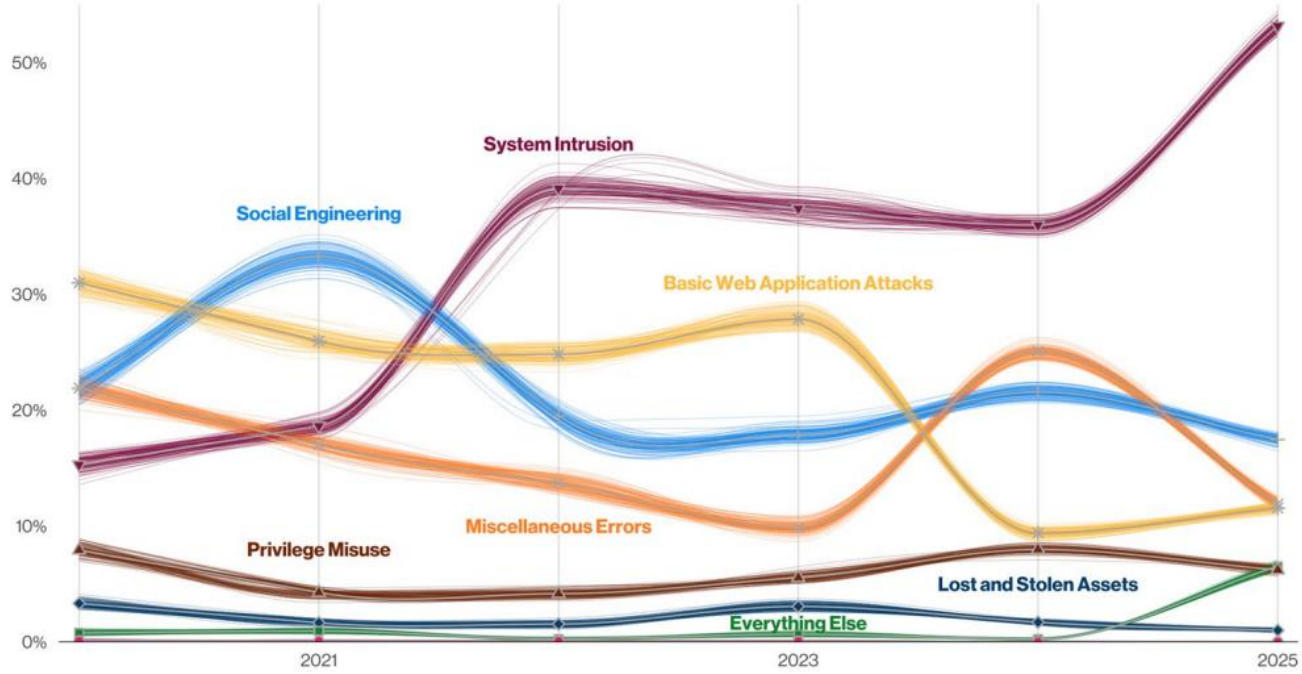

Рис. 1.4: Класифікація типів інцидентів з часом [1]

Звіт [1] охоплює понад 30 000 інцидентів у 94 країнах. Основні висновки:

- 74% інцидентів пов’язані з соціальною інженерією та компрометацією облікових даних;
- значне поширення zero-day експлойтів, зокрема уразливості типу MOVEit;
- зростання APT-кампаній з тривалим латентним проникненням;

– рекомендовано впровадження багатофакторної автентифікації, поведінкового моніторингу та навчання персоналу.

У[2], IBM X-Force Threat Intelligence Index показав найбільш розповсюджені атаки на інформаційні системи (рис.1.5).

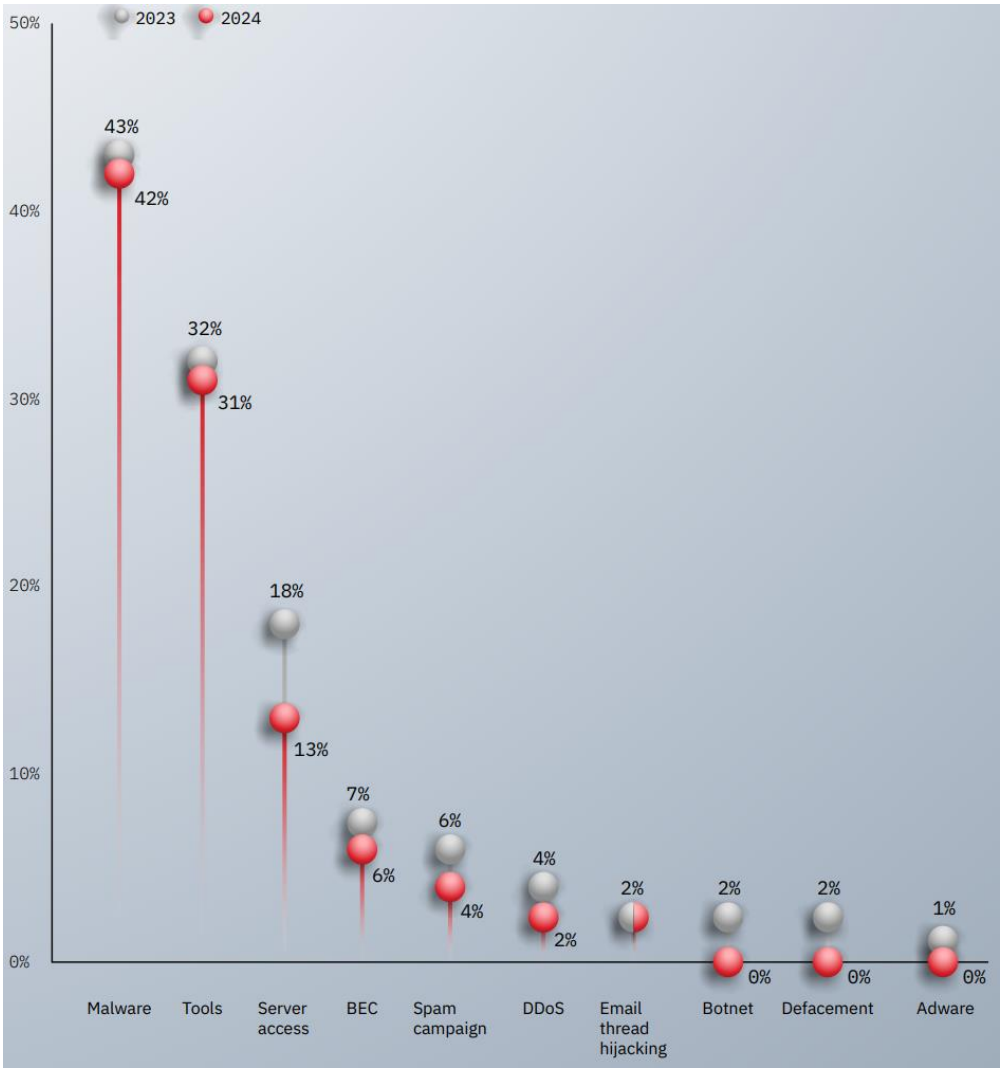

Рис. 1.5: Топ атак у 2024 році в порівнянні з 2023 [2]

Звіт [2] базується на даних з глобальних SOC, honeypots та клієнтських інцидентів та надає таку інформацію:

– 71% зростання атак із використанням валідних облікових записів;
– програми-вимагачі становлять понад 25% усіх інцидентів;
– середній час виявлення атаки – 204 дні, реагування – 73 дні;
– зростає ризик атак на платформи штучного інтелекту;

– рекомендовано впровадження Zero Trust, сегментацію мережі та моніторинг доступу.

У [3], ENISA Threat Landscape Report надає інформацію щодо кількості інцидентів по типу загрози (рис. 1.6).

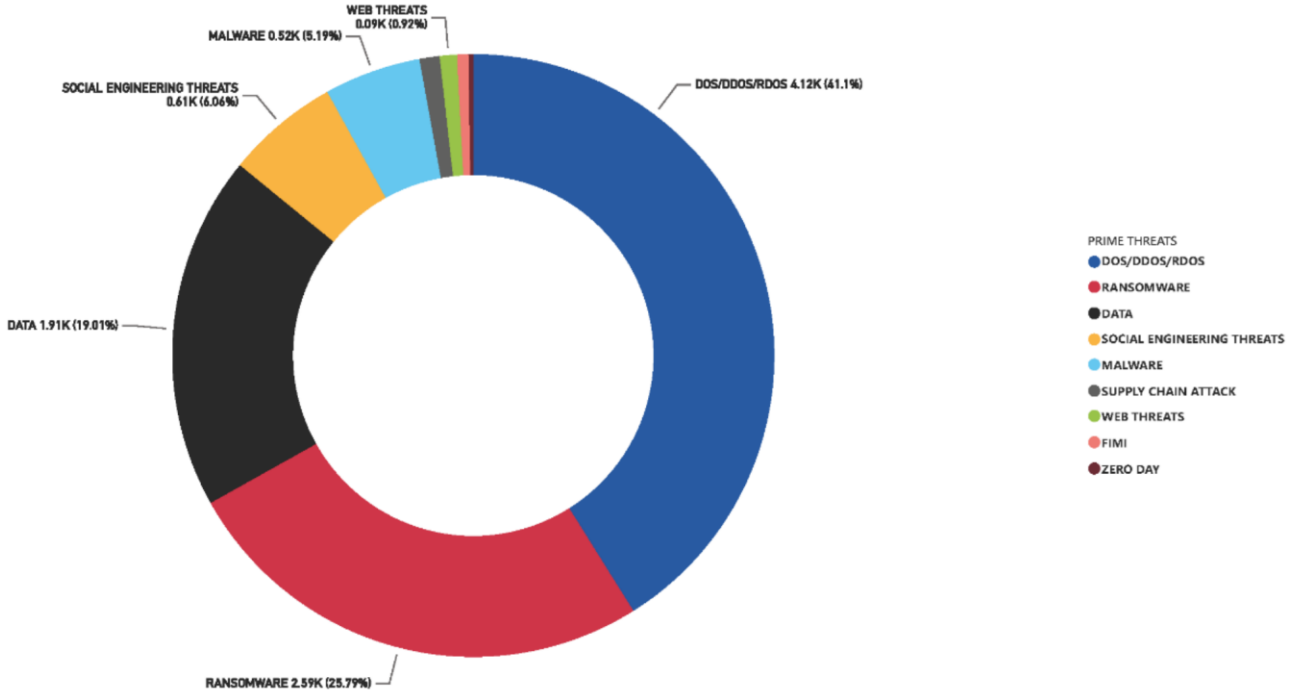

Рис. 1.6. Кількість інцидентів по типу загрози у 2024 р. [3]

Європейський звіт [3] охоплює понад 11 000 інцидентів у країнах ЄС та надає таку інформацію:

– найпоширеніші загрози – атаки на доступність (DoS, DDoS), ransomware та витоки даних;

– зростання атак на критичну інфраструктуру в умовах геополітичної нестабільності;

– рекомендовано впровадження стандартів NIS2, кіберстійкість та обмін інформацією між секторами.

Обсяг мережевого трафіку та кількість подій безпеки в інформаційних системах організацій досягли масштабів, що унеможливлюють ефективний ручний аналіз та виявлення шкідливих процесів. Типова інформаційна система середньої організації генерує мільйони записів в логах щодня, серед яких необхідно ідентифікувати аномалії та потенційно шкідливу активність [5, 9].

Також, існує проблема балансу між точністю виявлення загроз та кількістю хибних спрацювань. Традиційні системи виявлення шкідливої активності часто генерують надмірну кількість хибних спрацівать виявлень, що призводить до втоми аналітиків безпеки та можливого пропуску реальних інцидентів серед шуму хибних тривог. Дослідження показують, що більшість хибних спрацювань у системах виявлення вторгнень (IDS/IPS) складають саме FP – понад 90% від усіх помилкових випадків. Це підтверджує проблему перевантаження аналітиків безпеки шумом, що може призвести до пропуску реальних атак. [4, 15]

Таблиця 1.2.

Порівняння звітів по ключовим показникам

Звіт	**Основні загрози**	**Середній час виявлення**	**Географія**	**Рекомендації**
Verizon DBIR	Соціальна інженерія, облікові дані, APT	~ 200 днів	Глобально	MFA, поведінковий аналіз
SBM X-Force	Валідні облікові записи, Ransomware, AI	204 дні + 73 дні реагування	Глобально	Zero Trust, сегментація
ENISA	DDoS	Не вказано	ЄС	NIS2 кіберстійкість

Тому надалі надамо останні дослідження, які розкривають проблему хибних спрацювань.

Дослідження Cheng-Yuan Ho та ін. (IJFCC) [4] показав: [4]

- за 16 місяців було зібрано понад 2000 випадків хибних спрацювань;
- 92,85% спрацювань – це false positives.

У [121], дослідження IEEE про XAI (Explainable AI - пояснюваний штучний інтелект) для IDS показав: [80, 121]

– традиційні IDS генерують значну кількість хибних спрацювань, що ускладнює роботу аналітиків безпеки;

– використання пояснюваного штучного інтелекту (XAI) пропонується як спосіб зменшення кількості false positives.

У досліджженні [122] Springer (ICDF2C 2024) показав: [122]

– надмірна кількість false positives є однією з головних проблем IDS;

– Необхідно створювати нову модель агрегації та кореляції спрацювань для зменшення шуму та підвищення точності виявлення шкідливих процесів.

Однією з важливих проблем є адаптивність зловмисників. Атакуючі постійно модифікують свої техніки та тактики, що робить сигнатурні методи виявлення неефективними проти нових варіантів атак. Це вимагає розробки систем, здатних виявляти невідомі раніше загрози на основі аналізу поведінки та аномалій.

Також існує проблема пов'язана з багатовекторністю сучасних атак. Такі комплексні кіберзагрози як Advanced Persistent Threat (APT) використовують множинні вектори проникнення, поетапне просування в мережі та довготривалу латентну присутність, що ускладнює виявлення повного ланцюга атаки традиційними методами.

Також критичною є проблема часу реагування. Згідно з дослідженнями, середній час від початку атаки до її виявлення становить десятки днів, протягом яких зловмисники можуть завдати значної шкоди інфраструктурі організації. Це підкреслює необхідність розробки методів автоматизованого виявлення загроз в режимі реального часу або наближеному до нього. Основні дві метрики для цього MTTI та MTTC [7, 9].

MTTI (Mean Time to Identify) – середній час, який потрібен організації, щоб виявити факт інциденту або атаки. Це показник ефективності кіберзахисту, що відображає, скільки часу в середньому проходить від моменту початку атаки до її ідентифікації. Він показує ефективність реагування та здатність організації обмежити поширення атаки. MTTC часто застосовується разом із MTTI у звітах

IBM (Cost of a Data Breach Report) для оцінки загальної швидкості реагування на загрози (рис.1.7. та 1.8) [9].

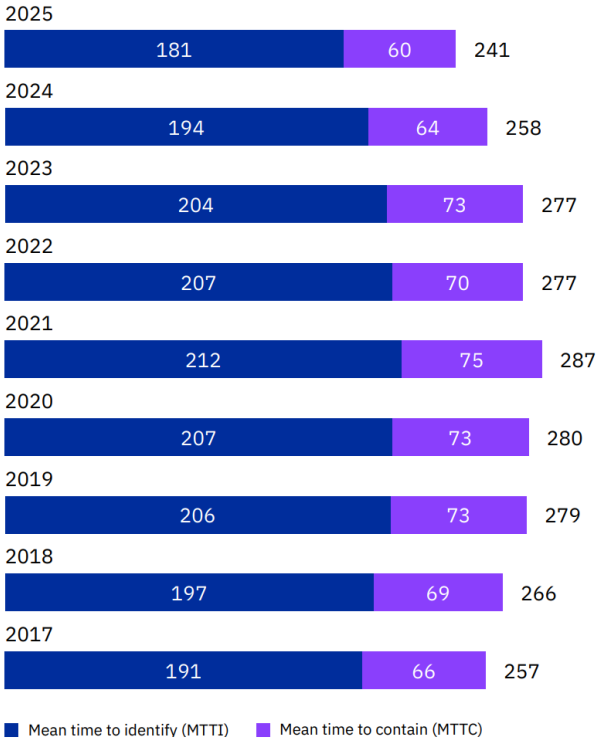

Рис. 1.6. MTTI та MTTC (у днях) згідно з Cost of a Data Breach Report 2025 у 2017-2025 [7].

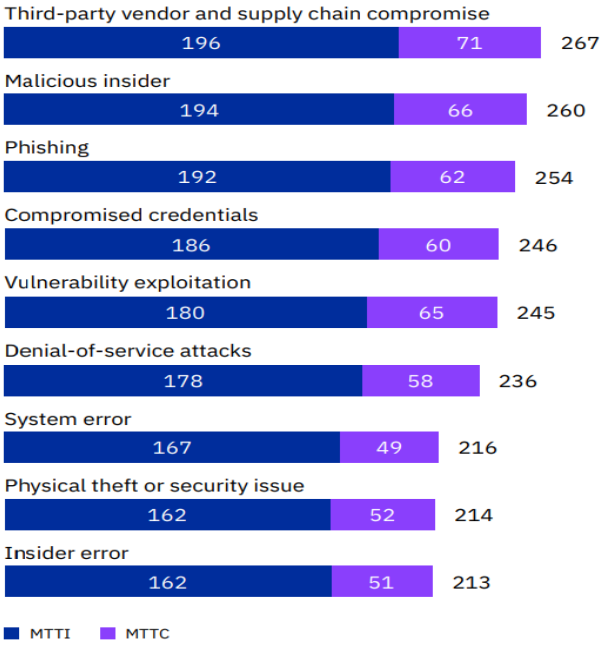

Рис. 1.7. MTTI та MTTC (у днях) за типом загрози згідно з Cost of a Data Breach Report 2025

MTTC (Mean Time to Contain) – середній час, який потрібен організації, щоб стримати інцидент після його виявлення. Це середній час, який потрібен для того, щоб стримати або локалізувати інцидент після його виявлення. Він показує ефективність реагування та здатність організації обмежити поширення атаки [9].

Окремої уваги заслуговує проблема інсайдерських загроз, коли шкідлива активність здійснюється легітимними користувачами системи, що мають відповідні права доступу. Виявлення таких загроз вимагає глибокого аналізу поведінкових патернів та відхилень від нормальної активності користувачів [114].

Нарешті, існує проблема інтеграції та координації різних засобів захисту. Сучасні інформаційні системи використовують множину компонентів безпеки: міжмережеві екрани, антивірусні рішення, системи виявлення вторгнень, системи запобігання витоку даних. Однак відсутність ефективної інтеграції та корреляції даних з цих джерел знижує загальну ефективність виявлення комплексних атак.

Отже, для виявлених проблем (рис.1.8) необхідно розробити можливі варіанти рішення.

Рис. 1.8. Проблеми виявлення швідливої активності

Всі зазначені проблеми обґрунтовують необхідність розробки нових моделей та методів виявлення шкідливої активності, які б поєднували переваги різних підходів, забезпечували високу точність класифікації при мінімізації хибних спрацювань, володіли здатністю до виявлення нових типів загроз та могли ефективно функціонувати в умовах великих обсягів даних (рис.1.8).

1.2. Класифікація та порівняльний аналіз існуючих методів виявлення шкідливої активності

У сучасних інформаційних системах питання забезпечення кібербезпеки набуває критичного значення. Зростання кількості атак, їх складність та багатовекторність вимагають застосування різноманітних методів виявлення шкідливої активності. Жоден окремий підхід не може гарантувати повну захищеність, тому організації комбінують різні техніки для досягнення балансу між точністю, швидкістю реагування та мінімізацією хибних спрацювань.

Методи виявлення шкідливої активності можна класифікувати за різними ознаками: [11, 18]

– за принципом роботи (сигнатурні, аномальні, специфікаційні, поведінкові, гібридні);

– за об’єктом аналізу (мережевий трафік, події на рівні хостів, користувацька поведінка, системні журнали);

– за типом реакції на загрози (пасивне сповіщення, активне блокування, кореляція та автоматизоване реагування) [17, 18].

Для класифікації виявлення методів виявлення шкідливої активності за визначеними категоріями необхідно провести узагальнення категорій за принцим роботи методу та виділити основні переваги та недоліки, які притаманні обраним методам виявлення шкідливої активності.

Розглянемо основні категорії існуючих методів та їх характеристики, які представлено у таблиці 1.3 [14, 15].

Таблиця 1.3.

Узагальнена класифікація методів виявлення шкідливої активності

Категорія	**Принципи роботи**	**Основні переваги**	**Основні недоліки**
Сигнатурні методи	Порівняння активності з базою відомих шаблонів атак	Висока точність для відомих загроз; низький рівень false positives	Не працюють проти нових атак; потребують постійного оновлення
Аномальні методи	Виявлення відхилень від профілю нормальної поведінки	Можуть виявляти нові атаки; не потребують знання про загрози	Високий рівень false positives; складність налаштування
Методи на основі специфікацій	Використання формальних правил допустимої поведінки	Низька кількість хибних спрацювань; здатність ловити невідомі атаки	Трудомістке створення специфікацій; потреба у глибокому знанні системи
Гібридні методи	Комбінація сигнатурного та аномального аналізу	Поєднання переваг різних підходів; адаптивність	Складність архітектури; потреба у значних ресурсах
Поведінковий аналіз	Вивчення послідовностей дій користувачів та процесів	Виявлення багатокрокових атак та інсайдерських загроз	Вимагає складних моделей; ризик помилкової інтерпретації
Кореляція подій	Агрегація та аналіз сигналів з різних джерел	Виявлення комплексних атак; зменшення шуму	Складність налаштування правил; потреба у масштабованій інфраструктурі
Аналіз мережевого трафіку	DPI, NetFlow, DNS-аналіз	Виявлення ботнетів, C2-каналів, ексфільтрації даних	Високе навантаження на систему; потреба у великих обчислювальних ресурсах

Продовження таблиці 1.3.

Узагальнена класифікація методів виявлення шкідливої активності

Категорія	**Принципи роботи**	**Основні переваги**	**Основні недоліки**
Аналіз на рівні хостів	Моніторинг системних викликів, процесів, файлових змін	Ефективність проти руткітів, шкідливого ПЗ	Обмежений контекст; потреба у глибокій інтеграції з ОС
ШІ/ML	Навчання моделей на даних	Виявлення zero-day, зменшення false positives	Потреба у великих даних та ресурсах

Розглянемо основні категорії існуючих методів та їх характеристики [15, 20].

Сигнатурні методи виявлення

Сигнатурні методи базуються на порівнянні спостережуваної активності з заздалегідь визначеними шаблонами відомих атак. Цей підхід широко використовується в системах виявлення вторгнень (Intrusion Detection Systems, IDS) та антивірусних рішеннях. Основною перевагою сигнатурних методів є висока точність виявлення відомих загроз та низька частота хибних спрацювань при правильно складених сигнатурах [11, 15, 16, 18, 116].

Процес виявлення включає порівняння мережевих пакетів, системних викликів або послідовностей файлових операцій з базою сигнатур. Сигнатура може представляти собою байтову послідовність, регулярний вираз, хеш-функцію або складніше правило, що описує характеристики атаки [9].

Однак сигнатурні методи мають суттєві обмеження. Вони неефективні проти нових, невідомих раніше атак (zero-day exploits), вимагають постійного оновлення бази сигнатур, можуть бути обійдені за допомогою обфускації або поліморфного коду. Крім того, зростання кількості сигнатур призводить до зниження продуктивності системи виявлення [11, 18].

Аномальні методи виявлення

Методи виявлення аномалій базуються на припущенні, що шкідлива активність відрізняється від нормальної поведінки системи. Спочатку створюється профіль нормальної активності на основі статистичних характеристик мережевого трафіку, поведінки користувачів або системних процесів. Відхилення від цього профілю класифікуються як потенційні аномалії.

Існують різні підходи до побудови профілів нормальної поведінки: статистичні методи, що використовують середні значення та стандартні відхилення параметрів; методи на основі порогових значень; методи аналізу часових рядів; кластерні методи для групування подібної активності [9, 13, 19].

Перевагою аномальних методів є здатність виявляти нові, невідомі раніше типи атак, що не потребує попереднього знання про конкретні загрози. Однак ці методи схильні до високої частоти хибних спрацювань, оскільки не всі аномалії є шкідливою активністю. Легітимні зміни в поведінці системи або користувачів можуть бути помилково класифіковані як атаки [15, 20].

Методи на основі специфікацій

Ці методи використовують формальні специфікації допустимої поведінки системи або протоколу. Будь-яка активність, що порушує ці специфікації, розглядається як потенційна загроза. Специфікації можуть визначати правила використання системних ресурсів, послідовності системних викликів для легітимних програм, допустимі стани протоколів зв'язку [9].

Методи на основі специфікацій забезпечують низьку частоту хибних спрацювань та здатні виявляти певні класи невідомих атак, які порушують визначені правила. Проте створення повних та точних специфікацій є трудомістким процесом, що вимагає глибокого розуміння системи та можливих сценаріїв її використання [9].

Гібридні методи

Усвідомлення обмежень окремих підходів призвело до розробки гібридних методів, які поєднують переваги різних технік виявлення. Типові гібридні підходи комбінують сигнатурний аналіз для виявлення відомих загроз з аномальним виявленням для ідентифікації нових атак.

Архітектура гібридних систем може передбачати послідовне або паралельне застосування різних методів, використання результатів одного методу для покращення роботи іншого, адаптивне перемикання між методами залежно від контексту та характеристик аналізованої активності [19, 20].

Методи поведінкового аналізу

Поведінковий аналіз фокусується на виявленні підозрілих патернів активності на основі вивчення послідовностей дій користувачів, процесів або мережевих з'єднань. На відміну від простого аномального виявлення, поведінковий аналіз враховує контекст, часові залежності та причинно-наслідкові зв'язки між подіями [15, 20].

Цей підхід особливо ефективний для виявлення комплексних багатоетапних атак, інсайдерських загроз та APT-кампаній, де окремі дії можуть виглядати легітимно, але їх послідовність вказує на зловмисну активність [9].

Методи кореляції подій

Кореляція подій безпеки з різних джерел дозволяє виявляти складні атаки, які не можуть бути ідентифіковані аналізом окремих подій. Системи кореляції агрегують дані з міжмережевих екранів, IDS/IPS, антивірусних систем, систем аутентифікації, мережевого обладнання та інших джерел [9].

Методи кореляції використовують правила, що визначають зв'язки між подіями, статистичні техніки для виявлення значущих паттернів, граф-орієнтовані підходи для моделювання ланцюгів атак, темпоральні логіки для аналізу часових залежностей між подіями [9].

Методи аналізу мережевого трафіку

Спеціалізовані методи аналізу мережевого трафіку включають глибоку інспекцію пакетів (Deep Packet Inspection, DPI), аналіз потоків мережевого трафіку (NetFlow analysis), виявлення аномалій протоколів, аналіз DNS-запитів для виявлення зв'язків з шкідливими доменами [15, 20].

Ці методи дозволяють виявляти різноманітні типи мережевих атак: сканування портів, експлуатацію вразливостей, ботнет-активність, канали управління та контролю зловмисного ПЗ, спроби ексфільтрації даних.

Методи аналізу на рівні хостів

Хост-орієнтовані методи аналізують активність на окремих пристроях: системні виклики та поведінку процесів, зміни в файловій системі та реєстрі, спроби підвищення привілеїв, створення нових процесів та мережевих з'єднань, завантаження бібліотек та модулів [9].

Такий аналіз особливо ефективний для виявлення шкідливого програмного забезпечення, руткітів, програм-вимагачів та інших загроз, що виконуються безпосередньо на пристроях.

Методи на основі штучного інтелекту

Штучний інтелект (ШІ) та машинне навчання (Machine Learning, ML) дедалі активніше застосовуються для виявлення шкідливої активності в інформаційних системах. На відміну від класичних сигнатурних чи аномальних методів, ШІ здатний самостійно навчатися на великих масивах даних, виявляти приховані закономірності та адаптуватися до нових типів атак.

Основні підходи щодо використання методів на основі штучного інтелекту для вирішення задач виявлення шкідливої активності:

– ML використовує алгоритмиів класифікації, кластеризації та регресії для розпізнавання шкідливих патернів у мережевому трафіку чи поведінці користувачів;

– глибинне навчання (Deep Learning, DL) зазвичай використовує нейронні мережі для аналізу складних багатовимірних даних, таких як журнали подій, послідовності системних викликів чи мережеві пакети;

– обробка природної мови (Natural Language Processing, NLP) використовує аналіз текстових логів, повідомлень та команд для виявлення ознак соціальної інженерії чи фішингових атак;

– адаптивні моделі ML використовуються коли є здатність системи оновлювати профілі нормальної поведінки та автоматично підлаштовуватися під нові сценарії атак.

Виділемо переваги використання методів штучного інтелекту:

– виявлення невідомих атак (zero-day) без попередніх сигнатур;

– зменшення кількості false positives завдяки контекстному аналізу;

– можливість реального часу обробки великих обсягів даних;

– адаптивність до нових тактик і технік зловмисників.

Серед недоліків можна виділити наступне:

– потреба у великих обсягах якісних даних для навчання;

– ризик надмірного пристосування моделі до навчальних даних (overfitting);

– висока обчислювальна складність та потреба у потужних ресурсах;

– необхідність пояснюваності моделей (Explainable AI), щоб уникнути «чорної скриньки».

Кожен з розглянутих методів має свої сильні та слабкі сторони (таблиця 1.3). Вибір оптимального підходу залежить від характеристик інформаційної системи, типів найбільш актуальних загроз, доступних ресурсів та вимог до точності виявлення. Сучасні тенденції вказують на доцільність використання гібридних та інтегрованих підходів, що поєднують переваги різних методів для досягнення максимальної ефективності виявлення шкідливої активності [9].

Отже, процес виявлення шкідливої активності в інформаційній системі організації складається з послідовних етапів, кожен з яких має власні завдання та методи реалізації на основі методів машинного. Узгоджена робота цих етапів забезпечує своєчасне виявлення загроз та ефективне реагування на них (рис.1.9).

Основні етапи процесу виявлення шкідливої активності представимо наступним чином:

Етап 1. Збирання даних

На цьому етапі здійснюється акумуляція інформації з різних джерел:

– *Лог-файли операційних систем та додатків* (журнали подій Windows, syslog у Linux, журнали баз даних).

– *Мережеві пакети та трафік* (дані з мережевих сенсорів, IDS/IPS, NetFlow, PCAP).

– *Системні виклики та процеси* (моніторинг API-викликів, файлових операцій, доступу до пам’яті).

Рис. 1.9. Процес виявлення шкідливої активності

– *Додаткові джерела:* дані з антивірусів, міжмережевих екранів, систем управління доступом.

Збирання даних має бути безперервним і масштабованим, щоб охоплювати мільйони подій щодня.

Етап 2. Попередня обробка

Зібрані дані часто містять шум, дублікати та нерелевантну інформацію. Тому застосовуються:

- *нормалізація* – приведення даних до єдиного формату (наприклад, уніфікація часових міток, IP-адрес);
- *фільтрація* – видалення технічних записів, що не несуть інформаційної цінності;
- *агрегація* – об’єднання подій у часові вікна для аналізу поведінки;
- *зниження розмірності* – використання методів PCA або відбору ознак для оптимізації обчислень.

Цей етап критично важливий для підвищення точності подальшої класифікації.

Етап 3. Аналіз ознак

Формуються характеристики, які описують поведінку системи та користувачів:

- *статистичні параметри* – середня кількість пакетів, дисперсія розмірів, швидкість передачі даних.
- *поведінкові ознаки* – частота доступу до ресурсів, послідовність дій користувача, відхилення від звичних патернів.
- *контекстні ознаки* – джерело та призначення трафіку, тип протоколу, географічне розташування вузлів.
- *темпоральні ознаки* – часові інтервали між подіями, циклічність активності.

Формування ознак є ключовим для ефективності алгоритмів машинного навчання.

Етап 4. Класифікація

На цьому етапі застосовуються різні методи аналізу:

- *сигнатурний аналіз* – порівняння з базою відомих шаблонів атак;
- *аномальний аналіз* – виявлення відхилень від нормальної поведінки системи;

– *гібридний аналіз* – комбінація сигнатурного та аномального підходів, часто із застосуванням ансамблевих методів машинного навчання (Random Forest, XGBoost, нейронні мережі).

Класифікація дозволяє визначити, чи є процес або подія шкідливою, та оцінити рівень її загрози.

Етап 5. Реакція

Фінальний етап передбачає дії у відповідь на виявлену загрозу:

– *формування сповіщення* для аналітиків безпеки;

– *автоматичне блокування процесу* або мережевого з'єднання;

– *ізоляція вузла* від корпоративної мережі для запобігання поширенню атаки;

– *кореляція з іншими системами захисту* (SIEM, SOAR) для комплексного реагування.

Ефективність реакції визначається швидкістю виконання та мінімізацією впливу на бізнес-процеси.

Розроблений процес виявлення шкідливої активності в організаціях буде застосовано при розробці методу виявлення шкідливої активності на основі гібридних методів.

1.3. Аналіз методів машинного навчання та гібридних підходів для виявлення шкідливої активності

Методи машинного навчання відкрили нові можливості для виявлення шкідливої активності, забезпечуючи автоматизований аналіз великих обсягів даних, адаптацію до нових типів загроз та покращення точності класифікації. Розглянемо основні категорії методів машинного навчання, що застосовуються в цій області.

Методи навчання з вчителем (Supervised Learning)

Методи навчання з вчителем використовують розмічені датасети, де кожен приклад має відповідну мітку класу (нормальна активність або конкретний тип атаки). Навчена модель потім застосовується для класифікації нових, невідомих раніше зразків [9].

Дерева рішень та випадкові ліси є популярними методами завдяки їх інтерпретованості та здатності обробляти як категоріальні, так і числові ознаки. Випадковий ліс, як ансамбль дерев рішень, забезпечує вищу точність та стійкість до перенавчання. Ці методи ефективно використовуються для класифікації мережевих з'єднань, аналізу логів та виявлення аномалій в поведінці користувачів [21, 66].

Метод опорних векторів (Support Vector Machines, SVM) демонструє високу ефективність в задачах бінарної класифікації, зокрема для розмежування нормальної та шкідливої активності. SVM з різними функціями ядра може моделювати нелінійні залежності в даних. Основним недоліком є складність навчання на великих датасетах та необхідність ретельного підбору гіперпараметрів [115].

Наївний баєсівський класифікатор, базуючись на теоремі Баєса, забезпечує швидку класифікацію та добре працює навіть з обмеженими обсягами навчальних даних. Попри спрощене припущення про незалежність ознак, цей метод показує прийнятні результати в багатьох практичних задачах виявлення вторгнень.

Логістична регресія використовується для ймовірнісної класифікації та оцінки ризиків. Вона забезпечує інтерпретовані результати через коефіцієнти важливості ознак, що цінно для аналізу факторів, які впливають на класифікацію активності як шкідливої.

Нейронні мережі прямого поширення (Multi-Layer Perceptrons) здатні моделювати складні нелінійні залежності між ознаками. Глибокі нейронні мережі з множинними прихованими шарами показують вищу точність, але вимагають значних обчислювальних ресурсів та великих обсягів навчальних даних.

Методи навчання без вчителя (Unsupervised Learning) [22, 67]

Методи без вчителя не вимагають розмічених даних та використовуються переважно для виявлення аномалій і кластеризації подібної активності.

Кластерні алгоритми, такі як K-means, DBSCAN та ієрархічна кластеризація, групують схожі зразки активності. Аномалії ідентифікуються як зразки, що не належать до жодного кластеру або формують дуже малі кластери. DBSCAN особливо ефективний завдяки здатності виявляти кластери довільної форми та автоматично ідентифікувати викиди.

Методи редукції розмірності, включаючи Principal Component Analysis (PCA) та t-SNE, використовуються для зниження розмірності простору ознак та візуалізації даних. Аномалії можуть виявлятися як точки зі значною похибкою реконструкції після проєкції в простір меншої розмірності.

Автоенкодери – це нейронні мережі, що навчаються стискати дані до меншої розмірності та відновлювати їх. Аномальна активність характеризується високою похибкою реконструкції. Варіаційні автоенкодери та деношинг автоенкодери показують особливо хороші результати в задачах виявлення аномалій.

Алгоритми на основі ізоляції (Isolation Forest) спеціально розроблені для виявлення аномалій шляхом ізоляції спостережень в дереві. Аномалії вимагають меншої кількості поділів для ізоляції, що дозволяє ефективно їх ідентифікувати навіть у великих датасетах [33, 69].

Методи напівконтрольованого навчання (Semi-supervised Learning)

Напівконтрольоване навчання комбінує невелику кількість розмічених даних з великим обсягом нерозмічених даних. Цей підхід особливо актуальний для виявлення вторгнень, оскільки отримання повністю розмічених датасетів є трудомістким.

Самонавчання передбачає ітеративне розширення навчальної вибірки шляхом додавання нерозмічених зразків, для яких модель робить впевнені передбачення. Коконтрольоване навчання використовує кілька моделей, що навчаються на різних представленнях даних та взаємно розмічають нерозмічені зразки один для одного.

Генеративні методи, такі як генеративно-змагальні мережі (Generative Adversarial Networks, GAN), можуть використовуватися для генерації синтетичних зразків атак, збільшуючи та балансуючи навчальні датасети.

Методи глибокого навчання (Deep Learning)

Глибокі нейронні мережі демонструють вражаючі результати в складних задачах розпізнавання патернів.

Згорткові нейронні мережі (Convolutional Neural Networks, CNN) ефективно обробляють просторово структуровані дані. В контексті кібербезпеки CNN застосовуються для аналізу мережевого трафіку, представленого у вигляді зображень, аналізу бінарних файлів для виявлення зловмисного ПЗ, виявлення візуальних патернів в логах та метриках системи.

Рекурентні нейронні мережі (Recurrent Neural Networks, RNN) та їх удосконалені варіанти LSTM (Long Short-Term Memory) і GRU (Gated Recurrent Unit) призначені для обробки послідовних даних. Вони ефективні для аналізу часових послідовностей системних викликів, моделювання послідовностей мережевих з'єднань, виявлення аномальних патернів в поведінці користувачів протягом часу, передбачення майбутньої шкідливої активності.

Трансформери та механізми уваги (Attention mechanisms) дозволяють моделі фокусуватися на найбільш релевантних частинах вхідних даних. Ці архітектури показують видатні результати в задачах аналізу послідовностей та можуть ефективно обробляти довгі залежності в даних.

Ансамблеві методи

Ансамблеві підходи комбінують передбачення кількох моделей для покращення точності та стабільності класифікації.

Бутстрап-агрегація (Bagging) навчає множину моделей на різних підвибірках даних та усереднює їх передбачення. Випадковий ліс є прикладом bagging-методу.

Бустинг послідовно навчає моделі, де кожна наступна модель фокусується на зразках, які попередні моделі класифікували неправильно. AdaBoost, Gradient Boosting та XGBoost є популярними алгоритмами бустингу, що демонструють видатні результати в задачах виявлення вторгнень [27, 28].

Стекінг комбінує передбачення різнорідних моделей за допомогою мета-класифікатора, що навчається оптимально поєднувати їх результати.

Методи навчання з підкріпленням (Reinforcement Learning)

Навчання з підкріпленням розглядає виявлення вторгнень як задачу прийняття послідовних рішень, де агент навчається оптимальній стратегії реагування на події безпеки. Q-learning та Deep Q-Networks можуть використовуватися для адаптивного вибору методів виявлення залежно від контексту.

Онлайн-навчання та адаптивні методи

Онлайн-навчання дозволяє моделям адаптуватися до нових типів загроз без повного перенавчання. Інкрементальні версії класичних алгоритмів, адаптивні порогові значення та механізми активного навчання забезпечують динамічне оновлення моделей в умовах еволюції загроз.

Виклики та обмеження

Застосування машинного навчання в задачах виявлення вторгнень стикається з низкою викликів: дисбаланс класів, оскільки шкідлива активність зустрічається значно рідше за нормальну; концептуальний дрейф, коли статистичні властивості даних змінюються з часом; інтерпретованість моделей, особливо глибоких нейронних мереж; вимоги до обчислювальних ресурсів для навчання та інференсу; адверсаріальні атаки, коли зловмисники цілеспрямовано намагаються обійти ML-системи виявлення [34, 35, 36].

Згідно проведеного аналізу, застосування окремих методів машинного навчання (SVM, k-NN, Random Forest, нейронні мережі), жоден із моно-алгоритмів не здатен забезпечити одночасно високу точність виявлення різнорідних атак, низький рівень хибних спрацювань та прийнятну обчислювальну складність для роботи в режимі реального часу [91, 93]. Це явище узгоджується з теоремою «про відсутність безкоштовних обідів» (No Free Lunch Theorem) Д. Волперта, яка стверджує, що універсального алгоритму оптимізації, який перевершував би інші на всіх класах задач, не існує [8]

В умовах сучасної кібербезпеки це означає, що:

Лінійні моделі (наприклад, логістична регресія) ефективні для швидкої фільтрації фонового трафіку, але пропускають складні нелінійні патерни атак [30].

Методи на основі дерев рішень (Decision Trees) та їх ансамблів (Random Forest) добре інтерпретуються, але схильні до перенавчання на зашумлених даних [22, 40].

Глибокі нейронні мережі (DNN) забезпечують високу точність, проте вимагають значних ресурсів та тривалого часу на навчання, що є критичним обмеженням для динамічних систем захисту [8, 33, 38, 77].

Таким чином, виникає протиріччя між стохастичною природою сучасних кібератак та обмеженою узагальнюючою здатністю окремих класифікаторів. Вирішення цього протиріччя лежить у площині гібридизації методів класифікації, а саме – використання гетерогенних (різнорідних) ансамблів.

Серед методів ансамблювання (беггінг, бустінг, стекінг) особливої уваги заслуговує метод стекінгу (Stacking Generalization). На відміну від беггінгу, який зменшує дисперсію помилки шляхом усереднення однотипних моделей, та бустінгу, який зменшує зсув помилки, стекінг дозволяє об'єднати сильні сторони принципово різних математичних апаратів [8, 34, 113, 123].

Ключова перевага стекінгу полягає у використанні дворівневої архітектури:

На першому рівні різнорідні алгоритми генерують гіпотези щодо приналежності трафіку до класу атак. Завдяки своїй різнорідності, ці алгоритми допускають помилки на різних підмножинах даних (їхні помилки є некорельованими) [34, 36].

На другому рівні мета-класифікатор навчається оптимально комбінувати прогнози базових моделей, фактично виправляючи помилки одних алгоритмів за рахунок правильних прогнозів інших [8, 38, 39].

Отже, аналіз показує, що для підвищення точності та надійності систем виявлення шкідливої активності найбільш перспективним є перехід від використання ізольованих моделей до гібридних архітектур на основі стекінгу. Такий підхід дозволяє нівелювати слабкі сторони окремих алгоритмів (зокрема, схильність до хибних спрацювань та перенавчання) та забезпечити адаптивність

системи до нових типів загроз. Це обумовлює необхідність розробки нового методу виявлення шкідливої активності, який би базувався на інтеграції сучасних градієнтних методів та класичних алгоритмів у єдиний стекінг-ансамбль [124].

1.4. Постановка наукового завдання щодо розробки методу виявлення шкідливої активності в інформаційній системі організації на основі гібридної класифікації

На основі проведеного аналізу проблем виявлення шкідливої активності, існуючих методів та підходів з використанням машинного навчання можна сформулювати наукове завдання дослідження.

Актуальність наукового завдання

Аналіз показав, що традиційні методи виявлення вторгнень мають суттєві обмеження: сигнатурні методи неефективні проти нових загроз, аномальні методи генерують велику кількість хибних спрацювань, окремі методи машинного навчання не забезпечують достатньої точності та адаптивності в умовах еволюції кіберзагроз.

Тому виникають наступні протиріччя:

Зовнішні протиріччя полягають між необхідністі виявлення нових, раніше невідомих типів атак (Zero-day, APT) в режимі реального часу та статичною природою баз вирішальних правил (сигнатур) існуючих систем захисту та між експоненційним зростанням обсягів мережевого трафіку в організаціях та обмеженими обчислювальними ресурсами для його глибокого аналізу існуючими засобами [9, 10].

Внутрішні протиріччя між точністю детектування (яка вимагає складних ансамблевих моделей) та швидкістю прийняття рішень (яка вимагає спрощення обчислень для роботи в реальному часі) та між здатністю моделі до узагальнення (detecting variations of attacks) та її стійкістю до "шуму" (false positives) на

нормальному трафіку (рис.1.10). Тому, для вирішення виявленого протиріччя є необхідність реалізації актуального наукового завдання, що полягає у розв’язанні протиріччя між зростаючою складністю, гетерогенністю та обсягами шкідливої активності в сучасних інформаційних системах організацій і обмеженими можливостями існуючих методів виявлення (сигнатурних, евристичних та моно-методів машинного навчання), які не забезпечують необхідного рівня точності класифікації загроз реального часу при допустимому рівні хибних спрацювань.

Мета наукового завдання

Метою дослідження є підвищення точності виявлення шкідливої активності в інформаційній системі організації на основі гібридної класифікації.

Об'єкт дослідження

Об’єктом дослідження дисертаційної роботи є процес виявлення шкідливої активності в інформаційних системах організацій.

Предмет дослідження

Предметом дослідження дисертаційної роботи є методи виявлення шкідливої активності в інформаційних системах організацій.

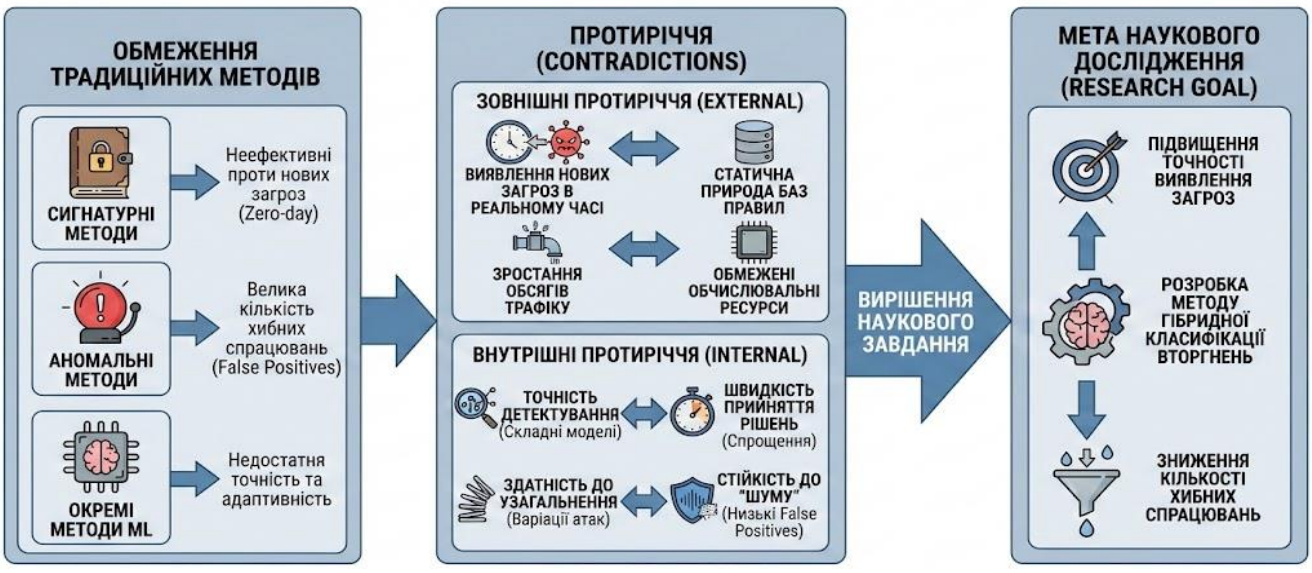

Рис.1.10. Наукове протиріччя щодо виявлення шкідливої активності

Формалізована постановка наукового завдання

Завдання: Розробити метод M виявлення шкідливої активності, який відображає вхідний вектор ознак трафіку X у клас стану системи Y.

Дано:

$X = \{x_1, x_2, \ldots, x_n\}$ – множина вхідних даних (вектори ознак мережевого трафіку: статистичні, темпоральні, контентні) або множина спостережень активності в інформаційній системі, де кожне спостереження $x_i \in \mathbb{R}^d$ представлено вектором з d ознак.

$C = \{c_{norm}, c_{attack}\}$ – множина класів (нормальна активність, атака).

D_{train} – навчальна вибірка, що містить прецеденти [31, 32](x_i, y_i).

Критерій оптимізації цільової функціяї полягає у максимізації інтегральної метрики ефективності (*F1*-score), яка враховує баланс між точністю (*Precision*) та повнотою (*Recall*):

$$F1(M) \to max \tag{1.1}$$

Для досягнення поставленої мети необхідно вирішити наступні наукові завдання:

1. Розробити метод гібридної класифікації на основі стекінг-архітектури, що інтегрує прогнози різнорідних алгоритмів машинного навчання через мета-класифікатор для підвищення точності виявлення шкідливої активності.

2. Розробити комплексний метод оптимізації набору даних для гібридного класифікатора, який поєднує балансування класів (SMOTE), Min-Max нормалізацію та зниження розмірності методом PCA для зменшення обчислювального навантаження без втрати точності класифікації.

3. Удосконалити метод багатокритеріального відбору оптимальних архітектур ансамблевого навчання на основі послідовного застосування стратегії above-average rule та побудови фронту Парето для виявлення шкідливих процесів у реальному часі.

У відповідності до поставленої мети дисертаційної роботи, для вирішення наукового завдання в роботі сформульовані наступні часткові завдання дослідження:

Проаналізувати існуючі методи та підходи до виявлення шкідливої активності в інформаційній системі організації.

Розробити гібридний метод виявлення шкідливої активності, визначити його архітектуру, специфікацію компонентів та обґрунтувати вибір мета-класифікатора.

Розробити метод попередньої обробки даних для методу виявлення шкідливої активності на основі гібридної класифікації в інформаційних системах організацій класифікації з метою зниження обчислювального навантаження та підвищення точності виявлення.

Розробити метод багатокритеріального відбору оптимальних архітектур ансамблевого навчання для виявлення шкідливих процесів у реальному часі.

Провести експериментальне дослідження розробленого методу виявлення шкідливої активності в інформаційній системі організаціїна на основі гібридної класифікації.

Розробити рекомендації щодо створення та впровадження програмного рішення на базі розробленого методу виявлення шкідливої активності в інформаційній системі організації на основі гібридної класифікації.

Критерії оцінки ефективності

Для оцінки ефективності розробленого методу будуть використані наступні метрики:

Точність класифікації (Accuracy) відображає загальну частку правильно класифікованих зразків.

Точність позитивних передбачень (Precision) показує частку зразків, класифікованих як атака, які дійсно є атаками, що критично важливо для мінімізації хибних тривог.

Повнота виявлення (Recall/Detection Rate) вказує на частку реальних атак, які були успішно виявлені системою.

F1-міра є гармонічним середнім точності та повноти, забезпечуючи збалансовану оцінку ефективності.

Area Under ROC Curve (AUC-ROC) характеризує здатність класифікатора розрізняти класи при різних порогових значеннях.

Частота хибних спрацювань (False Positive Rate) є критичним параметром для практичного застосування системи.

Час виявлення атаки відображає швидкість роботи методу та його придатність для використання в режимі реального часу.

Обчислювальна складність характеризує вимоги до апаратних ресурсів для розгортання та експлуатації системи.

Очікувані обмеження та припущення

При розробці методу будуть враховані наступні обмеження та припущення:

Часові обмеження: Час обробки одного екземпляра даних $t(x)$ не повинен перевищувати критичний поріг для роботи в реальному часі T_{max} :

$$t(x) \leq T_{max}$$

Обмеження помилок: Рівень хибних тривог (FPR) не повинен перевищувати допустиму норму α, щоб не блокувати легітимних користувачів:

$$FPR(M) \leq \alpha \quad (1.2)$$

Обмеження ресурсів: Використання CPU та RAM для інференсу моделі повинно бути в межах доступних потужностей типового сервера організації (масштабованість під 1000 співробітників).

Шукане рішення:Гібридна функція класифікації $H(x)$, реалізована через стекінг-ансамбль базових алгоритмів $B = \{b_1, \dots, b_k\}$ (де $b_i \in \{XGBoost, LightGBM, CatBoost, SVM, RF, kNN\}$) та мета-класифікатора $G: Y = G\big(b_1(X), b_2(X), \dots, b_k(X)\big)$.

Припускається наявність репрезентативного датасету для навчання моделі, що містить приклади різних типів атак та нормальної активності.

Метод розробляється для застосування в критичній інфраструктурі та корпоративних інформаційних системах з типовою архітектурою, що включає мережеву інфраструктуру, сервери, робочі станції та засоби захисту.

Передбачається можливість збору даних про мережевий трафік, системні події та активність користувачів без критичного впливу на продуктивність інформаційної системи.

Метод орієнтовано на виявлення широкого спектру типових кіберзагроз, проте можуть існувати специфічні складні атаки, що вимагають додаткових механізмів виявлення.

Ефективність методу може залежати від якості та повноти навчальних даних, налаштування параметрів та адаптації до специфіки конкретної інформаційної системи.

Висновки до розділу 1

У першому розділі проведено комплексний аналіз проблематики виявлення шкідливої активності в інформаційних системах організацій та сформульовано наукове завдання дослідження. На основі виконаного аналізу можна зробити наступні висновки:

1. Проблема виявлення шкідливої активності в інформаційних системах організацій характеризується високим рівнем складності сучасних кіберзагроз, значними обсягами даних, що потребують аналізу, необхідністю забезпечення балансу між точністю виявлення та кількістю хибних спрацювань, адаптивністю зловмисників та постійною еволюцією тактик і технік атак, багатовекторністю сучасних кампаній та необхідністю комплексного аналізу. Традиційні підходи виявляються недостатньо ефективними для протидії актуальним загрозам, що обґрунтовує необхідність розробки нових методів виявлення.

2. Аналіз існуючих методів виявлення шкідливої активності показав, що кожен з основних підходів має як переваги, так і суттєві обмеження. Сигнатурні методи забезпечують високу точність виявлення відомих загроз, але неефективні проти нових атак.. Гібридні підходи виявлення шкідливих процесів поєднують переваги різних методів, демонструють найбільш перспективні результати, однак потребують подальшого вдосконалення для досягнення оптимального балансу між різними характеристиками ефективності.

3. Методи машинного навчання відкрили нові можливості для автоматизованого виявлення шкідливої активності. Навчання з вчителем ефективне для класифікації відомих типів атак, але потребує значних обсягів розмічених даних. Навчання без вчителя дозволяє виявляти аномалії та нові загрози, проте характеризується високою частотою хибних спрацювань. Глибоке навчання демонструє високу точність на складних задачах, але вимагає значних обчислювальних ресурсів та великих датасетів. Ансамблеві методи покращують стабільність та точність класифікації. Найбільш перспективним напрямком є розробка гібридних підходів, що інтегрують переваги різних методів машинного навчання в єдину систему виявлення.

4. На основі проведеного аналізу сформульовано наукове завдання щодо розробки методу виявлення шкідливої активності в інформаційній системі організації на основі гібридної класифікації. Мета дослідження полягає в створенні методу, який забезпечить підвищення точності виявлення загроз при одночасному зниженні кількості хибних спрацювань порівняно з існуючими підходами.

5. Очікувані результати дослідження включають удосконалену систему ознак для виявлення загроз, нову модель гібридної класифікації з механізмами взаємного посилення компонентів, метод виявлення з багаторівневою архітектурою, адаптивний алгоритм онлайн-навчання та методику оцінки ефективності. Наукова новизна полягає в динамічній інтеграції різнорідних алгоритмів з адаптивними ваговими коефіцієнтами, багаторівневій архітектурі класифікації та механізмі активного навчання для адаптації моделі. Практична цінність результатів визначається можливістю їх використання для створення ефективних систем захисту інформаційних систем організацій.

6. Визначено критерії оцінки ефективності та наукові завдання для виконання роботи. Для оцінки розробленого методу будуть використані стандартні метрики класифікації (точність, повнота, F1-міра, AUC-ROC) з особливою увагою до частоти хибних спрацювань та часу виявлення атак. Дослідження передбачає послідовне виконання аналітичного, проектувального, розробницького та експериментального етапів з подальшим аналізом отриманих результатів.

7. Сформульовані в розділі завдання створюють фундамент для подальшого дослідження та розробки методу виявлення шкідливої активності. У наступних розділах буде детально розглянуто теоретичні основи гібридної класифікації, розроблено модель та метод виявлення, проведено їх експериментальне дослідження та аналіз ефективності.

Таким чином, проведений аналіз підтвердив актуальність обраної теми дослідження та обґрунтував необхідність розробки нового методу виявлення шкідливої активності на основі гібридної класифікації з використанням методів машинного навчання. Результати аналізу визначають напрямки подальшої роботи та створюють теоретичну базу для вирішення поставленого наукового завдання.

РОЗДІЛ 2 РОЗРОБКА МЕТОДУ ВИЯВЛЕННЯ ШКІДЛИВОЇ АКТИВНОСТІ В ІНФОРМАЦІЙНІЙ СИСТЕМІ ОРГАНІЗАЦІЇ НА ОСНОВІ ГІБРИДНОЇ КЛАСИФІКАЦІЇ

2.1. Обґрунтування вибору методів машинного навчання для побудови гібридної моделі виявлення шкідливої активності

На основі аналізу, проведеного в першому розділі, необхідно обґрунтувати вибір відповідних методів машинного навчання для побудови гібридної моделі класифікації. Вибір методів повинен базуватися на специфіці завдання виявлення шкідливої активності, характеристиках даних про мережеву та системну активність, вимогах до точності та продуктивності системи (роботи в реальному часі).

Критерії вибору методів машинного навчання

При виборі методів для гібридної моделі класифікації необхідно враховувати наступні критерії:

Точність класифікації є першочерговим критерієм, оскільки помилки в виявленні загроз можуть призвести до серйозних наслідків для організації. Метод повинен забезпечувати високу точність як для відомих типів атак, так і для нових варіантів зловмисної активності. Отже, рівень хибних тривог (FPR) не повинен перевищувати допустиму норму.

Здатність до узагальнення характеризує спроможність гібридної моделі правильно класифікувати нові, раніше не бачені зразки. Це критично важливо в умовах реального часу та постійної еволюції кіберзагроз, коли зловмисники модифікують свої техніки для обходу систем виявлення.

Обчислювальна ефективність визначає можливість використання методу в режимі реального часу або наближеному до нього. Система виявлення повинна обробляти великі обсяги даних з мінімальною затримкою, щоб забезпечити

своєчасне виявлення та реакцію на шкідливу активність (загрози) та не перевищувати критичний поріг для роботи в реальному часі T_{max}.

Стійкість до дисбалансу класів є важливою характеристикою, оскільки в реальних умовах шкідлива активність зустрічається значно рідше за нормальну. Метод повинен ефективно навчатися навіть при значному переважанні одного класу над іншими.

Здатність до роботи з багатовимірними даними необхідна для аналізу множини різнорідних ознак, що характеризують мережеву та системну активність. Типовий датасет містить десятки або сотні параметрів, включаючи числові, категоріальні та послідовні дані.

Інтерпретованість результатів важлива для розуміння логіки прийняття рішень та можливості пояснення алертів фахівцям з безпеки. Це особливо актуально для критичних систем, де кожне рішення про блокування активності повинно бути обґрунтованим.

Адаптивність до змін у даних характеризує здатність методу ефективно функціонувати в умовах концептуального дрейфу, коли статистичні властивості даних змінюються з часом внаслідок еволюції як нормальної діяльності, так і тактик атакуючих.

Стійкість до шуму та аномалій у навчальних даних є важливою, оскільки реальні датасети можуть містити помилки розмітки, неповну або зашумлену інформацію.

Робота метода, як було зазначено, повинна працювати у режимі реального часу, що повинно корелюватися з технічними характеристиками та потужностей серверу організації щодо використання CPU та RAM.

Аналіз методів навчання з вчителем для гібридної моделі

Для компонента гібридної моделі, що базується на навчанні з вчителем, доцільно розглянути наступні методи: Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Gradient Boosting Models (XGBoost, CatBoost, LightGBM.

Support Vector Machine (SVM) - метод опорних векторів демонструє високу ефективність у задачах бінарної та багатокласової класифікації завдяки своїй

здатності знаходити оптимальну гіперплощину, що максимізує margin між класами. У контексті виявлення кіберзагроз, SVM особливо ефективний завдяки кільком ключовим властивостям [21].

По-перше, алгоритм демонструє стійкість у високовимірних просторах ознак, що є критичним для аналізу мережевого трафіку, де кількість ознак може значно перевищувати кількість спостережень. Використання kernel trick дозволяє SVM ефективно працювати з нелінійно роздільними даними, відображаючи вхідний простір у простір вищої розмірності, де класи стають лінійно роздільними. Найпоширенішими ядрами є *RBF* (Radial Basis Function), поліноміальне та сигмоїдне ядро.

Проте, основним недоліком SVM є квадратична або кубічна обчислювальна складність відносно розміру навчальної вибірки, що робить його менш придатним для обробки надвеликих datasets у реальному часі. Час прогнозу залежить від кількості опорних векторів, яка зростає зі складністю задачі. У задачах виявлення мережевих атак, де потрібна швидка реакція на потенційні загрози, це може становити суттєве обмеження.

Додатково, SVM чутливий до вибору гіперпараметрів (*C*, *gamm*а для *RBF* ядра), що вимагає ретельної процедури налаштування через крос-валідацію. Параметр *C* контролює баланс між максимізацією *margin* та мінімізацією помилок класифікації, тоді як *gamma* визначає радіус впливу окремих прикладів навчальної вибірки.

Алгоритм роботи методу

Для лінійно роздільних даних SVM розв'язує наступну задачу оптимізації. Нехай у нас є навчальна вибірка $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, де $x_i \in \mathbb{R}^d$ – вектор ознак, $y_i \in \{-1, +1\}$ – мітка класу. SVM шукає гіперплощину, що максимізує margin між класами.

Задача оптимізації формулюється як:

$$min(w, b)\ ½||w||^2$$

при обмеженнях:

$$y_i(w^T x_i + b) \geq 1, \text{для всіх } i = 1, \ldots, n$$

Тут w – вектор нормалі до гіперплощини, b – зсув (bias term). Величина margin дорівнює 2/||w||, тому мінімізація ||w||² еквівалентна максимізації margin.

Для випадку лінійно нероздільних даних вводяться slack змінні $\xi_i \geq 0$, що дозволяють деяким точкам порушувати margin. Задача стає:

$$min(w, b, \xi)\ ½||w||^2 + C\sum\xi_i$$

при обмеженнях:

$$y_i(w^T x_i + b) \geq 1 - \xi_i, \xi_i \geq 0$$

Параметр C контролює баланс між максимізацією *margin* та мінімізацією помилок класифікації. Великі значення C призводять до жорсткого *margin* (менше помилок на навчальній вибірці, але можливе перенавчання), малі значення C дають м'який *margin* (більше помилок допускається, але краща узагальнююча здатність).

Dual формулювання та *kernel trick*

Через перехід до *dual* формулювання за допомогою множників Лагранжа, задачу можна переформулювати у вигляді, що залежить тільки від скалярних добутків $x_i{}^T x_j$. Це дозволяє застосувати *kernel trick* – замінити скалярний добуток на функцію ядра $K(x_i, x_j)$, що неявно відображає дані у простір вищої (можливо нескінченної) розмірності.

Цільва функція в *dual* формі:

$$f(x) = sign(\sum_{i=1}^{ns} \alpha_i y_i K(x_i, x) + b)$$

де α_i – множники Лагранжа, n_s – кількість опорних векторів (зразків, для яких $\alpha_i > 0$).

Важливо розглянути вибір функції ядра для SVM.

Вибір функції ядра визначає, який простір ознак буде використано для класифікації. Найпоширеніші ядра:

1. RBF (Radial Basis Function) ядро:

Також відоме як *Gaussian kernel*. Це найпопулярніше ядро завдяки своїй здатності обробляти нелінійні залежності будь-якої складності. Воно відображає дані в нескінченновимірний простір:

$$K(x_i, x_j) = exp(-\gamma||x_i - x_j||^2)$$

Параметр γ (*gamma*) контролює радіус впливу кожного навчального зразка:

$$\gamma = 1/(2\sigma^2)$$

Малі значення γ означають великий радіус впливу (широкий *Gaussian*), що призводить до більш гладкої границі рішень. Великі значення γ означають малий радіус впливу, що може призвести до перенавчання.

2. Поліноміальне ядро:

Дозволяє вивчати поліноміальні залежності заданого ступеня d:

$$K(x_i, x_j) = (\gamma x_i^T x_j + r)^d$$

де r – константа (coef0), d – ступінь полінома. При $d=1$ отримуємо лінійне ядро, при $d=2$ – квадратичні ознаки, і т.д.

3. Сигмоїдне ядро:

$$K(x_i, x_j) = tanh(\gamma x_i^T x_j + r)$$

Нагадує сигмоїдну функцію активації в нейронних мережах. Проте, не завжди дає позитивно визначену матрицю Грама, що може призвести до проблем оптимізації.

Визначимо переваги та недоліки SVM.

Переваги:

– Ефективність у високовимірних просторах – SVM добре працює навіть коли кількість ознак d більше кількості зразків n;

– Робастність до перенавчання через максимізацію *margin* та регуляризацію;

– *Kernel trick* дозволяє вивчати складні нелінійні залежності;

– Рішення визначається тільки опорними векторами, що робить модель розрідженою;

– Теоретично обґрунтований метод з гарантіями узагальнення.

Недоліки:

– обчислювальна складність $O(n^2 d)$ до $O(n^3 d)$ робить навчання повільним для великих datasets;

– час прогнозу залежить від кількості опорних векторів, що може бути великою для складних задач;

– чутливість до вибору *kerne*l та гіперпараметрів (C, γ).

– не надає природних ймовірнісних оцінок (потрібна додаткова калібрація)

– складність інтерпретації результатів, особливо при використанні нелінійних ядер.

Random Forest - являє собою ансамблевий метод, що базується на побудові множини дерев рішень з подальшим агрегуванням їхніх прогнозів через механізм голосування. Ця архітектура забезпечує кілька фундаментальних переваг для задач кібербезпеки.

Механізм *bagging* (bootstrap aggregating) у поєднанні з випадковим вибором підмножини ознак на кожному розбитті вузла створює декорельовані дерева, що значно знижує дисперсію моделі та ризик перенавчання. Це особливо важливо при роботі з зашумленими даними мережевого трафіку, де присутні аномалії, викиди та неповні спостереження.

Алгоритм природним чином обробляє як категоріальні, так і числові ознаки без необхідності їх попереднього перетворення. Крім того, *Random Forest* надає

вбудований механізм оцінки важливості ознак через обчислення *Gini impor*tance або *permutation importance*, що дозволяє ідентифікувати найбільш інформативні характеристики трафіку для виявлення атак.

Модель демонструє робастність до викидів та не потребує складної процедури нормалізації даних. Паралельна природа навчання дерев робить алгоритм добре масштабованим на багатоядерних системах. Основними гіперпараметрами для налаштування є кількість дерев, максимальна глибина, мінімальна кількість зразків для розбиття вузла та кількість ознак для розгляду при кожному розбитті.

Алгоритм роботи методу

Random Forest будує B дерев рішень, кожне на окремій bootstrap вибірці даних. Bootstrap вибірка створюється шляхом випадкового вибору n зразків з поверненням з оригінального dataset розміру n. Це означає, що деякі зразки можуть з'явитися кілька разів, а деякі не з'явитися взагалі (в середньому ~63.2% унікальних зразків).

Фінальний прогноз отримується усередненням (для регресії) або голосуванням більшістю (для класифікації): [23]

$$f_r f(x) = 1/B \sum_{\beta=1}^{B} T_\beta(x)$$

де $T_\beta(x)$ – прогноз b-го дерева для входу x.

Особливістю вибору ознак у цьому методі наступний.

На відміну від звичайних дерев рішень, що розглядають всі d ознак при кожному розбитті вузла, Random Forest випадково вибирає підмножину з m ознак. Типово:

– для класифікації: $m \approx \sqrt{d}$

– для регресії: $m \approx d/3$

Цей додатковий рівень рандомізації зменшує кореляцію між деревами в ансамблі, що знижує дисперсію фінальної моделі без значного збільшення *bias*.

Критерії розбиття методу. Для класифікації найчастіше використовується *Gini impurity* – міра неоднорідності вузла:

$$Gini(t) = 1 - \sum_{i=1}^{c} p_i^2(t)$$

де C – кількість класів, $p_i(t)$ – частка зразків класу i у вузлі t. Значення $Gini = 0$ означає чистий вузол (всі зразки одного класу), $Gini = 0.5$ для бінарної задачі означає максимальну невпевненість.

Альтернативно можна використовувати *information gain* на основі ентропії:

$$H(t) = -\sum_{i=1}^{c} p_i(t) log_2(p_i(t))$$

При розбитті вузла обирається ознака та поріг, що максимізують *information gain*:

$$IG = Gini(parent) - \sum child\ (nchild/nparent) \cdot Gini(child)$$

де сума береться по лівому та правому дочірніх вузлах.

Важливість ознак

Random Forest надає вбудований механізм оцінки важливості ознак через *Mean Decrease in Impurity* (MDI):

$$Importance(x_j) = 1/B \sum_{\beta=1}^{B} \sum t \in T_\beta : split\_on\_x_j\ IG_t$$

Ознаки, що частіше використовуються для розбиття та приводять до більшого зменшення *impurity*, отримують вищі оцінки важливості.

Альтернативний метод – *Permutation Importance* – вимірює збільшення помилки моделі при випадковій перестановці значень ознаки:

$$PI(x_j) = 1/B \sum_{\beta=1}^{B} [error\_permuted^{(}_{\beta}) - error\_original^{(}_{\beta})]$$

Out-of-Bag оцінка

Зразки, що не потрапили в *bootstrap* вибірку конкретного дерева (~36.8%), називаються *out-of-bag* (OOB) зразками. Вони можуть використовуватися для неупередженої оцінки помилки моделі без необхідності окремої валідаційної вибірки:

$$OOB\_error = 1/n \sum_{i=1}^{n} \mathbb{1}(\hat{y}_i^{OOB} \neq y_i)$$

де $\hat{y}_i^{OOB}$ – прогноз для i-го зразка, отриманий усередненням тільки тих дерев, для яких цей зразок був OOB.

Переваги та недоліки Random Forest

Переваги:

– висока точність класифікації завдяки ансамблюванню;

– робастність до перенавчання – зазвичай можна використовувати багато дерев без погіршення;

– природна обробка пропущених значень та категоріальних ознак;

– не вимагає нормалізації/стандартизації ознак;

– вбудована оцінка важливості ознак для feature selection;

– ефективна паралелізація – дерева будуються незалежно;

– OOB оцінка дає безкоштовну валідацію.

Недоліки:

– великий розмір моделі – потрібно зберігати всі дерева;

– повільніший inference порівняно з одним деревом; [24, 65]

– може бути bias до категоріальних ознак з багатьма рівнями;

– гірша інтерпретованість порівняно з одним деревом рішень;

– *feature importance* може бути *misleading* для корельованих ознак.

K-Nearest Neighbors (KNN) - KNN є лінивим до навчання (*lazy learning)* алгоритмом, що не будує експліцитної моделі під час навчання, а натомість зберігає

всі навчальні приклади та приймає рішення під час прогнозування на основі локальної структури даних. Для нового спостереження алгоритм знаходить k найближчих сусідів у навчальній вибірці та визначає клас через голосування більшістю.

Основна проблема KNN у контексті виявлення загроз полягає у його обчислювальній складності O(nd) для кожного прогнозу, де n - розмір навчальної вибірки, d - кількість ознак. Це робить алгоритм непрактичним для систем реального часу з великими обсягами даних. Використання оптимізованих структур даних, таких як KD-trees або Ball trees, може пришвидшити пошук, але їхня ефективність знижується у високовимірних просторах через "прокляття розмірності".

KNN критично чутливий до вибору метрики відстані (Euclidean, Manhattan, Minkowski) та масштабу ознак. Ознаки з великими числовими діапазонами домінують у обчисленні відстані, що вимагає обов'язкової стандартизації або нормалізації. Вибір оптимального значення k також є нетривіальною задачею: малі значення призводять до високої дисперсії та чутливості до шуму, великі - до надмірного згладжування границь класів.

У задачах з незбалансованими класами (що типово для виявлення атак, де нормальний трафік значно переважає) KNN демонструє bias до більшого класу. Weighted KNN, що враховує відстань до сусідів, частково вирішує цю проблему.

Алгоритм класифікації полягає у наступному.

Для класифікації нового зразка x алгоритм виконує наступні кроки:

1. Обчислює відстань від x до всіх зразків у навчальній вибірці
2. Знаходить k найближчих сусідів
3. Визначає клас через голосування більшістю серед цих k сусідів

Математично, прогноз формулюється як:

$$\hat{y}(x) = argmax_c \sum i \in N_k(x) \mathbb{1}(y_i = c)$$

де $N_k(x)$ – множина індексів k найближчих сусідів, $\mathbb{1}(\cdot)$ – індикаторна функція, c – клас.

Метрики відстані

Вибір метрики відстані критично впливає на продуктивність KNN. Найпоширеніші метрики:

1. Евклідова відстань (Euclidean distance):

Найінтуїтивніша метрика, що вимірює "пряму" відстань між точками:

$$d(x_i, x_j) = \sqrt{(\sum_{l=1}^{d} (x_{il} - x_{jl})^2)}$$

Чутлива до масштабу ознак – ознаки з великими числовими діапазонами домінують у обчисленні відстані.

2. Манхеттенська відстань (Manhattan/L1 distance):

Також відома як taxicab або city block distance:

$$d(x_i, x_j) = \sum_{l=1}^{d} |x_{il} - x_{jl}|$$

Менш чутлива до викидів порівняно з Euclidean, корисна для високовимірних розріджених даних.

3. Відстань Мінковського (Minkowski distance):

Узагальнення Euclidean та Manhattan метрик:

$$d(x_i, x_j) = (\sum_{l=1}^{d} |x_{il} - x_{jl}|^p)\text{^}(1/p)$$

При $p=1$ отримуємо *Manhattan*, при p=2 – *Euclidean*, при $p \to \infty$ – *Chebyshev* відстань ($max|x_{il} - x_{jl}|$).

4. Косинусна відстань (Cosine distance):

Вимірює кут між векторами, незалежно від їхньої довжини:

$$similarity = (x_i \cdot x_j)/(||x_i|| \cdot ||x_j||)$$

Особливо корисна для текстових даних та коли важлива напрямок, а не величина.

Weighted KNN

Стандартний KNN дає однакову вагу всім *k* сусідам незалежно від відстані. *Weighted* KNN надає більшу вагу ближчим сусідам:

$$\hat{y}(x) = argmaxc \sum i \in N_k(x)\, w_i \cdot \mathbb{1}(y_i = c)$$

Типові схеми зважування:

Обернена відстань:

$$w_i = 1/d(x, x_i)$$

Обернений квадрат відстані:

$$w_i = 1/d(x, x_i)^2$$

Gaussian kernel:

$$w_i = exp(-d(x, x_i)^2/\sigma^2)$$

Вибір *k*

Параметр *k* контролює *bias-variance tradeoff*:

– малі *k* (наприклад, *k=1*): низький *bias*, висока *variance*, чутливість до шуму, складні границі рішень;

– великі *k*: високий *bias*, низька *variance*, згладжені границі, можливий *underfitting*;

– типово обирають непарне *k* для уникнення *ties* у бінарній класифікації;

– емпірично часто використовують $k \approx \sqrt{n}$, де *n* – розмір навчальної вибірки;

– оптимальне *k* визначається через крос-валідацію.

Прискорення пошуку

Naive реалізація KNN має складність *O(nd)* для кожного прогнозу. Для великих datasets використовують структури даних:

– KD-trees: бінарне дерево розбиття простору, ефективне для $d < 20$;

– *Ball trees*: розбиття на гіперсфери, краще для високих розмірностей;

– LSH (Locality-Sensitive Hashing): *approximate nearest nei*ghbors для дуже високих розмірностей.

Переваги та недоліки KNN

Переваги:

– простота реалізації та розуміння;

– не робить припущень про розподіл даних (non-parametric);

– природна multi-class класифікація;

– адаптивність до локальної структури даних;

– може вивчати дуже складні boundaries;

– online learning – легко додавати нові зразки.

Недоліки:

– обчислювальна складність *O(nd*) для кожного прогнозу; [26]

– потребує зберігання всієї навчальної вибірки;

– критична залежність від метрики відстані та масштабу ознак;

– прокляття розмірності – ефективність падає у високовимірних просторах;

– чутливість до незбалансованих класів;

– погана робота з нерелевантними ознаками.

Gradient Boosting Models (XGBoost, CatBoost, LightGBM) - алгоритми градієнтного бустингу представляють сучасний state-of-the-art підхід до табличних даних, послідовно будуючи ансамбль слабких моделей (зазвичай дерев рішень малої глибини) для мінімізації функції втрат через градієнтний спуск у функціональному просторі.

Основна ідея

Gradient Boosting виконує градієнтний спуск у функціональному просторі. Замість оптимізації параметрів (як у нейронних мережах), оптимізується сама функція прогнозування шляхом ітеративного додавання нових моделей.

Ансамбль будується ітеративно:

$$F_0(x) = argmin_\gamma \sum_{i=1}^{n} L(y_i, \gamma)$$
$$F_m(x) = F_{m-1}(x) + \nu \cdot h_m(x)$$

де F_0 – початкова модель (часто константа), h_m – m-ий weak learner, ν – learning rate (швидкість навчання, типово *0.01-0.3*).

Gradient Descent у функціональному просторі

Для мінімізації функції втрат *L(y, F(x))* обчислюються градієнти:

$$g_{im} = [\partial L(y_i, F(x_i))/\partial F(x_i)]F = F_{m-1}$$

Наступний weak learner h_m навчається апроксимувати ці негативні градієнти (pseudo-residuals), ефективно рухаючи модель в напрямку зменшення втрат.

Функція втрат для класифікації

Для бінарної класифікації зазвичай використовується *log loss* (binary cross-entropy):

$$L(y, F(x)) = -[y \cdot log(p) + (1 - y) \cdot log(1 - p)]$$

де ймовірність класу 1 визначається через логістичну функцію:

$$p = \sigma(F(x)) = 1/(1 + e\text{^}(-F(x)))$$

Градієнт *log loss*:

$$g_i = \partial L(y_i, F(x_i))/\partial F(x_i) = p_i - y_i$$

Це різниця між передбаченою ймовірністю та істинною міткою – інтуїтивно зрозумілий residual.

XGBoost (Extreme Gradient Boosting) впроваджує регуляризацію у функцію втрат (L1 та L2), що запобігає перенавчанню. Алгоритм використовує другі похідні (гесіани) для більш точної апроксимації функції втрат, а також parallel tree learning для прискорення навчання. Вбудована обробка пропущених значень та можливість роботи з розрідженими даними роблять його особливо ефективним для реальних datasets. Column block structure для паралельного навчання та cache-aware access patterns забезпечують високу обчислювальну ефективність.

XGBoost покращує базовий gradient boosting через використання другого порядку інформації (Hessian) та регуляризацію.

Гесіан (друга похідна):

$$h_i = \partial^2 L(y_i, F(x_i))/\partial F(x_i)^2 = p_i(1-p_i)$$

Objective function включає регуляризацію:

$$Obj = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

де регуляризаційний член для дерева:

$$\Omega(f) = \gamma T + \lambda/2 \sum_{j=1}^{T} w_j^2$$

T – кількість листків дерева, w_j – ваги листків, γ та λ – гіперпараметри регуляризації.

Оптимальна вага для листка j (використовуючи Taylor expansion):

$$w_j* = -\sum i \in I_j \, g_i \, / \, (\sum i \in I_j \, h_i + \lambda)$$

де I_j – множина індексів зразків у листку j.

Gain від розбиття вузла (використовується для вибору найкращого split):

$$Gain = ½[(\sum i \in I_L\, g_i)^2/(\sum i \in I_L\, h_i + \lambda) + (\sum i \in I_R\, g_i)^2/(\sum i \in I_R\, h_i + \lambda) - (\sum i \in I\, g_i)^2/(\sum i \in I\, h_i + \lambda)] - \gamma$$

де I_L та I_R – множини зразків у лівому та правому дочірніх вузлах.

LightGBM від Microsoft використовує novel leaf-wise tree growth замість традиційного level-wise підходу, що дозволяє будувати глибші та точніші дерева за менший час. Техніка Gradient-based One-Side Sampling (GOSS) зменшує обчислювальну складність, фокусуючись на зразках з великими градієнтами. Exclusive Feature Bundling (EFB) зменшує розмірність через об'єднання взаємовиключних ознак. Це робить LightGBM найшвидшим серед gradient boosting моделей, особливо на великих datasets.

LightGBM від Microsoft впроваджує дві ключові оптимізації:

1. Gradient-based One-Side Sampling (GOSS):

Зразки з малими градієнтами вже добре апроксимовані та менш важливі для навчання. GOSS:

– зберігає всі зразки з великими градієнтами (top a%);

– випадково семплює b% з решти зразків з малими градієнтами;

– застосовує ваговий коефіцієнт (1-a)/b до малих градієнтів для компенсації.

Це зменшує обчислювальну складність при збереженні точності.

2. Exclusive Feature Bundling (EFB):

У розріджених datasets багато ознак ніколи не приймають ненульові значення одночасно (mutually exclusive). EFB об'єднує такі ознаки в bundles, зменшуючи ефективну розмірність без втрати інформації.

3. Leaf-wise tree growth:

На відміну від level-wise росту (який використовує XGBoost), LightGBM росте дерева leaf-wise – розбиває листок з максимальним delta loss. Це призводить до глибших, більш асиметричних дерев, але вищої точності.

CatBoost вирішує fundamental проблеми gradient boosting - target leakage та prediction shift через впровадження ordered boosting та ordered target statistics. Алгоритм має нативну підтримку категоріальних ознак через optimal target statistics, що усуває необхідність ручного кодування. Symmetric trees architecture забезпечує швидке прогнозування та ефективне використання пам'яті. CatBoost особливо ефективний при роботі з datasets, що містять багато категоріальних змінних, типових для логів та системних метрик.

Усі три алгоритми підтримують різні функції втрат, включно з тими, що враховують незбалансованість класів. Вони надають feature importance, часткові залежності та SHAP values для інтерпретації моделей, що критично важливо для систем безпеки, де потрібна пояснюваність рішень.

CatBoost вирішує фундаментальні проблеми gradient boosting:

1. Target Leakage:

При обчисленні статистик для категоріальних ознак використовується той самий зразок, що призводить до overfitting. CatBoost використовує ordered target statistics – для кожного зразка статистики обчислюються тільки на попередніх зразках у випадковій перестановці:

$$\hat{x}_k^{Ts} = \left(\sum_{j=1}^{p-1} \mathbb{1}(x\sigma_{j,k} = x\sigma_{p,k}) \cdot y\sigma_j + a \cdot P\right) / \left(\sum_{j=1}^{p-1} \mathbb{1}(x\sigma_{j,k} = x\sigma_{p,k}) + a\right)$$

де σ – випадкова перестановка, P – prior (середнє значення target), a – параметр згладжування.

2. Prediction Shift:

У стандартному boosting використовується та сама вибірка для навчання дерева та обчислення residuals, що призводить до conditional shift. CatBoost використовує ordered boosting – різні перестановки для різних дерев.

3. Symmetric Trees:

CatBoost будує oblivious decision trees (всі вузли на одному рівні використовують однаковий split condition). Це призводить до:

– швидшого inference (можна використовувати binary indexing);
– кращої регуляризації (менше параметрів);
– ефективнішого використання CPU cache.

Переваги та недоліки Gradient Boosting [40]

Переваги:

– State-of-the-art продуктивність на табличних даних;
– природна обробка різнотипних ознак (числові, категоріальні);
– вбудована feature importance та SHAP values для інтерпретації;
– робастність до викидів та пропущених значень;
– ефективна регуляризація через shrinkage та tree constraints;
– підтримка custom loss functions для специфічних задач.

Недоліки:

– послідовна природа ускладнює паралелізацію (на відміну від Random Forest);
– багато гіперпараметрів для налаштування;
– ризик перенавчання при недостатній регуляризації;
– довше навчання порівняно з Random Forest;
– чутливість до шумних міток у даних.

Extra Trees (Extremely Randomized Trees) - є модифікацією Random Forest з додатковим рівнем рандомізації: замість пошуку оптимального порогу розбиття для кожної ознаки, алгоритм випадково вибирає cutoff point. Це призводить до кількох важливих наслідків.

По-перше, значно зменшується обчислювальна складність навчання, оскільки відсутній пошук оптимального розбиття. По-друге, підвищується рівень декореляції між деревами ансамблю, що теоретично має покращувати

узагальнюючу здатність через зменшення дисперсії. По-третє, збільшується bias окремих дерев, але це компенсується через агрегацію великої кількості дерев.

У контексті виявлення мережевих атак, Extra Trees демонструє меншу чутливість до локальних особливостей навчальної вибірки, що може бути як перевагою (більша робастність до шуму), так і недоліком (потенційна втрата важливих паттернів атак). Алгоритм особливо ефективний, коли потрібна швидка побудова моделі на великих datasets без надмірного налаштування гіперпараметрів.

Random Forest вже використовує два джерела рандомізації:

– Bootstrap sampling – кожне дерево навчається на випадковій підвибірці даних з поверненням;

– Feature bagging – на кожному розбитті розглядається випадкова підмножина з m ознак (типово $m \approx \sqrt{d}$).

Extra Trees додає третє джерело рандомізації, а саме Random split selection – для кожної з m розглянутих ознак випадково обирається поріг розбиття замість пошуку оптимального.

Ансамбль Extra Trees будується аналогічно до Random Forest:

$$f(x) = 1/B \sum_{\beta=1}^{B} T_{\beta}(x)$$

де B – кількість дерев в ансамблі, $T_{\beta}(x)$ – прогноз b-го дерева.

Для класифікації використовується *majority voting*:

$$\hat{y} = argmax c \sum_{\beta=1}^{B} \mathbb{1}(T_{\beta}(x) = c)$$

де c – клас, $\mathbb{1}(\cdot)$ – індикаторна функція.

На відміну від Random Forest, що обов'язково використовує bootstrap sampling, Extra Trees може працювати в двох режимах:

1. З bootstrap sampling (за замовчуванням в scikit-learn): кожне дерево навчається на bootstrap вибірці розміру n (вибір з поверненням).

2. Без bootstrap sampling (оригінальна пропозиція): всі дерева навчаються на повному dataset.

Процедура розбиття вузла

Для кожної з m обраних ознак x_j генерується випадковий поріг:

$$\theta_j \sim Uniform(x_j^{\min}, x_j^{\max})$$

Information gain від розбиття:

$$IG = Gini(parent) - \sum child\ (|child|/|parent|) \cdot Gini(child)$$

Обирається розбиття з максимальним IG.

Проведемо порівняльний аналіз методів машинного навчання, які можна використовувати для виявлення аномальних процесів гіблиднимим методами класифікації. Вона наочно дозволить продемонструвати, чому окремі методи мають недоліки, а які є оптимальним вибором.

Таблиця 2.1.

Порівняльний аналіз методів машинного навчання для виявлення шкідливої активності

Критерій порівняння	Support Vector Machine (SVM)	k-Nearest Neighbors (k-NN)	Random Forest (RF)	Deep Learning (DNN/CNN)	Gradient Boosting (XGBoost/LightGBM)
Точність виявлення відомих атак	Середня	Середня	Висока	Дуже висока	Дуже висока

Продовження таблиці 2.1.

Порівняльний аналіз методів машинного навчання для виявлення шкідливої активності

Критерій порівняння	Support Vector Machine (SVM)	k-Nearest Neighbors (k-NN)	Random Forest (RF)	Deep Learning (DNN/CNN)	Gradient Boosting (XGBoost/LightGBM)
Виявлення нових атак (Zero-day)	Низька (залежить від ядра)	Низька	Середня	Висока	Висока
Швидкість навчання (Training Time)	Низька (повільно на великих даних)	Дуже висока (ліниве навчання)	Середня	Дуже низька (вимагає GPU)	Висока (особливо LightGBM)
Швидкість роботи (Inference Time)	Середня	Дуже низька (залежить від обсягу даних)	Висока	Середня (залежить від архітектури)	Дуже висока
Інтерпретованість (Interpretability)	Низька	Висока	Середня	Дуже низька ("Black box")	Середня (є feature importance)
Стійкість до перенавчання (Overfitting)	Висока (при правильні	Низька	Висока	Низька (вимагає Dropout/Regularization)	Середня (контролюється параметрами)

	й регуляризації)				
Вимоги до пам'яті (RAM)	Високі (квадратична складність)	Дуже високі (зберігає всі дані)	Середні	Високі	Низькі
Робота з незбалансованими даними	Погана	Погана	Добра	Добра	Дуже добра (має вбудовані ваги класів)

Як видно з таблиці 2.1, кожен із розглянутих методів має свої обмеження в контексті задачі виявлення вторгнень у реальному часі:

k-NN є неприйнятним для високонавантажених мереж через критично низьку швидкість класифікації, яка залежить від розміру навчальної вибірки.

SVM демонструє високу обчислювальну складність ($O(N^2)$) при навчанні на великих датасетах, що ускладнює часте оновлення моделі при появі нових загроз.

Deep Learning (DNN), попри найвищу точність, вимагає значних апаратних ресурсів та є «чорною скринькою», що ускладнює інтерпретацію результатів для аналітиків SOC (Security Operations Center).

Random Forest є надійним методом, але часто поступається у точності градієнтним методам на складних гетерогенних даних.

Висновок: Алгоритми градієнтного бустингу (Gradient Boosting), зокрема їх сучасні реалізації LightGBM, XGBoost та CatBoost, демонструють найкращий баланс між точністю детектування, швидкістю роботи та споживанням ресурсів. Вони здатні ефективно працювати з незбалансованими класами, що є критичним для задач кібербезпеки.

Проте, враховуючи, що різні алгоритми бустингу можуть помилятися на різних типах атак, доцільно використати гібридний підхід (Stacking Ensemble). Це дозволить об'єднати прогнози LightGBM (швидкість), XGBoost (точність) та CatBoost (робота з категоріальними ознаками) за допомогою мета-класифікатора, нівелюючи індивідуальні помилки та підвищуючи загальну надійність системи.

Далі проведемо порівняльний аналіз методів машинного навчання для для побудови гібридної моделі, де ансамблеві методи будуть обрані як базові для подальшого стекінгу.

Таблиця 2.2

Порівняльний аналіз методів машинного навчання для задач виявлення шкідливої активності

Критерій порівняння	k-Nearest Neighbors (k-NN)	Support Vector Machine (SVM)	Random Forest (RF)	Gradient Boosting (XGBoost, LightGBM)	Deep Learning (DNN)
Точність класифікації (Accuracy / F1)	Середня	Висока (на малих вибірках)	Висока	Дуже висока	Дуже висока (на надвеликих вибірках)
Швидкість класифікації (Час інференсу)	Дуже низька (лінійно залежить від розміру бази)	Низька (обчислення ядерних функцій)	Висока	Дуже висока (оптимізовані дерева)	Низька (потребує GPU/TPU для *real-time*)
Час навчання моделі	Відсутній (ліниве навчання)	Дуже високий (кубічна складність $O(n^3)$)	Середній (паралелізується)	Середній / Швидкий (LightGBM)	Дуже тривалий
Стійкість до дисбалансу класів (False Positives)	Низька	Середня	Висока	Дуже висока (за рахунок зважування класів)	Висока

Продовження таблиці 2.2

Порівняльний аналіз методів машинного навчання для задач виявлення шкідливої активності

Критерій порівняння	k-Nearest Neighbors (k-NN)	Support Vector Machine (SVM)	Random Forest (RF)	Gradient Boosting (XGBoost, LightGBM)	Deep Learning (DNN)
Інтерпретованість результатів (Пояснненість)	Низька	Низька («чорна скринька» у багатовимірному просторі)	Висока (важливість ознак)	Висока (Feature Importance, SHAP)	Дуже низька («чорна скринька»)
Вимоги до апаратних ресурсів (CPU / RAM)	Високі вимоги до RAM	Високі (матриця Грама)	Середні	Низькі / Середні	Надвисокі (GPU)
Схильність до перенавчання (Overfitting)	Висока (при малому k)	Середня (залежить від параметра регуляризації *C*)	Низька (завдяки беггінгу)	Середня (вимагає налаштування гіперпараметрів)	Висока (без Dropout та регуляризації)

Як видно з наведеного порівняльного аналізу, класичні методи, такі як k-NN та SVM, не задовольняють вимогам сучасних систем виявлення шкідливої активності. Метод k-NN має неприпустимо високий час інференсу на великих обсягах мережевого трафіку, оскільки вимагає обчислення відстаней до всіх об'єктів навчальної вибірки в режимі реального часу. Метод SVM демонструє високу алгоритмічну складність навчання ($O(n^3)$), що робить його масштабування на мільйони мережевих з'єднань практично неможливим без втрати інформативності [9, 16].

Методи глибокого навчання (Deep Learning), хоча і забезпечують найвищу здатність до вилучення складних нелінійних патернів атак, мають суттєві обмеження для впровадження на рівні шлюзів організацій: вони вимагають

наявності спеціалізованих апаратних прискорювачів (GPU) та характеризуються нульовою інтерпретованістю («чорна скринька»), що ускладнює роботу аналітиків SOC (Security Operations Center) при розслідуванні інцидентів.

Найкращий баланс між точністю, швидкістю інференсу (критично для роботи $t(x) \leq T_{max}$ та здатністю обробляти незбалансовані дані демонструють ансамблеві методи на основі дерев рішень – Random Forest (беггінг) та Gradient Boosting (бустинг). Алгоритми сімейства градієнтного бустингу (зокрема XGBoost, LightGBM та CatBoost) дозволяють ефективно працювати з гетерогенними даними та мають вбудовані механізми регуляризації.

Обґрунтування гібридної моделі (Stacking): Незважаючи на високу ефективність градієнтного бустингу, використання єдиного (моно) алгоритму залишає ризик зміщення (Bias) моделі щодо специфічних класів атак (наприклад, алгоритм може ідеально виявляти DDoS, але пропускати атаки типу R2L). Тому доцільним є перехід до гібридної архітектури на основі стекінгу (Stacking Ensemble) [8, 34, 38].

2.2. Метод виявлення шкідливої активності на основі гібридної класифікації

У постановці завдання було сформульовано формалізовано постановку завдання, яке полягає в отриманні цільової функції (формула 1.1). Оптимізаційна задача виконується на максимізації інтегральної метрики ефективності (*F1*-score), яка враховує баланс між точністю (*Precision*) та повнотою (*Recall*).

Сформульоване математичне представлення виявлення шкідливої активності на основі гібридної класифікації буде виглядати наступним чином [8, 38].

Нехай $D = \{(x_i, y_i)_{i=1}^{N}\}$ – множина прецедентів (навчальна вибірка), де:

$x_i, \in \mathbb{R}^n$ – вектор ознак i-го спостереження мережевої активності (де n – кількість ознак).

$y_i \epsilon Y = \{0, 1, \dots, K-1\backslash$ – мітка класу, де $y = 0$ відповідає нормальній активності, а $y \in 1, \dots, K-1)$ – різним типам шкідливої активності (атак).

Базовий рівень класифікації полягає у наступному. Задано множину базових класифікаторів $B = \{b_1, b_2 \dots, b_k)$, де кожен класифікатор b_i є функцією відображення простору ознак у простір ймовірностей класів:

$$b_j : \mathbb{R}^n \rightarrow [0,1]^K \tag{2.1}$$

Для вхідного вектора x, вихід $j-$го базового класифікатора є вектором ймовірностей:

$$P_j(x) = \left[p_{j,0}(x), p_{j,1}(x), \dots, p_{j,K-1}(x)\right]^T \tag{2.2}$$

де $\sum_{k=0}^{K-1} p_{j,k}(x) = 1$.

Для реалізації гібридної моделі (Stacking) формується новий простір мета-ознак Z. Для кожного спостереження x вектор мета-ознак z утворюється конкатенацією прогнозів усіх базових моделей:

$$z = concat(P_1(x), P_2(x), \dots, P_M(x)) \in \mathbb{R}^{M \times K} \tag{2.3}$$

Остаточне рішення приймає мета-класифікатор G який відображає мета-ознаки в остаточний розподіл ймовірностей:

$$P_{final}(x) = G(z) = Softmax(W \cdot z + \beta) \tag{2.4}$$

де:

W – матриця вагових коефіцієнтів мета-класифікатора (навчається).

β – вектор зсуву (bias).

$Softmax(\cdot)$ – функція активації для отримання нормованих ймовірностей.

Вирішальне правило

Клас $\hat{y}$ для спостереження x визначається за правилом максимальної апостеріорної ймовірності:

$$\hat{y} = argmax\, P_{final}(x) \tag{2.5}$$

$$x \in (0, \dots, K-1)$$

Цільова функція (Оптимізація)

Завдання навчання полягає у знаходженні цільової функції з такими параметрами базових моделей та мета-моделі, які максимізують метрику *F*1-*score* на валідаційній множині:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall \rightarrow} \rightarrow max \tag{2.6}$$

При цьому на етапі навчання мета-класифікатора мінімізується функція втрат (Log-Loss або Cross-Entropy):

$$L\left(Y, P_{final}\right) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{K=0}^{K-1} \mathbb{I}\ (y_i = k) \dots ln(P_{inal,k}(x_i\_i))\ \rightarrow min\$\$ \tag{2.7}$$

Запропонована математичний метод гібридної класифікація має дворівневу структуру (Базові моделі→ Мета-ознаки →Мета-модель), де враховано вектори, простори $\mathbb{R}^n$, обмеження суми ймовірностей.

У розробленому методі пропонується використовувати алгоритми градієнтного бустингу (XGBoost, LightGBM, CatBoost) як базові класифікатори (Base Learners) першого рівня для швидкого та точного аналізу трафіку, а їхні прогнози об'єднувати за допомогою мета-класифікатора (Meta-Learner) на основі логістичної регресії (Logistic Regression) або SVM. Такий гібридний підхід дозволить компенсувати слабкі сторони окремих алгоритмів, зменшити загальну дисперсію помилки прогнозування та мінімізувати рівень хибних спрацювань (False Positives) на легітимному трафіку організації.

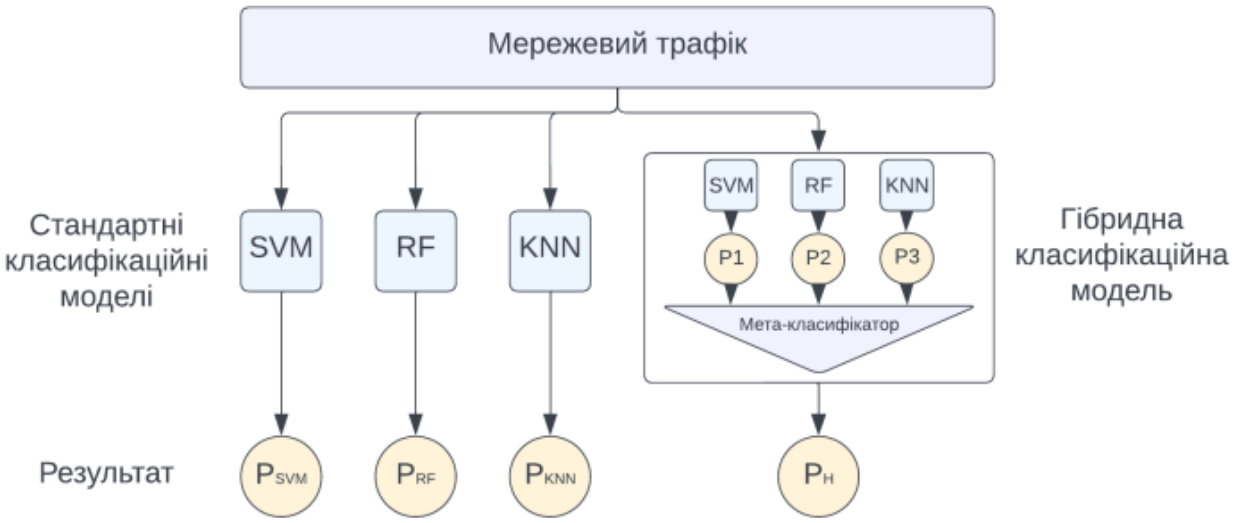

Рис. 2.1. Метод виявлення шкідливої активності на основі гібридної класифікації

Аналіз переваг та обмежень різних методів машинного навчання у попередніх розділах показав, що жоден окремий алгоритм не може оптимально вирішити всі аспекти задачі виявлення шкідливої активності. Гібридна архітектура дозволяє поєднати сильні сторони різних підходів.

Для гібридної класифікації буде використовуватися методика ансамблевого навчання.

Ансамблеве навчання - це метод машинного навчання, що полягає в комбінації кількох моделей для отримання кращих прогнозних результатів. Основна ідея полягає в тому, що ансамбль (або група) моделей, в цілому, дає більш точні і надійні прогнози, ніж окрема модель. Це досягається за рахунок того, що різні моделі можуть ефективно вивчати різні аспекти даних, і, комбінуючи їх, ви можете отримати більш повне і точне розуміння. Ансамблеві методи, такі як беггінг, бустінг та стекінг, використовуються для підвищення стабільності та точності прогнозування.

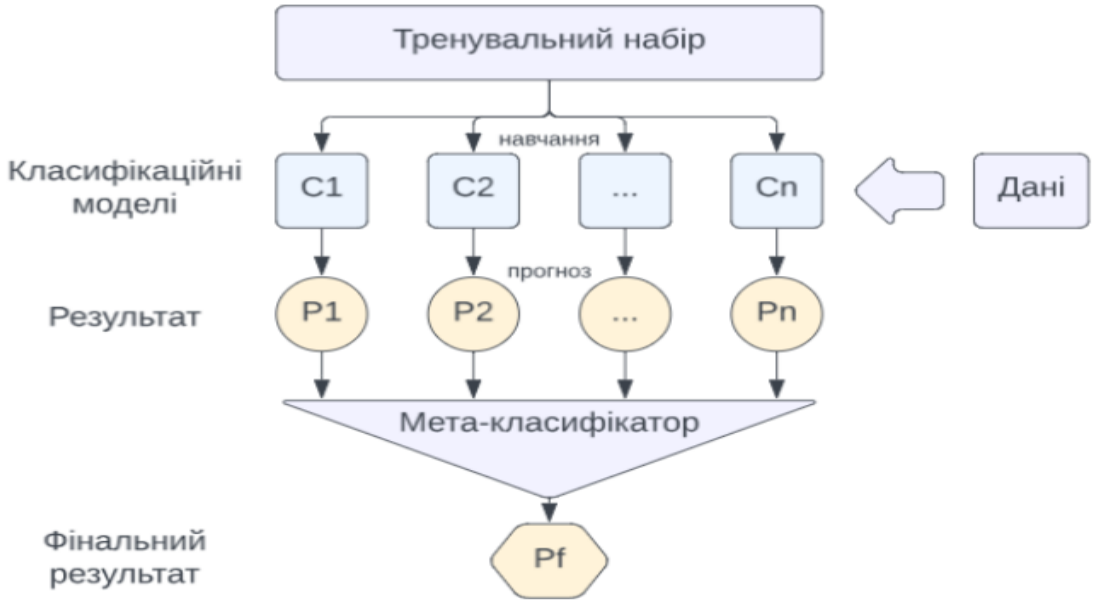

Рис. 2.2. Модульна архітектура методу гібридної класифікації

Мета-класифікатор у гібридній моделі машинного навчання - це модель, яка навчається на виходах інших моделей. Ця модель, також називається метамоделлю, аналізує результати роботи базових моделей (моделей першого рівня) і визначає, як їх комбінувати для отримання кінцевого прогнозу. Це дозволяє виходити за рамки можливостей окремих моделей та підвищує точність передбачень. Гібридна модель дозволяє балансувати між точністю виявлення та частотою хибних спрацювань, налаштовуючи вагові коефіцієнти різних компонентів залежно від пріоритетів безпеки організації.

Остаточний вибір методів для гібридної моделі

На основі проведеного аналізу для гібридної моделі класифікації обрано ансамблеве навчання де будуть використовуватися наступні базові алгоритми: - Метод опорних векторів (Support Vector Machine, SVM): відомий своєю здатністю ефективно розділяти класи навіть у високовимірному просторі за допомогою побудови оптимальної гіперплощини.

- Випадкові ліси (Random Forest): ансамбль дерев рішень, що ефективно працює з табличними даними, стійкий до перенавчання та здатен обробляти велику кількість ознак.

- Метод k-найближчих сусідів (k-Nearest Neighbors, KNN): простий та інтуїтивно зрозумілий алгоритм, що базується на вимірюванні відстані до найближчих елементів, добре зарекомендував себе в задачах класифікації.

- XGBoost (Extreme Gradient Boosting): високоефективна та оптимізована реалізація градієнтного бустингу, відома своєю швидкістю та точністю [37].

- LightGBM: ще одна швидка та пам'яттєво ефективна реалізація градієнтного бустингу, що добре масштабується на великих наборах даних.

- CatBoost: бустингова модель, що має вбудовану підтримку для обробки категоріальних ознак та менш чутлива до вибору гіперпараметрів.

Як мета-класифікатор обрано логістичну регресію, XGBoost, GradientBoost, Random Forest. Вибір мета-класифікаторів у стекінгу обумовлений поєднанням простоти, інтерпретованості та високої продуктивності: логістична регресія ефективно агрегує ймовірнісні прогнози та мінімізує перенавчання, тоді як XGBoost і Gradient Boosting забезпечують потужне моделювання складних залежностей і високу точність, а Random Forest додає стабільність і стійкість за рахунок агрегації багатьох дерев, що разом підвищує точність і надійність системи виявлення шкідливої активності [8]. Комбінування [різних алгоритмів дозволяє компенсувати слабкі сторони кожного окремого методу, мінімізувати ризик перенавчання та, як наслідок, забезпечити суттєве покращення точності та стійкість системи до нових, раніше невідомих типів кібератак.

Така комбінація методів забезпечує баланс між точністю виявлення відомих загроз, обчислювальною ефективністю та стійкістю системи до різноманітних типів шкідливої активності.

Архітектурні принципи інтеграції методів

Обрані методи мають бути інтегровані в єдину архітектуру з урахуванням наступних принципів:

Багаторівнева обробка передбачає послідовне застосування різних методів, де результати попередніх рівнів використовуються як додаткові ознаки для наступних. Це дозволяє виявляти складні багатоетапні атаки.

Ансамблеве агрегування результатів різних класифікаторів через зважену комбінацію їх передбачень або метакласифікатор, що навчається оптимально поєднувати виходи базових моделей.

Адаптивні вагові коефіцієнти для різних компонентів ансамблю, що динамічно налаштовуються на основі поточної ефективності кожного методу, дозволяють системі адаптуватися до змін у характері даних.

Механізм активного навчання для селективного запиту міток експерта у найбільш невизначених випадках забезпечує ефективне оновлення моделі з мінімальними затратами на ручну розмітку.

Модульна архітектура дозволяє гнучко додавати нові методи або замінювати існуючі компоненти без перебудови всієї системи.

Обґрунтований вибір методів та принципів їх інтеграції створює основу для розробки ефективної гібридної моделі класифікації, що описується в наступному підрозділі.

На основі обраних методів машинного навчання розробимо детальну модель гібридної класифікації. Модель включає компоненти попередньої обробки даних, формування простору ознак, базові класифікатори, механізми агрегації рішень та адаптації.

Загальна архітектура моделі

Архітектура гібридної моделі класифікації складається з наступних основних компонентів (рис.2.2):

Модуль збору та попередньої обробки даних забезпечує інтеграцію з джерелами інформації про активність в інформаційній системі, нормалізацію та очищення даних, обробку пропущених значень та викидів.

Модуль формування та вибору ознак створює простір інформативних характеристик для класифікації, включаючи статистичні, поведінкові та контекстні ознаки, виконує відбір найбільш релевантних параметрів.

Модуль базових класифікаторів містить реалізації обраних методів машинного навчання, кожен з яких навчається незалежно та генерує власні передбачення.

Модуль агрегації рішень поєднує результати базових класифікаторів за допомогою зважених схем голосування або метакласифікатора.

Модуль адаптації забезпечує оновлення параметрів моделі на основі нових даних та зворотного зв'язку від аналітиків безпеки [31, 32].

Модуль інтерпретації надає пояснення щодо класифікаційних рішень для фахівців з інформаційної безпеки.

Модуль попередньої обробки даних

Попередня обробка включає наступні етапи:

Нормалізація числових ознак приводить їх до єдиної шкали для забезпечення коректної роботи алгоритмів, чутливих до масштабу даних. Використовується стандартизація:

$$x'_{ij} = (x_{ij} - \mu_j) / \sigma_j \quad (2.8)$$

де μ_j та σ_j – середнє значення та стандартне відхилення j-ї ознаки, обчислені на навчальній вибірці.

Кодування категоріальних ознак перетворює нечислові параметри (наприклад, тип протоколу, флаги з'єднання) у числове представлення. Для бінарних ознак використовується просте кодування 0/1. Для категоріальних ознак з невеликою кількістю значень застосовується *one-hot encoding*. Для високомощних категоріальних змінних використовується *target encoding* або *embedding*.

Обробка пропущених значень вирішує проблему неповних даних. Для числових ознак пропуски заповнюються медіанним значенням або передбачуються за допомогою регресійних моделей. Для категоріальних ознак використовується режим (найчастіше значення) або створюється окрема категорія "невідомо".

Виявлення та обробка викидів застосовує статистичні методи (правило трьох сигм, міжквартильний розмах) або алгоритми виявлення аномалій для ідентифікації екстремальних значень, що можуть бути результатом помилок збору даних. Викиди можуть бути видалені або обмежені пороговими значеннями.

Балансування класів компенсує дисбаланс у навчальних даних через технік и oversampling (SMOTE – Synthetic Minority Over-sampling Technique для генерації синтетичних зразків міноритарного класу), undersampling (зменшення кількості зразків мажоритарного класу), налаштування вагових коефіцієнтів класів у функції втрат алгоритмів навчання.

Модуль формування простору ознак

Ефективність класифікації критично залежить від якості ознак. Модель використовує комплексний набір характеристик:

Статистичні ознаки мережевого трафіку включають тривалість з'єднання, кількість переданих байтів та пакетів у прямому та зворотному напрямках, відношення вхідного до вихідного трафіку, середній розмір пакету, варіацію розмірів пакетів, швидкість передачі даних, міжпакетні інтервали та їх варіацію.

Поведінкові ознаки характеризують патерни активності: кількість з'єднань з одного джерела за часовий інтервал, кількість різних портів призначення, співвідношення успішних та неуспішних з'єднань, послідовність прапорців TCP (SYN, ACK, FIN, RST), патерни доступу до ресурсів системи, частота певних типів системних викликів.

Контекстні ознаки враховують зовнішню інформацію: репутація IP-адреси джерела на основі зовнішніх баз загроз, географічне розташування джерела з'єднання, час доби та день тижня (деякі атаки характерні для певного часу), приналежність до внутрішньої чи зовнішньої мережі, історія попередньої активності джерела.

Ознаки на рівні додатку аналізують специфічні характеристики протоколів: аномалії в HTTP-заголовках або параметрах запитів, підозрілі DNS-запити (DGA-домени, тунелювання даних), характеристики SSL/TLS-з'єднань (версія протоколу, набір шифрів), патерни SQL-запитів для виявлення ін'єкцій.

Темпоральні ознаки відображають часову динаміку: зміна інтенсивності активності у часі, циклічні патерни (денні, тижневі), аномалії в часових послідовностях подій, затримки між пов'язаними подіями.

Відбір інформативних ознак

З початкового великого набору параметрів відбираються найбільш інформативні для класифікації:

Фільтруючі методи оцінюють релевантність кожної ознаки незалежно від алгоритму класифікації. Використовуються статистичні тести (x_i-квадрат для категоріальних ознак, *ANOVA F-test* для числових), кореляційні коефіцієнти для виявлення зв'язку з цільовою змінною, інформаційні критерії (mutual information) для оцінки інформативності.

Вбудовані методи визначають важливість ознак у процесі навчання моделі. Градієнтний бустинг та випадковий ліс природним чином надають оцінки важливості ознак на основі того, наскільки часто та ефективно кожна ознака використовується для розбиття вузлів дерев. L1-регуляризація (Lasso) у лінійних моделях призводить до зануження коефіцієнтів нерелевантних ознак до нуля.

Обгорткові методи оцінюють підмножини ознак на основі продуктивності моделі. Рекурсивне виключення ознак ітеративно видаляє найменш важливі параметри. *Forward/backward selection* послідовно додає або видаляє ознаки для оптимізації метрики якості.

Редукція розмірності методами PCA або t-SNE може застосовуватися для створення нових ознак, що є комбінаціями початкових параметрів та захоплюють максимум варіації даних.

Архітектура базових класифікаторів

Градієнтний бустинг (XGBoost) налаштовується з наступними ключовими параметрами: кількість дерев 100-500 залежно від складності датасету, максимальна глибина дерев 5-10 для балансу між складністю моделі та перенавчанням, швидкість навчання 0.01-0.1 з використанням *early stopping*, ваги класів налаштовуються відповідно до дисбалансу для забезпечення уваги до рідких класів атак.

Функція втрат для багатокласової класифікації: multi:softprob для отримання ймовірностей належності до кожного класу.

Випадковий ліс конфігурується з параметрами: кількість дерев 100-300 для забезпечення стабільності ансамблю, максимальна глибина не обмежена або

обмежена значенням 15-20, кількість ознак для кожного розбиття: $\sqrt{d}$ або $log_2(d)$, де d – загальна кількість ознак, мінімальна кількість зразків для розбиття вузла: 2-10 для контролю складності дерев.

Anomaly score для спостереження x обчислюється на основі середньої довжини шляху до ізоляції в деревах ансамблю:

$$s(x) = 2\text{^}(-E[h(x)] / c(n)),$$

де $E[h(x)]$ – очікувана довжина шляху, $c(n)$ – нормалізуючий коефіцієнт. Значення $s(x)$ близьке до 1 вказує на аномалію, близьке до 0 – на нормальне спостереження.

Автоенкодер має архітектуру: вхідний шар розмірності d (кількість ознак), кодуючі шари з послідовним зменшенням розмірності: $d \to d/2 \to d/4 \to bottleneck$, *bottleneck layer* з розмірністю $d/8 - d/16$, декодуючі шари, симетричні до кодуючих: *bottleneck → d/4 → d/2 → d*, вихідний шар розмірності *d* для реконструкції входу.

Для послідовних даних використовуються LSTM-шари в кодері та декодері.

Функція втрат *MSE* для числових ознак або комбінована функція втрат для змішаних типів даних:

$$L(x, x') = \alpha \cdot MSE(x_{num}, x'_{num,}) + \beta \cdot BCE(x_{cat}, x'_{cat})$$

де x' – реконструйований вектор, α та β – вагові коефіцієнти для числових та категоріальних ознак відповідно.

Аномальність спостереження визначається за величиною похибки реконструкції:

$$anomaly_{score}(x) = ||x - x'||^2$$

Спостереження з похибкою, що перевищує адаптивний поріг, класифікуються як аномальні.

Модуль агрегації рішень

Агрегація передбачень базових класифікаторів реалізується через кілька механізмів:

Зважене голосування використовує статичні або динамічні вагові коефіцієнти. Для статичних ваг коефіцієнти визначаються на основі продуктивності класифікаторів на валідаційній вибірці:

$$w_j = performance_{metric_j} / \Sigma_k performance_{metric_k}$$

де *performance_metric* може бути F1-мірою, точністю або іншою релевантною метрикою.

Динамічні ваги адаптуються на основі характеристик конкретного спостереження. Наприклад, якщо для спостереження x клас ифікатор f_j демонструє високу впевненість (максимальна ймовірність класу значно перевищує інші), його вага збільшується:

$$w_j(x) = confidence_{score_j}(x) / \Sigma_k confidence_{score_k}(x)$$

де *confidence_score* може бути різницею між максимальною та другою за величиною ймовірностями класів або ентропією розподілу ймовірностей.

Метакласифікатор (стекінг) використовує передбачення базових класифікаторів як ознаки для навчання моделі вищого рівня. Вхідний вектор для метакласифікатора формується як конкатенація ймовірностей від базових моделей:

$$z = [p^{(1)}(x), p^{(2)}(x), \ldots, p^{(M)}(x)]$$

Метакласифікатор g: $\mathbb{R}^{MxK} \rightarrow \{0, 1, \ldots, K-1\}$ навчається на навчальній вибірці методом крос-валідації для запобігання перенавчанню. Як метакласифікатор може використовуватися логістична регресія, градієнтний бустинг або невелика нейронна мережа.

Каскадна агрегація передбачає послідовне застосування класифікаторів з різними порогами впевненості. Спочатку застосовуються класифікатори з вчителем (XGBoost, Random Forest). Якщо їх впевненість перевищує високий поріг θ_{high} (наприклад, 0.9), приймається їх рішення. Для спостережень з впевненістю нижче θ_{high} додатково аналізуються результати детекторів аномалій. Якщо класифікатори або автоенкодер вказують на аномалію, спостереження класифікується як потенційна загроза для подальшого аналізу.

Така каскадна схема забезпечує баланс між швидкістю обробки (більшість нормальних спостережень класифікуються швидко на першому етапі) та повнотою виявлення (підозрілі випадки піддаються додатковому аналізу).

Обробка невизначеності та конфліктів

При агрегації можуть виникати ситуації невизначеності або конфлікту між передбаченнями базових класифікаторів:

Невизначеність виникає, коли жоден клас не має достатньо високої ймовірності. В такому випадку обчислюється міра невизначеності:

$$U(x) = 1 - max_k P_k(x),$$

Де

$P_k(x)$ – це ймовірність того, що вхідний об'єкт x належить до класу k;

$max_k P_k(x)$ – це впевненість моделі у своєму «найкращому» прогнозі (найвища ймовірність серед усіх можливих класів);

$U(x)$ – значення невизначеності.

Або через ентропію розподілу:

$$H(x) = -\sum_k P_k(x)\, log\, P_k(x)\,.$$

При високій невизначеності ($U(x) > \theta_{\text{uncertainty}}$ або $H(x) > \theta_{\text{entropy}}$) спостереження позначається для ручного аналізу експертом або піддається додатковій перевірці детекторами аномалій.

Конфлікт виникає, коли різні класифікатори впевнено передбачають різні класи. Для виявлення конфлікту обчислюється міра розбіжності:

$$D(x) = \sum_{j=1}^{M} w_j \cdot KL(p^{(j)}(x) \,||\, P(x))$$

де KL – дивергенція Кульбака-Лейблера між передбаченням j-го класифікатора та агрегованим розподілом.

При високій розбіжності $D(x) > \theta_{\text{conflict}}$ активується механізм експертного аналізу або використовується консервативна стратегія (класифікація як потенційна загроза).

Модуль адаптації моделі

Адаптація забезпечує підтримання ефективності моделі в умовах еволюції загроз та зміни нормальних патернів активності:

Інкрементальне оновлення дозволяє моделі навчатися на нових даних без повного перенавчання. Для градієнтного бустингу використовується *warm start* – додавання нових дерев до існуючого ансамблю. Для випадкового лісу нові дерева додаються до ансамблю, старі видаляються за принципом FIFO при досягненні максимальної кількості. Автоенкодер дотренується на нових нормальних даних із зниженою швидкістю навчання.

Активне навчання мінімізує потребу в ручній розмітці даних, селективно запитуючи мітки експерта. Стратегії вибору зразків для розмітки включають uncertainty sampling (вибір спостережень з найбільшою невизначеністю передбачення), query-by-committee (вибір спостережень, для яких різні класифікатори дають найбільш різні передбачення), diversity sampling (вибір спостережень, що представляють невивчені області простору ознак).

Адаптивні порогові значення автоматично налаштовуються на основі статистики класифікації. Для детекторів аномалій поріг anomaly_score адаптується для підтримання заданого рівня false positive rate. Для класифікаторів з вчителем порогові ймовірності для прийняття рішення налаштовуються на основі зворотного зв'язку від аналітиків безпеки.

Виявлення концептуального дрейфу моніторить зміни в розподілі даних. Використовуються статистичні тести (Kolmogorov-Smirnov test, Page-Hinkley test)

для виявлення значущих змін у розподілі ознак. Моніторинг деградації продуктивності моделі на потоковому вході через ковзне вікно. При виявленні дрейфу активується процес перенавчання моделі на недавніх даних або адаптація параметрів існуючої моделі.

Модуль інтерпретації рішень

Інтерпретованість критично важлива для довіри до системи та можливості розслідування інцидентів:

Важливість ознак для конкретного передбачення обчислюється методами SHAP (SHapley Additive exPlanations) або LIME (Local Interpretable Model-agnostic Explanations). Ці методи пояснюють внесок кожної ознаки в класифікаційне рішення для конкретного спостереження.

Правила класифікації екстрагуються з дерев рішень у випадковому лісі або градієнтному бустингу. Для спостереження, класифікованого як атака, виділяється шлях у найбільш впливових деревах, що визначає умови (набір нерівностей на значення ознак), при яких приймається таке рішення.

Подібні випадки з історії надаються аналітику разом з поточним алертом. Пошук проводиться в базі попередньо класифікованих спостережень на основі мінімальної відстані в просторі ознак або максимальної косинусної подібності.

Візуалізація аномальності для детекторів без вчителя показує, які характеристики спостереження відрізняються від нормального профілю. Для автоенкодера це ознаки з найбільшою похибкою реконструкції. Для Isolation Forest – ознаки, за якими спостереження легко ізолюється.

Оптимізація гіперпараметрів моделі

Гіперпараметри моделі (кількість дерев, глибина, швидкість навчання, вагові коефіцієнти, порогові значення) оптимізуються методами:

Grid Search здійснює вичерпний пошук по сітці заданих значень параметрів.

Random Search вибірково тестує випадкові комбінації параметрів, що часто ефективніше для великих просторів пошуку.

Bayesian Optimization використовує попередні результати для вибору наступних комбінацій параметрів, що обіцяють покращення.

Оптимізація проводиться з використанням крос-валідації на навчальних даних з метою максимізації F1-міри, що балансує точність та повноту виявлення при заданому рівні false positive rate.

Масштабованість та продуктивність

Для забезпечення роботи в режимі реального часу модель оптимізується:

Паралелізація обчислень використовує багатоядерні процесори для одночасного виконання передбачень різними класифікаторами.

Батчева обробка групує множину спостережень для одночасної класифікації, що підвищує ефективність обчислень.

Кешування результатів для ідентичних або близьких спостережень скорочує повторні обчислення.

Пріоритезація обробки віддає перевагу спостереженням з більшою підозрілістю за попередньою швидкою оцінкою.

Розроблений метод виявлення шкідливої активності на основі гібридної гібридної класифікації забезпечує комплексний підхід до виявлення шкідливої активності, поєднуючи переваги різних методів машинного навчання та механізми адаптації до динамічного середовища кіберзагроз.

2.3. Опис набору даних для навчання та тестування моделі гібридної класифікації

Опис набору даних для навчання та тестування моделі гібридної класифікації [56, 57]

Для навчання, валідації та тестування розробленої моделі необхідні якісні датасети, що репрезентативно відображають різноманітні типи нормальної та шкідливої активності в інформаційних системах.

Вимоги до навчальних датасетів

Для забезпечення репрезентативності дослідження обрано датасет CSE-CIC-IDS2018, оскільки він задовольняє вимогам щодо відповідності поточному ландшафту загроз, а не лише історичних даних..

Репрезентативність означає, що дані повинні охоплювати широкий спектр типів нормальної активності (різні легітимні сервіси, протоколи, патерни використання) та різноманітні категорії атак (сканування, експлуатація вразливостей, DoS/DDoS, ботнети, інсайдерські загрози, витік даних).

Збалансованість навчальних датасетів надає достатню кількість зразків кожного класу для ефективного навчання, хоча певний дисбаланс (переважання нормальних зразків) є природним та навіть бажаним для реалістичності.

Якість анотації означає точну та послідовну розмітку даних фахівцями, мінімізацію помилок у мітках класів.

Повнота ознак вимагає наявності всіх необхідних атрибутів мережевих з'єднань, системних подій для формування інформативного простору ознак.

Масштаб датасету повинен бути достатнім для навчання складних моделей – від сотень тисяч до мільйонів зразків залежно від складності задачі.

Для проведення експериментального дослідження було обрано датасет CSE-CIC-IDS2018, розроблений спільно Communications Security Establishment (CSE) та Canadian Institute for Cybersecurity (CIC) при Університеті Нью-Брансвік (Канада) у 2018 році [57]. Вибір цього датасету обґрунтовується кількома критичними факторами, що визначають його придатність для дослідження сучасних методів виявлення мережевих аномалій.

По-перше, датасет відображає актуальний ландшафт кіберзагроз станом на 2018 рік, включаючи найбільш поширені типи атак та експлойти, що активно використовувалися зловмисниками у цей період. По-друге, на відміну від застарілих датасетів (DARPA 1999, KDD Cup 1999, NSL-KDD), CSE-CIC-IDS2018 містить реалістичний мережевий трафік, згенерований у контрольованому, але максимально наближеному до реального корпоративного середовищі. По-третє, датасет надає повний набір екстрагованих ознак мережевих потоків (flows), що

дозволяє безпосередньо застосовувати алгоритми машинного навчання без необхідності додаткової обробки raw traffic captures.

CSE-CIC-IDS2018 є еволюційним продовженням попередніх робіт дослідницької групи CIC, зокрема датасету CICIDS2017, та усуває ряд недоліків, притаманних існуючим публічним датасетам: відсутність різноманітності атак, нереалістичні паттерни трафіку, застарілі типи атак, недостатня документація методології збору даних [41].

Розподіл датасетів на навчальну, валідаційну та тестову вибірки

Для коректної оцінки моделі дані розділяються:

Навчальна вибірка (60-70% даних) використовується для навчання базових класифікаторів, підбору параметрів моделей.

Валідаційна вибірка (15-20% даних) використовується для налаштування гіперпараметрів, вибору оптимальних порогових значень, раннього зупинення навчання для запобігання перенавчанню, навчання метакласифікатора в стекінгу.

Тестова вибірка (15-20% даних) використовується виключно для остаточної оцінки продуктивності моделі, не використовується під час навчання та налаштування [45, 49].

Розподіл виконується з урахуванням стратифікації для збереження пропорцій класів у кожній вибірці та часової послідовності (тестові дані хронологічно пізніші за навчальні для реалістичної оцінки).

Попередня обробка та нормалізація датасетів

Перед використанням датасетів для навчання виконується:

Видалення дублікатів для уникнення завищення оцінок продуктивності.

Обробка пропущених значень шляхом імпутації або видалення зразків з надто великою кількістю пропусків.

Нормалізація числових ознак для приведення до спільної шкали. Параметри нормалізації (середнє, стандартне відхилення) обчислюються лише на навчальній вибірці та застосовуються до валідаційної та тестової. Нормалізація (застосовувалася для чутливих до масштабу алгоритмів (SVM, kNN, LogReg), тоді

як для XGBoost та RF використовувалися вихідні дані для збереження інтерпретованості порогів розбиття.

Кодування категоріальних змінних з використанням одного з методів залежно від кількості категорій та їх семантики.

Балансування класів у навчальній вибірці проводилось через SMOTE.

Метрики та характеристики датасетів

1. Статистичні характеристики розподілу кожної ознаки (середнє, медіана, квартилі, стандартне відхилення) [47, 48].
2. Кореляція між ознаками для виявлення надлишкових або дублюючих параметрів.
3. Розподіл класів та ступінь дисбалансу.
4. Наявність викидів та аномалій у даних.
5. Темпоральні характеристики (сезонність, тренди) для часових рядів.

Верифікація якості датасетів

Якість датасетів перевіряється через експертний аналіз випадкової вибірки розмічених зразків фахівцями з безпеки, крос-валідацію якості розмітки шляхом повторного аналізу підмножини даних різними експертами, порівняння розподілів ознак у різних підвибірках для перевірки однорідності, аналіз типових помилок класифікації простих моделей для виявлення потенційних проблем у даних.

Забезпечення репрезентативності

Для підвищення репрезентативності датасетів використовується інтеграція даних з множини джерел (публічні датасети + власні дані), доповнення рідких класів атак синтетичними зразками, періодичне оновлення датасетів новими зразками для відображення еволюції загроз.

Етичні та юридичні аспекти

При підготовці власного датасету дотримуються вимоги законодавства про захист персональних даних (GDPR, місцеві регуляції), отримано відповідні дозволи на збір та обробку даних, виконана анонімізація всієї чутливої інформації, забезпечена безпека зберігання датасетів з обмеженням доступу [58, 59].

Описані датасети забезпечують надійну основу для навчання та всебічного тестування розробленої гібридної моделі класифікації, дозволяючи оцінити її ефективність як на стандартних бенчмарках, так і в реальних умовах роботи інформаційних систем організацій.

Висновки до розділу 2

У другому розділі вирішено завдання розробки методу виявлення шкідливої активності в інформаційній системі організації на основі гібридної класифікації. Основні здобутки та результати розділу полягають у наступному:

1. *Вперше* розроблено метод виявлення шкідливої активності в інформаційній системі організації на основі гібридної класифікації, який базується на використанні гетерогенного набору базових класифікаторів та мета-класифікаторі на основі XGBoost для дворівневої стекінг-архітектури, що дає можливість застосовувати різнорідні ансамблеві методи машинного навчання у системах виявлення шкідливої активності в режимі реального часу з можливістю динамічного переналаштування класифікаторів.

2. Формалізовано математичне представлення наукової задачі виявлення шкідливої активності як багатокритеріальну оптимізаційну задачу. Визначено цільову функцію максимізації F1-міри при жорстких обмеженнях для роботи в режимі реального часу та рівеня хибних тривог (FPR). Математичний опис включає процедуру трансформації простору ознак у простір мета-ознак, що створює теоретичне підґрунтя для програмної реалізації методу.

3. Підготовлено та верифіковано репрезентативну базу даних для експериментальних досліджень. Сформовано комбінований датасет, що включає еталонний набір CSE-CIC-IDS2018 та власний унікальний датасет (обсягом 1,5 млн зразків), зібраний з реальної інформаційної системи організації. Валідація на реальних даних дозволяє перевірити стійкість моделі до оцінки її ефективності в умовах реального трафіку.

РОЗДІЛ 3 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ МЕТОДУ ВИЯВЛЕННЯ ШКІДЛИВОЇ АКТИВНОСТІ

3.1. Комплексний метод оптимізації набору даних CSE-CIC-IDS2018 для навчання моделі гібридної класифікації

Ефективність гібридного методу виявлення шкідливої активності, описаного у попередніх підрозділах, значною мірою визначається якістю навчальної вибірки. Навіть найдосконаліша архітектура ансамблевого класифікатора не здатна компенсувати системні дефекти вхідних даних: дисбаланс класів, надлишкову розмірність ознакового простору та гетерогенність числових шкал. У зв'язку з цим у межах дисертаційної роботи розроблено комплексну методику підготовки датасету *CSE-CIC-IDS2018*, яка передбачає три послідовні кроки: балансування класів засобами алгоритму SMOTE, масштабування ознак методом Min-Max нормалізації та зниження розмірності методом головних компонент (PCA). Нижче наведено детальний аналіз виявлених проблем і опис кожного з реалізованих рішень [56, 57, 60, 117, 125].

Результати аналізу вихідного набору даних та ідентифікація системних дефектів (проблем).

CSE-CIC-IDS2018, незважаючи на широке використання у дослідженнях з кібербезпеки як еталонний бенчмарк, містить низку структурних недоліків, що унеможливлюють його безпосереднє застосування для навчання надійних систем виявлення вторгнень. За результатами проведеного аудиту виявлено чотири класи дефектів.

Дефект 1. Екстремальна нерівномірність розподілу класів.

Найбільш деструктивною перешкодою для навчання багаторівневої архітектури стекінгу є аномальний дисбаланс між легітимним трафіком та векторами атак. У підмножинах датасету частка класу *Benign* сягає 85–95%, тоді як

окремі типи атак (наприклад, *Infiltration* або *Botnet*) представлені лише кількома сотнями записів на тлі мільйонів рядків фонового трафіку. В умовах такого розподілу базові алгоритми першого рівня демонструють формально високу *Accuracy*, однак фактично ігнорують міноритарні класи атак, оскільки помилки на них майже не впливають на загальну функцію втрат. Система виявлення фактично перетворюється на тривіальний класифікатор, що прогнозує виключно нормальну активність.

Дефект 2. Висока розмірність та мультиколінеарність ознакового простору.

Датасет містить понад 80 ознак для кожного мережевого потоку. Проведений кореляційний аналіз виявив сильні лінійні залежності між групами параметрів. Зокрема, ознаки *Fwd Packet Length Max, Fwd Packet Length Mean* та *Fwd Packet Length Std* дублюють інформацію про розподіл довжини пакетів у прямому напрямку (коефіцієнт кореляції між ними наближається до 0,98). Аналогічна ситуація спостерігається для часових характеристик міжаррівальних інтервалів (IAT). Присутність корельованих ознак призводить до «прокляття розмірності», суттєво збільшує час навчання мета-класифікатора та підвищує ризик перенавчання. Для систем, що працюють у режимі реального часу з бюджетом затримки менше 7,16 мс, обробка 80-ти ознак є неприйнятною.

Дефект 3. Наявність технічних артефактів і «отруєних» даних.

Під час аналізу структури датасету виявлено значну кількість некоректних записів: нескінченні значення (*Infinity*) та порожні значення (*NaN*) у полях швидкості передачі пакетів (*Flow Packets/s*), дублювання рядків, а також записи з від'ємними значеннями тривалості потоку, що виникають через помилки логування під час генерації датасету. Безпосереднє використання таких даних дестабілізує процес градієнтного навчання та унеможливлює коректне застосування методів масштабування.

Дефект 4. Гетерогенність числових шкал.

Ознаки датасету представлені в кардинально різних одиницях вимірювання: кількість байтів може сягати мільярдів, тоді як TCP-прапорці (SYN, ACK, PSH) є

бінарними значеннями. Такий масштабний розрив є критичною проблемою для алгоритмів, що використовують евклідову відстань або матрицю ваг (SVM, логістична регресія у ролі мета-класифікатора). Ознаки з великими числовими значеннями домінують у функції класифікації, знецінюючи вплив бінарних параметрів, які нерідко є більш діагностично значущими для ідентифікації атак типу Port Scanning або DoS.

Таким чином, вихідний датасет *CSE-CIC-IDS2018* у первинному вигляді є малопридатним для навчання надійних гібридних систем кібербезпеки. Виявлені дефекти стали визначальним стимулом для розробки комплексного методу попередньої обробки, який складається з таких етепів: [41, 43]

Етап 1. Усунення дисбалансу класів за допомогою алгоритму SMOTE

Для вирішення проблеми дисбалансу класів застосовано алгоритм *SMOTE (Synthetic Minority Over-sampling Technique)*. На відміну від примітивних методів випадкового дублювання (*random oversampling*), що призводять до перенавчання моделі, SMOTE реалізує стратегію синтетичної генерації нових зразків у векторному просторі ознак. Це дозволяє суттєво розширити межі розподілу міноритарних класів без механічного копіювання існуючих записів.

Алгоритм функціонує таким чином: для кожного об'єкта x_i міноритарного класу визначається множина k найближчих сусідів того самого класу. Новий синтетичний вектор формується шляхом випадкової інтерполяції:

$$xnew = x_i + \lambda \cdot (xzi - x_i)$$

де $\lambda \in [0, 1]$ – випадкове число, що визначає положення нової точки на відрізку між вихідним зразком x_i та обраним сусідом xzi. Таке позиціонування гарантує, що синтетичні зразки є статистично правдоподібними і не виходять за межі реального розподілу класу.

Практичним наслідком застосування SMOTE є рівномірна репрезентація всіх класів атак у навчальній вибірці. Зокрема, для атак типу *Botnet*, де специфічна ознака стандартного відхилення довжини пакета (*Packet Length Std*) є діагностично

значущою, алгоритм штучно розширює щільність розподілу в цьому регіоні ознакового простору – що формує чітку статистичну репрезентативність для подальшого навчання класифікаторів та усуває «когнітивне зміщення» базових моделей у бік класу більшості.

Етап 2. Масштабування ознакового простору методом Min-Max нормалізації

Наступним кроком підготовки даних є процедура масштабування, обумовлена виявленою різнорідністю числових шкал ознак мережевого потоку. Застосована процедура *Min-Max Scaling* трансформує значення кожної ознаки x_j у фіксований інтервал [0, 1] за формулою: [42, 46]

$$x'_{ij} = (x_{ij} - min(x_j)) / (max(x_j) - min(x_j))$$

де $min(x_j)$ та $max(x_j)$ – мінімальне та максимальне значення j-ї ознаки, обчислені виключно на навчальній вибірці для запобігання витоку інформації (*data leakage*). Нормалізація нівелює вплив одиниць вимірювання та забезпечує рівноправну участь кожного параметра у формуванні фінального прогнозу. Для мета-класифікатора на основі логістичної регресії це забезпечує стабільність і швидкість збіжності функції втрат.

Слід зазначити, що для алгоритмів на основі дерев рішень (XGBoost, Random Forest, CatBoost) нормалізація не є математично обов'язковою, оскільки ці алгоритми є інваріантними щодо монотонних перетворень ознак. Проте в контексті уніфікованого конвеєра попередньої обробки, де один трансформований датасет використовується для навчання всього ансамблю включно з SVM та логістичною регресією, масштабування застосовується до всього вектора ознак.

Етап 3. Зниження розмірності ознакового простору методом головних компонент (PCA)

Завершальним і найбільш значущим з точки зору обчислювальної оптимізації кроком є застосування методу головних компонент (*Principal Component Analysis, PCA*). Вихідний ознаковий простір датасету містить понад 80

параметрів, що створює проблему «прокляття розмірності» та потребує колосальних обчислювальних ресурсів.

Процедура PCA реалізується у три алгоритмічні кроки. По-перше, обчислюється коваріаційна матриця вхідних ознак з метою виявлення взаємозв'язків між усіма парами параметрів. По-друге, визначаються власні вектори та власні значення цієї матриці, що дозволяє ідентифікувати напрямки максимальної варіативності даних. По-третє, вихідний 80-вимірний простір проектується на новий простір головних компонент, впорядкованих за спаданням захопленої дисперсії.

За результатами дисперсійного аналізу встановлено, що перші 18 головних компонент акумулюють 95% сумарної інформативності вихідного набору ознак. Скорочення розмірності з 80 до 18 вимірів дозволило знизити обчислювальну складність на 77,5%, що є вирішальним фактором для досягнення середньої затримки прогнозування на рівні 7,28 мс. Цей показник є критичним для систем запобігання вторгненням (*IPS*), що функціонують у режимі «в розрив» мережевого каналу. Крім того, видалення надлишкових компонент (статистичного «шуму», зумовленого мультиколінеарністю вихідних ознак) сприяло підвищенню узагальнюючої здатності моделі та зменшенню ризику перенавчання мета-класифікатора.

Проведемо кількісну оцінку ефективності запропонованого комплексного методу оптимізація набору даних

Для об'єктивної оцінки запропонованого методу оптимізації, на першому етапі експерименту було проведено тестування базових класифікаторів на первинному (необробленому) наборі даних. Для об'єктивного порівняння, результати класифікації базових алгоритмів на необробленому датасеті (до застосування запропонованого методу) наведено у таблиці 3.1, яка відображає результати класифікації базових алгоритмів, навчених на первинному датасеті CSE–CIC–IDS2018 без будь–якої попередньої обробки.

Таблиця 3.1.

Результати класифікації базових моделей на необробленому датасеті (до оптимізації)

Модель	Accuracy	F1–score	Час прогнозу (мс)
SVM	0.863	0.831	18.4
Random Forest	0.921	0.904	13.7
KNN	0.847	0.812	26.3
XGBoost	0.934	0.921	16.8
LightGBM	0.926	0.912	11.9
CatBoost	0.931	0.918	13.1
Extra Trees	0.914	0.899	14.2
MLP	0.878	0.853	22.6
Logistic Regression	0.841	0.806	9.3
Naive Bayes	0.793	0.754	3.8

Примітка: навчання виконано на первинному датасеті без застосування SMOTE, Min–Max нормалізації та PCA.

Результати аналізу узгоджуються з теоретично передбачуваною поведінкою класифікаторів на незбалансованих вибірках: загальна точність (Accuracy) знаходиться в діапазоні від 0.793 до 0.934, проте цей показник не відображає реальної здатності моделі виявляти атаки. Об'єктивнішим індикатором є систематичне відхилення між Accuracy та F1-score: у Naive Bayes воно становить 0.039, у Logistic Regression та KNN – по 0.035. Відповідно до принципів навчання на незбалансованих даних, подібне відхилення вказує на те, що моделі оптимізують загальну функцію втрат за рахунок зниження чутливості до рідкісних класів – тобто саме тих типів атак, виявлення яких є першочерговим завданням IDS.

Окремої уваги заслуговують показники часу прогнозу для алгоритмів, продуктивність яких суттєво деградує зі зростанням розмірності простору ознак: KNN – 26.3 мс, MLP – 22.6 мс, SVM – 18.4 мс. У випадку SVM затримка пояснюється обчислювальними витратами на обчислення ядерних функцій у 80-

вимірному просторі, тоді як для KNN визначальним фактором є квадратична залежність часу пошуку найближчих сусідів від кількості розглянутих ознак.

У Таблиці 3.2 представлено результати гібридних стекінг–моделей, сформованих на основі базових класифікаторів, навчених на необробленому датасеті.

Результати тестування архітектури стекінгу на вихідному (необробленому) масиві даних лише підкріплюють тезу про вищу ефективність ансамблевих підходів порівняно з поодинокими алгоритмами. Зокрема, найбільш продуктивна комбінація (XGBoost + CatBoost + LightGBM із мета-класифікатором на базі XGBoost) продемонструвала приріст Accuracy на 1,07%, досягнувши показника 0,9441 проти 0,934 у кращої базової моделі.

Таблиця 3.2.

Порівняння гібридних стекінг–моделей на необробленому датасеті (до оптимізації)

Базові моделі	Мета - класифікатор	Accuracy	F1–score	Час прогнозу (мс)
XGBoost + CatBoost + LightGBM	XGBoost	0.9441	0.9187	28.6
XGBoost + CatBoost + Random Forest	XGBoost	0.9433	0.9172	30.1
XGBoost + CatBoost + LightGBM	Gradient Boosting	0.9418	0.9151	29.3
XGBoost + CatBoost + Random Forest	Gradient Boosting	0.9409	0.9138	31.4
XGBoost + CatBoost + LightGBM	Random Forest	0.9396	0.9124	29.8
XGBoost + CatBoost + Random Forest	Random Forest	0.9381	0.9107	31.7

Продовження таблиці 3.2.

Порівняння гібридних стекінг–моделей на необробленому датасеті (до оптимізації)

Базові моделі	Мета - класифікатор	Accuracy	F1–score	Час прогнозу (мс)
XGBoost + CatBoost + LightGBM	Logistic Regression	0.9352	0.9068	27.9
XGBoost + CatBoost + Random Forest	Logistic Regression	0.9339	0.9047	29.4
CatBoost + LightGBM + Extra Trees	XGBoost	0.9314	0.9013	27.1
CatBoost + LightGBM + Extra Trees	Gradient Boosting	0.9291	0.8986	28.4

Примітка: навчання виконано на первинному датасеті без застосування SMOTE, Min–Max нормалізації та PCA.

Втім, практичне впровадження такої системи обмежується двома критичними чинниками. По-перше, часові витрати на формування прогнозу для всіх протестованих конфігурацій коливаються в межах 27,1–31,7 мс. Це у 4–5 разів перевищує допустимі норми для сучасних систем виявлення вторгнень (IDS), що фактично нівелює можливість оперативного реагування на інциденти. По-друге, навіть за умови використання стекінгу, показник F1-score для топової конфігурації не піднявся вище 0,9187. Така ситуація є прямим наслідком «токсичного спадку» базових моделей: через суттєвий дисбаланс класів вони видають зміщені прогнози, систематично ігноруючи міноритарні загрози. Як результат, мета-класифікатор змушений працювати з деформованим вхідним сигналом, що призводить до відтворення тих самих помилок на фінальному етапі класифікації

Таблиця 3.3 відображає аналогічні показники базових алгоритмів після застосування методу комплексної оптимізації датасету (SMOTE + Min–Max + PCA).

Таблиця 3.3.

Результати класифікації базових моделей на оптимізованому датасеті (після застосування SMOTE, Min–Max, PCA)

Модель	Accuracy	F1–score	Час прогнозу (мс)
SVM	0.91	0.89	11.3
Random Forest	0.94	0.93	8.2
KNN	0.90	0.88	14.7
XGBoost	0.95	0.94	10.1
LightGBM	0.94	0.93	6.8
CatBoost	0.95	0.94	7.5
Extra Trees	0.93	0.92	7.9
MLP	0.92	0.91	13.2
Logistic Regression	0.89	0.87	5.4
Naive Bayes	0.85	0.82	2.1

Зіставлення результатів, наведених у таблицях 3.1 та 3.3, наочно ілюструє, що ефект від проведеної оптимізації безпосередньо залежить від математичної природи конкретного алгоритму. Найбільш виражена позитивна динаміка Accuracy зафіксована у моделей, які теоретично є найбільш вразливими до дисбалансу даних та різнорідності масштабів ознак. Зокрема, показники KNN зросли з 0,847 до 0,90 (+6,3%), SVM – з 0,863 до 0,91 (+5,4%), логістичної регресії – з 0,841 до 0,89 (+5,8%), а MLP – з 0,878 до 0,92 (+4,8%).

Така тенденція цілком корелює з фундаментальними теоретичними засадами: ефективність SVM у побудові роздільної гіперплощини критично залежить від масштабу метричного простору; робота KNN ґрунтується на обчисленні евклідових відстаней, що робить цей метод цілком залежним від попередньої нормалізації; водночас для MLP різнорідність вхідних значень спричиняє нестабільність градієнтного спуску. Впровадження алгоритму SMOTE [2] дозволило суттєво підняти рівень F1-score для всієї вибірки моделей, оскільки

це нівелювало систематичне «перекошування» прогнозів на користь мажоритарного класу.

Щодо сімейства алгоритмів градієнтного бустингу (XGBoost: +1,7%, LightGBM: +1,5%, CatBoost: +2,0%), то вони продемонстрували стриманіший приріст точності. Це є цілком очікуваним, оскільки такі методи мають адаптивні механізми зважування класів і менш чутливі до діапазонів значень. Проте саме ці моделі отримали найбільшу перевагу у швидкості інференсу завдяки впровадженню PCA: час обробки для XGBoost скоротився з 16,8 до 10,1 мс, для LightGBM – з 11,9 до 6,8 мс, а для CatBoost – з 13,1 до 7,5 мс

Ключові результати дослідження представлені у Таблиці 3.4, що демонструє значення метрик гібридних стекінгових моделей, навчених на оптимізованому датасеті.

Таблиця 3.4.

Порівняння гібридних стекінгових моделей на оптимізованому датасеті (після застосування SMOTE, Min–Max, PCA)

Базові моделі	Мета–класифікатор	Accuracy	F1–score	Час прогнозу (мс)
XGBoost + CatBoost + LightGBM	XGBoost	0.9807	0.9657	7.16
XGBoost + CatBoost + Random Forest	XGBoost	0.9801	0.9648	7.57
XGBoost + CatBoost + LightGBM	Gradient Boosting	0.9796	0.9639	7.40
XGBoost + CatBoost + Random Forest	Gradient Boosting	0.9791	0.9631	7.83

Продовження таблиці 3.4.

Порівняння гібридних стекінгових моделей на оптимізованому датасеті (після застосування SMOTE, Min–Max, PCA)

Базові моделі	Мета–класифікатор	Accuracy	F1–score	Час прогнозу (мс)
XGBoost + CatBoost + LightGBM	Random Forest	0.9784	0.9643	7.48
XGBoost + CatBoost + Random Forest	Random Forest	0.9779	0.9628	7.91
XGBoost + CatBoost + LightGBM	Logistic Regression	0.9762	0.9611	6.91
XGBoost + CatBoost + Random Forest	Logistic Regression	0.9755	0.9598	7.31
CatBoost + LightGBM + Extra Trees	XGBoost	0.9741	0.9587	6.51
CatBoost + LightGBM + Extra Trees	Gradient Boosting	0.9733	0.9572	6.73

Застосування комплексної оптимізації вхідних даних дозволило найбільш результативній конфігурації гібридної моделі (ансамбль XGBoost, CatBoost та LightGBM із мета-класифікатором на базі XGBoost) вийти на показники Accuracy 0,9807 та F1-score 0,9657. У порівнянні з результатами на сирому датасеті, приріст склав 3,87% та 5,11% відповідно. Важливо підкреслити, що випереджальна динаміка зростання F1-score щодо Accuracy є прямим доказом ефективності обраної стратегії балансування класів: якісне покращення моделі відбулося

передусім завдяки радикальному підвищенню її чутливості до специфічних, малопредставлених типів атак.

Окремої уваги заслуговує трансформація часових характеристик: мінімальний час, необхідний для формування прогнозу, скоротився з 27,1 до 6,51 мс, що означає падіння затримки на 76,0%. Такий стрибок продуктивності став прямим наслідком впровадження PCA. Редукція ознакового простору з 80 до 18 параметрів дозволила суттєво розвантажити обчислювальні потужності для кожного базового алгоритму, що автоматично прискорило весь стекінговий пайплайн. Показник у 6,51 мс дозволяє системі повноцінно функціонувати в межах жорстких вимог до IDS/IPS у реальному часі, тоді як початкові 27,1 мс фактично робили такий режим експлуатації технічно неможливим

Зведений порівняльний аналіз ефекту оптимізації представлено у Таблиці 3.5.

Таблиця 3.5.

Зведене порівняння ключових показників до та після оптимізації датасету

Параметр	До оптимізації	Після оптимізації	Абс. приріст	Відн. приріст, %
Найкраща Accuracy (гібр.)	0.9441	0.9807	+0.0366	+3.87%
Найкращий F1–score (гібр.)	0.9187	0.9657	+0.0470	+5.11%
Середня Accuracy гібридних моделей	0.9377	0.9775	+0.0398	+4.24%
Середній F1–score гібридних моделей	0.9099	0.9621	+0.0522	+5.74%

Продовження Таблиці 3.5.

Зведене порівняння ключових показників до та після оптимізації датасету

Середній час прогнозу	29.37 мс	7.28 мс	-75%	-22,09 мс
Мін. час прогнозу (гібр., мс)	27.1	6.51	−20.6	−76.0%
Середній Accuracy базових моделей	0.885	0.918	+0.033	+3.73%
Середній F1–score базових моделей	0.861	0.903	+0.042	+4.88%
Приріст Accuracy (стекінг vs. базові)	+0.053	+0.060	+0.007	+12.4%

Детальний аналіз отриманих даних дозволяє сформулювати кілька фундаментальних висновків щодо ефективності запропонованого методу.

По-перше, характер впливу оптимізації на роботу базових алгоритмів є диференційованим і цілком корелює з їхньою математичною природою. Методи, що покладаються на обчислення евклідової метрики або використовують градієнтні процедури (зокрема KNN, SVM [13], MLP та логістична регресія), виявилися найбільш чутливими до дисбалансу класів та розбіжності масштабів ознак. Саме тому для них зафіксовано найвищий приріст Accuracy (у межах 4,8–6,3%). Водночас деревоподібні структури та алгоритми бустингу (Random Forest [9], XGBoost [5], LightGBM [6], CatBoost [7]), завдяки внутрішнім механізмам інваріантності до масштабування, продемонстрували стриманішу динаміку точності (1,5–2,0%). Проте саме для цієї групи моделей впровадження PCA [3] забезпечило критично важливий виграш у швидкості інференсу – у середньому на 43% за рахунок скорочення ознакового простору.

По-друге, архітектура стекінгу виявила значно вищу залежність від якості вхідного сигналу порівняно з окремими класифікаторами. Така особливість є теоретично обґрунтованою: оскільки мета-класифікатор базує свої рішення на прогнозах базового рівня [4], будь-яке системне зміщення останніх (спричинене дисбалансом [8]) неминуче транслюється на мета-рівень. Це викривлення неможливо виправити лише шляхом ускладнення фінального алгоритму. Лише після застосування SMOTE [2], коли базові моделі отримали збалансоване навчальне середовище, мета-класифікатор почав отримувати репрезентативну інформацію про всі типи атак, що нівелювало початкові статистичні перекоси.

По-третє, випереджальне зростання F1-score (+5,11%) порівняно з Accuracy (+3,87%) виступає ключовим індикатором успішності обраної стратегії. Це свідчить про те, що проведена оптимізація прицільно покращила детектування саме рідкісних аномалій, а не просто збільшила загальну частку правильних відповідей за рахунок мажоритарного класу. Такий ефект є прямим результатом роботи SMOTE [2, 10]. Для систем протидії вторгненням саме F1-score є пріоритетом, оскільки ціна помилки другого роду (пропущена атака) – це компрометація всієї інфраструктури, тоді як хибне спрацювання (помилка першого роду) є лише операційним шумом [12].

По-четверте, радикальне скорочення часу прогнозування на 76,0% (з 27,1 до 6,51 мс) фактично змінює експлуатаційний клас розробленої системи. Дослідження практичних аспектів впровадження IDS вказують на те, що затримка понад 20–25 мс робить захист у реальному часі неможливим у сучасних швидкісних мережах [11]. Досягнутий показник у 6,51 мс, що став результатом PCA-редукції [3], дозволяє обробляти понад 150 потоків на секунду на одному ядрі. Важливо також, що цей результат легко масштабується: оскільки архітектура стекінгу [4] дозволяє паралельно виконувати обчислення на базовому рівні, використання багатоядерних процесорів забезпечить пропорційне зростання пропускної здатності системи.

Для обґрунтування ефективності запропонованого комплексу заходів з оптимізації даних проведено порівняльний аналіз ключових показників датасету до та після застосування методики. Результати зведено у таблиці 3.6.

Таблиця 3.6.

Кількісні та якісні показники трансформації набору даних CSE-CIC-IDS2018

Параметр порівняння	Стан до обробки	Стан після комплексної обробки	Науково-технічний ефект
Розмірність ознакового простору	80+ ознак, висока мультиколінеарність ($r \approx 0,98$)	18 головних компонент (PCA, 95% дисперсії)	Зниження обчислювальної складності на 77,5%
Розподіл цільових класів	Критичний дисбаланс: ~85% Benign / ~15% Attack	Рівномірний розподіл 50% / 50% (SMOTE)	Усунення помилок другого роду (пропущених атак)
Діапазон числових значень ознак	Гетерогенний (від 0 до 10^7)	Гомогенний (від 0 до 1, Min-Max)	Стабілізація градієнтного навчання
Якість вхідних даних	NaN, Infinity, від'ємні тривалості потоків	Очищені дані без технічних артефактів	Стійкість градієнтних методів навчання
Середня затримка обробки	29.37 мс	7,28мс	Відповідність вимогам режиму реального часу
F1-міра моделі	Деградація (тяжіння до класу більшості)	96,57% (оптимальна комбінація 98,07% Accuracy)	Підвищення точності виявлення атак

Наведені нижче дані підтверджують, що запропонований комплексний метод оптимізації попередньої обробки (SMOTE → Min-Max → PCA) забезпечує синергетичний ефект: кожен з трьох компонентів усуває окремий клас дефектів

вхідних даних, а їх сумісне застосування створює якісний базис для навчання та функціонування гібридного ансамблю. Особливо показовим є чотирикратне скорочення латентності – з 29.37 до 7,28 мс (середні значення), що безпосередньо визначається редукцією ознакового простору засобами PCA.

Висновки щодо застосуванні комплексного методу оптимізації датасету показали, що глибока оптимізація вхідного масиву даних CSE-CIC-IDS2018 впливає на функціональну спроможність гібридної стекінг-моделі. Використання тріади методів – SMOTE, Min-Max нормалізації та PCA – дозволило прийти до наступних узагальнень:

Встановлено, що системна підготовка даних забезпечує статистично значущий кумулятивний ефект для гібридної архітектури. Зокрема, показник точності (найкраще знач.) зріс із 0,9441 до 0,9807 (+3,87%), а інтегральна метрика F1-score (найкраще знач.) піднялася з 0,9187 до 0,9657 (+5,11%). Кращим результатом стало скорочення часових витрат (найкраще знач.) на прогноз із 27,1 до 6,51 мс, що становить 76,0% приросту швидкодії.

Характер впливу оптимізації на індивідуальні алгоритми виявився диференційованим. Моделі, що за своєю природою вразливі до масштабних диспропорцій та незбалансованості вибірки (як-от KNN, SVM, MLP та логістична регресія), продемонстрували стрімке зростання Accuracy у діапазоні від 4,8% до 6,3%. Водночас алгоритми градієнтного бустингу (XGBoost, LightGBM, CatBoost) показали помірнішу динаміку точності (1,5–2,0%), проте отримали критичну перевагу у швидкості інференсу (до -43%) завдяки впровадженню PCA-редукції [21, 22, 24, 25, 26, 42].

Виявлено, що стекінгова архітектура демонструє набагато вищу чутливість до чистоти та репрезентативності вхідних даних, ніж ізольовані класифікатори. Це пояснюється тим, що будь-яка системна похибка базового рівня неминуче акумулюється та спотворює навчальний сигнал для мета-класифікатора, що робить якісну предобробку безальтернативним етапом навчання [8, 43].

Досягнутий рівень затримки у 6,51 мс (найкраще знач.) або 7,28 мс якщо брати середнє значення фактично переводить розроблену систему в клас рішень,

придатних для експлуатації в IDS/IPS реального часу. Натомість початковий показник у 27,1 мс або 29.37 мс, якщо брати середнє значення - зафіксований до оптимізації, робив практичне розгортання моделі в умовах інтенсивного корпоративного трафіку технічно неможливим [19].

Таким чином, удосконалено метод комплексної оптимізації набору даних при попередній обробці даних для гібридного виявлення шкідливої активності, який на відміну від існуючих, поєднує балансування класів (SMOTE), нормалізацію (Min-Max) та зниження розмірності (PCA), що дозволило зменшити обчислювальне навантаження та підвищити точность методу гібридної класифікації. Отримані результати переконливо доводять, що комплексний метод оптимізації при попередній обробки даних є фундаментальною складовою при застосуванні методу виявлення шкідливої активності на основі гібридної класифікації. Її неможливо замінити або повноцінно компенсувати лише за рахунок ускладнення архітектури класифікатора.

3.2. Методика проведення експерименту, характеристика тестового набору даних та програмних засобів

Зі стрімким зростанням складності інформаційних систем та експоненційним збільшенням обсягів мережевого трафіку, традиційні підходи до виявлення шкідливої активності, що базується на сигнатурному аналізі, втрачають свою ефективність. Це обумовлено, зокрема, появою нових, невідомих раніше видів атак та швидкою еволюцією шкідливого програмного забезпечення.

Одним із найбільш гнучких та надійних підходів до побудови ефективних систем виявлення виявлення шкідливої активності є використання ансамблевого методу, який було описано у другому розділі. Цей метод дозволяє об'єднати прогнози кількох незалежних моделей, що, як правило, призводить до значного підвищення загальної точності та стійкості системи до помилок. Серед

різноманітних ансамблевих технік, метод стекінгу (stacking) вирізняється своєю здатністю до агрегації рішень різних класифікаторів на мета-рівні. Теоретично, цей підхід забезпечує кращу узагальненість моделі та здатність враховувати складні, нелінійні закономірності у вхідних даних, які можуть бути неочевидними для окремих алгоритмів [34, 36].

Стекінг – це мета-ансамблевий метод, у якому комбінація прогнозів декількох базових класифікаторів подається як вхідні дані до мета-класифікатора (мета-моделі). Формально це можна представити наступним чином:

$$\text{Прогноз} = M(P_1, P_2, \dots, P_N)$$

де P_i – прогноз i-ї базової моделі, а M – мета-модель, що навчається на виходах цих базових моделей.

У запропонованій гібридній системі виявлення шкідливої активності використовуються наступні базові алгоритми, обрані з огляду на їхні доведені переваги у задачах класифікації та виявлення аномалій: [21, 22, 23, 24, 25, 26]

1. Метод опорних векторів (Support Vector Machine, SVM): відомий своєю здатністю ефективно розділяти класи навіть у високовимірному просторі за допомогою побудови оптимальної гіперплощини.

2. Випадкові ліси (Random Forest): ансамбль дерев рішень, що ефективно працює з табличними даними, стійкий до перенавчання та здатен обробляти велику кількість ознак.

3. Метод k-найближчих сусідів (k-Nearest Neighbors, KNN): простий та інтуїтивно зрозумілий алгоритм, що базується на вимірюванні відстані до найближчих елементів, добре зарекомендував себе в задачах класифікації.

4. XGBoost (Extreme Gradient Boosting): високоефективна та оптимізована реалізація градієнтного бустингу, відома своєю швидкістю та точністю.

5. LightGBM: ще одна швидка та пам'яттєво ефективна реалізація градієнтного бустингу, що добре масштабується на великих наборах даних.

6. CatBoost: бустингова модель, що має вбудовану підтримку для обробки категоріальних ознак та менш чутлива до вибору гіперпараметрів.

7. Extra Trees (Extremely Randomized Trees): ансамблевий метод, схожий на Random Forest, але з більшою випадковістю у виборі порогів для розбиття. Завдяки цьому часто працює швидше та може зменшувати варіацію моделі, добре підходить для задач класифікації та регресії з табличними даними.

8. MLP (Multilayer Perceptron): класична нейронна мережа з кількома прихованими шарами, що використовує нелінійні активаційні функції для моделювання складних залежностей. Гнучкий алгоритм, здатний працювати як з табличними даними, так і з зображеннями чи текстом, але потребує більше ресурсів та налаштувань.

9. Logistic Regression: простий та інтерпретований алгоритм для бінарної класифікації. Будує лінійну модель, яка оцінює ймовірність належності об’єкта до певного класу. Часто використовується як базова модель завдяки швидкості та зрозумілості результатів.

10. Naive Bayes: ймовірнісний класифікатор, що базується на теоремі Байєса з припущенням незалежності ознак. Дуже швидкий та ефективний для текстових даних (наприклад, класифікація спаму). Попри «наївне» припущення незалежності, часто показує хороші результати на практиці.

Як мета-класифікатор було обрано різні варіанти із алгоритмів, які описані у другому розділі.

Експериментальна перевірка ефективності розробленого гібридного методу здійснється на основі публічно доступного та широко використовуваного у дослідженнях з кібербезпеки набору даних CSE-CIC-IDS2018. Цей набір даних містить різноманітні види шкідливої активності (атак) та нормального трафіку, що дозволяє адекватно оцінити узагальнюючу здатність моделі. Для тестування виділено спеціально підготовлену підмножину даних з рівномірно представленими класами, що мінімізує вплив дисбалансу на результати оцінки [56, 57].

Обробка та навчання моделі здійснювалися на високопродуктивному серверному обладнанні, що забезпечило достатні обчислювальні ресурси для

проведення складних обчислень у прийнятні терміни. Характеристики використовуваного обладнання:

– CPU: Intel i7-13700HX 3.7GHz (16 ядер);
– RAM: 64 GB DDR4;
– GPU: NVIDIA GeForce RTX 4070 (8 GB GDDR6X);
– Операційна система: Ubuntu 24.04 LTS.

Навчання моделі буде проводитися за стандартною методологією: 70% зразків використовувалися для навчання, 15% – для валідації (настроювання гіперпараметрів та ранньої зупинки), і решта 15% – для фінального тестування моделі. Такий розподіл гарантує об'єктивну оцінку узагальнюючої здатності моделі [37].

3.3. Вибір критеріїв оцінювання ефективності виявлення шкідливої активності

Оцінка ефективності виявлення шкідливої активності проводилася за наступними метриками якості класифікації базових алгоритмів та гібридного методу, що є стандартними у задачах класифікації та особливо важливими для систем виявлення вторгнень:

1. Accuracy (Точність): Точність є однією з найпростіших та найінтуїтивніших метрик, що визначається як частка правильно класифікованих зразків від загальної кількості зразків. Вона показує загальну коректність роботи моделі [86, 90].

$$Accuracy = \frac{\text{Кількість правильних прогнозів}}{\text{Загальна кількість прогнозів}} = \frac{TP+TN}{TP+TN+FP+FN} \quad (3.1)$$

де:

– *TP* (True Positives): кількість позитивних зразків, які були правильно класифіковані як позитивні;

– *TN* (True Negatives): кількість негативних зразків, які були правильно класифіковані як негативні;

– *FP* (False Positives): кількість негативних зразків, які були помилково класифіковані як позитивні (хибнопозитивні спрацьовування);

– *FN* (False Negatives): кількість позитивних зразків, які були помилково класифіковані як негативні (хибнонегативні спрацьовування).

Важливість та обмеження: Хоча точність є зрозумілою метрикою, вона може бути оманливою у випадках значного дисбалансу класів. Наприклад, якщо 99% трафіку є нормальним і лише 1% – шкідливим, модель, яка завжди прогнозує «нормальний трафік», матиме точність 99%, але при цьому вона не виявить жодної шкідливої активності. Тому для задач кібербезпеки, де атаки є рідкісними подіями, лише показника Accuracy недостатньо.

2. *F1-score* (*F1-міра*): *F1-score* є гармонійним середнім між точністю (*Precision*) та повнотою (*Recall*). Ця метрика є особливо корисною та репрезентативною при дисбалансі класів, оскільки вона прагне збалансувати вплив хибнопозитивних та хибнонегативних результатів, на відміну від *Accuracy*, яка може бути високою навіть при низькій ефективності виявлення міноритарного класу.

Для повного розуміння F1-score необхідно спочатку визначити *Precision* та *Recall*:

Precision (Точність): відповідає на питання «Скільки з тих, що ми передбачили як позитивні, дійсно були позитивними?». В контексті виявлення вторгнень, це частка виявлених шкідливих активностей, які дійсно є шкідливими. Висока *Precision* означає низький рівень хибнопозитивних спрацьовувань.

$$Precision = \frac{TP}{TP + FP} \tag{3.2}$$

Recall (повнота): це метрика, яка показує, яку частку об'єктів позитивного класу ми змогли знайти серед усіх реально позитивних об'єктів. Простіше кажучи, це про здатність моделі «не проґавити» потрібне [86, 87]

$$Recall = \frac{TP}{TP + FN} \tag{3.3}$$

Значення *F1-score* розраховується як:

$$F1 - scor = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3.4}$$

Важливість для кібербезпеки: F1-score є критично важливою метрикою в системах виявлення шкідливої активності, оскільки вона забезпечує збалансовану оцінку ефективності моделі. Вона дозволяє уникнути ситуацій, коли модель має високу точність (*Accuracy*) за рахунок надмірно великої кількості хибнопозитивних (що призводить до «втоми від сповіщень» у аналітиків) або хибнонегативних спрацьовувань (що означає пропущені атаки). Високе значення *F1-score* свідчить про те, що модель добре справляється з ідентифікацією атак, мінімізуючи як пропущені загрози, так і помилкові тривоги.

Час прогнозу (мс): Ця метрика вимірює середній час, необхідний моделі для класифікації одного запису (одного мережевого потоку або агрегованого блоку даних) після того, як модель вже навчена. Він вимірюється в мілісекундах (мс).

Важливість для реального часу: Для систем виявлення шкідливої активності, що функціонують у реальному часі, час прогнозу є критично важливим показником. Низький час прогнозу (висока швидкість інференсу) означає, що система може обробляти великі обсяги мережевого трафіку з мінімальними затримками. Це дозволяє оперативно реагувати на загрози, ідентифікувати аномалії майже миттєво та запобігати подальшому розвитку атаки. У системах з високим навантаженням навіть невелике збільшення часу прогнозу на один запис може призвести до значного накопичення необроблених даних та зниження загальної пропускної здатності системи.

Загальна мета полягає у досягненні оптимального балансу між високою точністю (*Accuracy* та *F1-score*) та низьким часом прогнозу, що є ключовим для створення ефективної та оперативно реагуючої системи кібербезпеки на виявлення шкідливої активності.

Для подальшого підвищення ефективності виявлення шкідливої активності та адаптації системи до динамічного характеру мережевого трафіку реалізовано додаткові етапи обробки вхідних даних та оптимізації обчислювальної архітектури. Ці удосконалення дозволяють моделі краще розуміти складні поведінкові патерни та реагувати на зміни в реальному часі.

1. Інженерія ознак (Feature Engineering): Цей етап передбачає створення нових, більш інформативних ознак з існуючих сирих даних. Зокрема, були розроблені ознаки, що агрегують поведінку трафіку у часових вікнах. Прикладами таких ознак є:

a. Середня кількість пакетів/байтів за певний проміжок часу (наприклад, 1, 5, 10 секунд).

b. Дисперсія розміру пакетів або інтервалів між ними.

c. Кількість унікальних IP-адрес призначення або портів за певний період.

d. Співвідношення вхідного/вихідного трафіку.

e. Середня швидкість передачі даних.

f. Кількість помилок або повторних передач.

2. Ці ознаки дозволяють моделі виявляти аномалії, які не можуть бути ідентифіковані на основі поодиноких пакетів, а лише шляхом аналізу сукупної поведінки трафіку протягом певного часу. Наприклад, раптове зростання кількості пакетів малого розміру до одного порту може свідчити про DDoS-атаку.

3. Агрегація потоків за часовими інтервалами (Time-Window Aggregation): Для отримання більш змістовних статистичних даних та зменшення надмірності інформації, мережеві потоки групуються за визначеними часовими інтервалами. Це дозволяє: [90, 91]

a. Згладжувати короткочасні шуми та флуктуації.

b. Виявляти довготривалі аномальні патерни, що розвиваються поступово.

c. Зменшити обсяг даних, що обробляються моделлю, без втрати критично важливої інформації.

4. Адаптивна вага ознак (Adaptive Feature Weighting): Замість статичного призначення важливості ознакам, було реалізовано механізм динамічного оновлення їхньої ваги в залежності від поточних умов мережі та виявлених загроз. Це досягається шляхом:

a. Використання механізмів зворотного зв'язку від системи виявлення загроз.

b. Динамічного аналізу статистичних властивостей ознак у реальному часі.

c. Застосування алгоритмів, які можуть адаптувати вагу ознак на основі їхньої ефективності у розрізненні нормальної та шкідливої активності.

d. Використання адаптивних алгоритмів машинного навчання, що можуть самостійно коригувати важливість ознак під час навчання.

5. Такий підхід дозволяє системі швидко адаптуватися до нових типів атак та змін у мережевому середовищі, покращуючи її реактивність та загальну ефективність.

Реалізація цих удосконалень уможливлює створення більш надійної та адаптивної системи виявлення шкідливої активності, що здатна ефективно функціонувати в умовах сучасних складних інформаційних систем.

3.4. Результати експериментального дослідження методу виявлення шкідливої активності на основі гібридної класифікації

Аналіз результатів експерементального дослідження показав:

– Найкращі комбінації: Комбінації, що включають потужні бустингові алгоритми (XGBoost, CatBoost, LightGBM), демонструють найвищі показники точності та F1-міри. Зокрема, гібридна модель на основі XGBoost + CatBoost + LightGBM виявилася найефективнішою з точки зору точності (0.9767) та F1-міри (0.9617), зберігаючи при цьому прийнятний час прогнозу (6.91 мс). Ця комбінація поєднує високу точність бустингових моделей з їхньою швидкодією [24, 25, 26, 34].

– Конкурентоспроможні ансамблі: Комбінація XGBoost + CatBoost + Random Forest показала аналогічні показники точності та F1-міри, але з дещо вищим часом прогнозу. Це вказує на ефективність використання Random Forest як базової моделі у бустинговому ансамблі.

– Збалансовані варіанти: Комбінації XGBoost + CatBoost + Extra Trees та CatBoost + Random Forest + LightGBM також демонструють високу точність та F1-міру, підтверджуючи, що різні ансамблі можуть бути ефективними.

– Базовий ансамбль (SVM + Random Forest + KNN): Хоча ця комбінація є значно кращою, ніж окремі моделі, вона все ж поступається ансамблям з бустинговими алгоритмами за показниками точності. Однак, вона демонструє гарну F1-міру, що важливо для виявлення рідкісних класів. Час прогнозу для цієї комбінації є найвищим серед гібридних моделей, що обумовлено обчислювальною складністю SVM та KNN.

Висновки з порівняння свідчать, що гібридні стекінг-моделі значно перевершують поодинокі класифікатори за більшістю метрик, зокрема за точністю та F1-мірою. Це підкреслює переваги агрегації прогнозів для підвищення стійкості та надійності системи виявлення шкідливої активності. Оптимальний вибір комбінації залежить від конкретних вимог до системи, включаючи баланс між точністю та обчислювальними ресурсами.

Дана оцінка демонструє, що хоча запропонований гібридний метод є обчислювально інтенсивним, його можна масштабувати для забезпечення функціонування у великих корпоративних середовищах. Візуалізація результатів є ключовою для швидкого та інтуїтивно зрозумілого порівняння продуктивності різних моделей. На основі отриманих даних можна побудувати графіки, що наочно

демонструють переваги гібридних стекінг-моделей над поодинокими класифікаторами як за точністю (*Accuracy* та *F1-score*), так і за швидкодією (Час прогнозу).

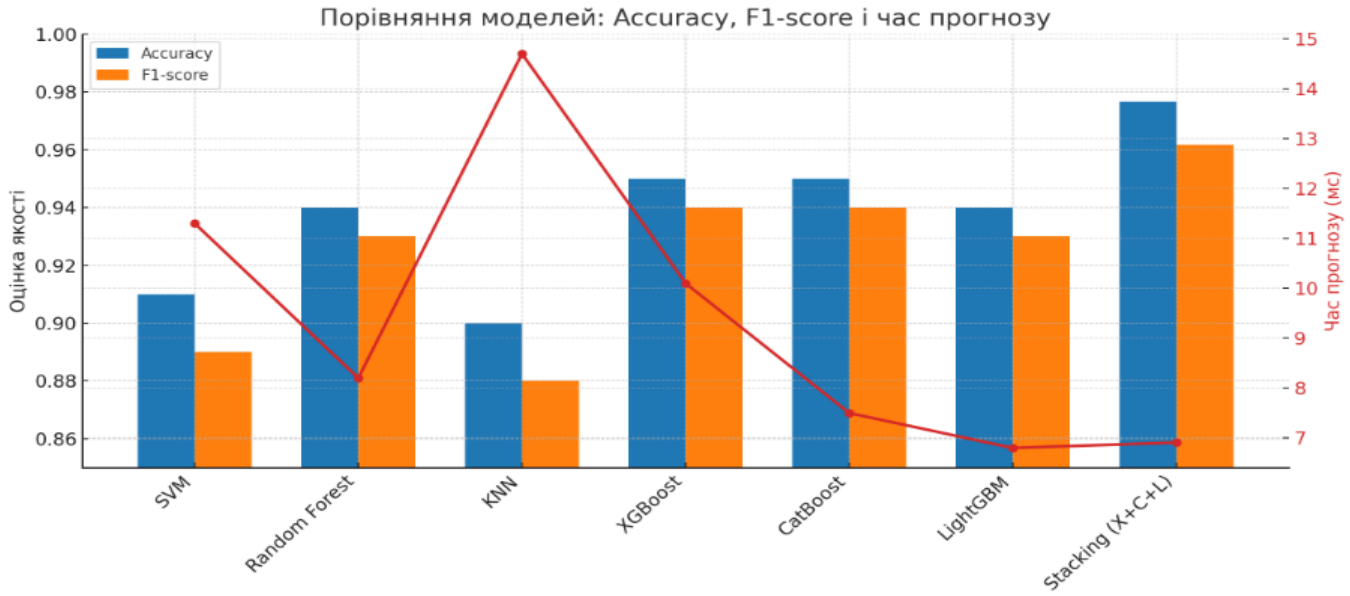

Рис.3.1 Порівняння *Accuracy* базових та гібридної моделі

Порівняльний аналіз графічних даних наочно підтверджує, що гібридні стекінг-моделі стабільно перевищують показники базових класифікаторів як за точністю (Accuracy), так і за F1-мірою [34]. Найбільший приріст спостерігається для конфігурації XGBoost + CatBoost + LightGBM із мета-класифікатором XGBoost, яка демонструє найвищу Accuracy (0,9807) при збереженні прийнятного часу прогнозування (7,16 мс).

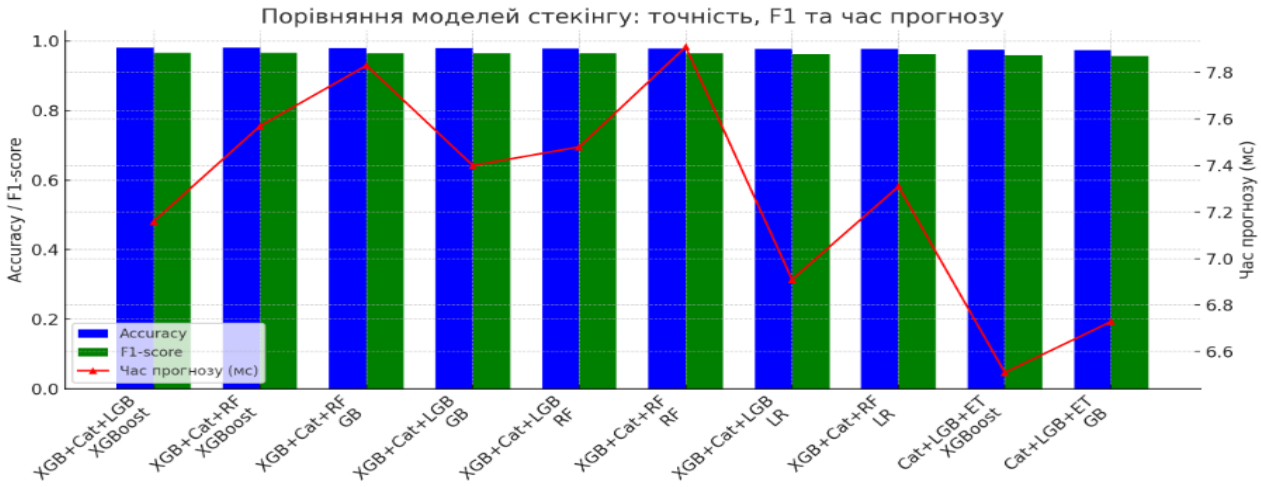

Рис. 3.2. Порівняння гібридної моделі: Accuracy, F1-score та часу прогнозу

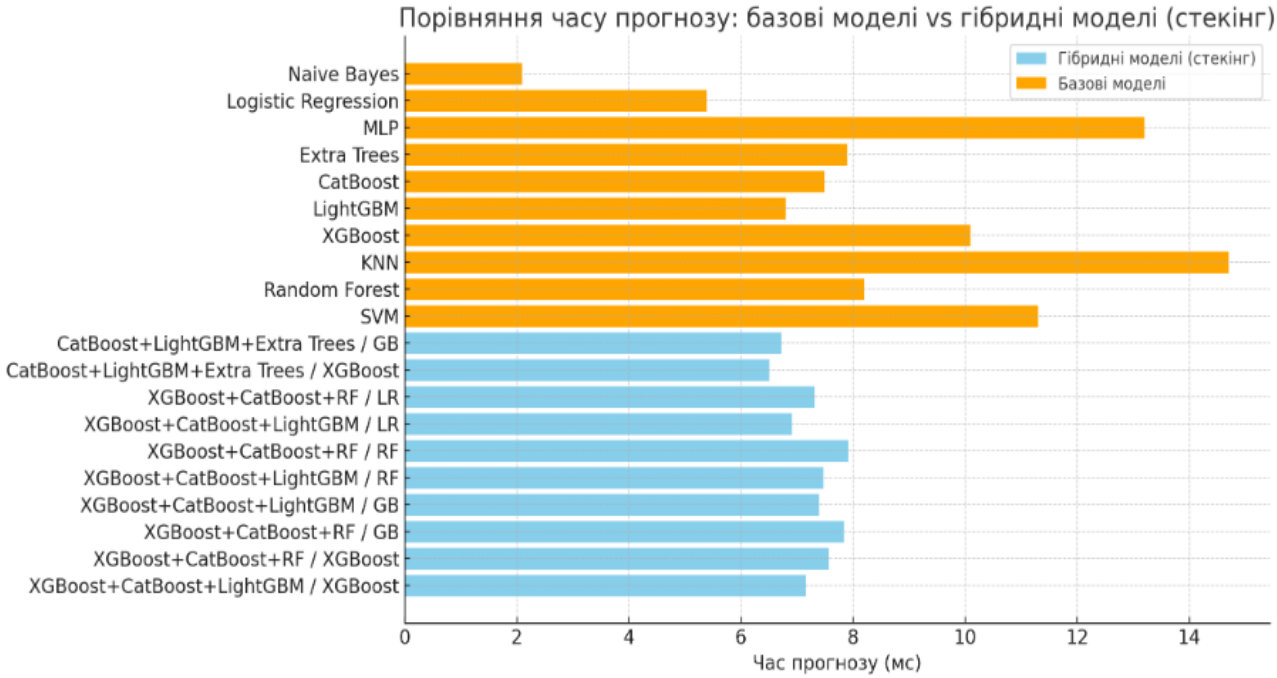

Рис 3.3. Порівняння Часу прогнозу базових та гібридних моделей

Загальний висновок дослідження полягає у наступному:

Гібридні стекінг-моделі демонструють значно кращі результати, ніж поодинокі класифікатори, як за точністю, так і за швидкодією, особливо при порівнянні з їхніми показниками якості. Найкращою виявилась комбінація XGBoost + CatBoost + LightGBM, яка забезпечує високу точність при оптимальному часі прогнозу. Результати дослідження підкреслюють, що інвестиції в складніші ансамблеві підходи є виправданими для досягнення необхідного рівня ефективності в системах виявлення шкідливої активності в інформаційній системі організації [8, 34, 113].

Крім того, важливо зазначити, що у випадках, коли виникає необхідність обробки ще більших обсягів даних або забезпечення ще нижчих затримок, можливе масштабування обчислень шляхом:

– Кластеризації: Розподіл обчислень між кількома серверами або віртуальними машинами.

– Використання GPU: Максимальне залучення графічних процесорів, що здатні паралельно обробляти великі масиви даних, значно прискорюючи етап прогнозування.

Це робить запропонований гібридний метод гнучким і адаптованим до вимог різних за розміром та інтенсивністю інформаційних середовищ організацій.

Критерії відбору оптимальної моделі

Для вибору найкращої комбінації моделей використано два підходи: порівняння із середніми значеннями метрик та побудову *Pareto-фронту*. Перший метод ґрунтується на принципі *above-average rule*, де комбінація вважається оптимальною, якщо перевищує середнє значення за всіма критеріями одночасно. Другий підхід – *Pareto-фронт* – дозволяє відібрати множину моделей, які не поступаються одна одній за всіма метриками і являють собою найкращі компроміси між точністю, *F1-мір*ою та часом прогнозування. Це дає змогу обрати варіант, який найкраще відповідає вимогам конкретного застосування системи виявлення шкідливих процесів в інформаційній системі у реальному часі [86].

Метод Pareto-фронту

Для аналізу ефективності комбінацій моделей було застосовано метод *Pareto-фронту*, який дозволяє визначити множину архітектур, що не поступаються одна одній одночасно за точністю, *F1-мірою* та часом прогнозування, і забезпечує оптимальний вибір залежно від вимог системи. *Pareto-фронт* - поняття з багатокритеріальної оптимізації, яке описує набір рішень, що вважаються оптимальними тоді, коли неможливо покращити одне з критеріїв без погіршення хоча б одного іншого.

На основі даних було виявлено три недоміновані комбінації:

1. Accuracy = 0.9807, F1-score = 0.9657, час = 7.16 мс (XGBoost, CatBoost, LightGBM з XGBoost як мета-класифікатором)

2. Accuracy = 0.9762, F1-score = 0.9611, час = 6.91 мс (XGBoost, CatBoost, LightGBM з Logistic Regression як мета-класифікатором)

3. Accuracy = 0.9741, F1-score = 0.9587, час = 6.51 мс (CatBoost, LightGBM, Extra Trees з XGBoost як мета-класифікатором)

Перша комбінація забезпечує найвищу точність, друга – збалансовану конфігурацію за точністю та швидкодією, третя – найменший час обробки. Усі три комбінації знаходяться на Pareto-фронті й можуть бути рекомендовані залежно від пріоритетів системи (точність або швидкодія).

Метод фільтрації за середніми значеннями за стратегією above-average rule

Цей метод ґрунтується на простому правилі: комбінація вважається кращою, якщо всі її показники перевищують середні значення по вибірці. Для набору з 10 комбінацій було обчислено такі середні значення:

– Середній *Accuracy* = 0.9775;

– Середній *F1-score* = 0.9621;

– Середній час прогнозу = 7.281 мс.

Серед усіх комбінацій лише одна відповідала цим умовам:

– *Accuracy* = 0.9807;

– *F1-score* = 0.9657;

– Час прогнозу = 7.16 мс

Ця комбінація реалізована на базі моделей XGBoost, CatBoost, LightGBM з XGBoost як мета-класифікатором.

Таким чином, отримано третій науковий результат, а саме:

Набув подальшого розвитку метод багатокритеріального вибору оптимальної архітектури системи виявлення шкідливої активності у режимі реального часу, в якому, за рахунок використання послідовного застосування стратегії фільтрації за середніми значеннями та побудови фронту Парето забезпечується баланс між максимальною точністю виявлення шкідливої активності та мінімальними витратами обчислювальних ресурсів. Це дозволило формалізувати процедуру вибору моделей, що забезпечують одночасне покращення показників точності (*Accuracy* > 0,978), повноти (*F1-score* > 0,963) та швидкодії (час відгуку < 7,28 мс), мінімізуючи ризик вибору суб'єктивно «вдалих» конфігурацій».

Висновки до розділу 3

1. Удосконалено метод комплексної оптимізації вхідного набору даних методу виявлення шкідливої активності в інформаційній системі, який, на відміну від існуючих, поєднує балансування класів (SMOTE), нормалізацію (Min-Max) та зниження розмірності (PCA), що дозволило зменшити обчислювальне навантаження та підвищити точність методу гібридної класифікації.

2. Набув подальшого розвитку метод багатокритеріального вибору оптимальної архітектури системи виявлення шкідливої активності у режимі реального часу, в якому, за рахунок використання послідовного застосування стратегії фільтрації за середніми значеннями та побудови фронту Парето забезпечується баланс між максимальною точністю виявлення шкідливої активності та мінімальними витратами обчислювальних ресурсів.

3. Результати експериментів свідчать про досягнення точності до 98,07%, F1-міри 96,57% та часу прогнозу 7,16 мс, що відповідає вимогам до систем виявлення шкідливої активності, здатних функціонувати в реальному часі. що дозволило об’єктивно оцінити компроміс між точністю та швидкодією.

4. Отримані результати демонструють, що стекінгова модель здатна забезпечити ефективність обраних класифікаторів та може бути практично реалізована у високонавантажених корпоративних середовищах.

РОЗДІЛ 4 ПРАКТИЧНА РЕАЛІЗАЦІЯ РОЗРОБЛЕНОГО МЕТОДУ ВИЯВЛЕННЯ ШКІДЛИВОЇ АКТИВНОСТІ В ІНФОРМАЦІЙНІЙ СИСТЕМІ ОРГАНІЗАЦІЇ НА ОСНОВІ ГІБРИДНОЇ КЛАСИФІКАЦІЇ ТА РОЗРОБКА РЕКОМЕНДАЦІЙ ЩОДО ЙОГО ВПРОВАДЖЕННЯ

4.1. Архітектура системи виявлення шкідливої активності в інформаційній системі організації на основі гібридної класифікації

У попередніх розділах дисертаційної роботи було послідовно розроблено теоретичну модель гібридної класифікації, реалізовано метод на основі стекінг-архітектури та проведено його експериментальну верифікацію на датасеті CSE-CIC-IDS2018 (Розділ 3). Метою даного підрозділу є систематизація отриманих наукових результатів та їх представлення як цілісного програмного рішення, готового до впровадження в інформаційну систему організації.

Узагальнення результатів експериментального дослідження

За підсумками експериментального моделювання, проведеного у третьому розділі, встановлено, що найкращу конфігурацію забезпечує ансамбль базових класифікаторів XGBoost, CatBoost та LightGBM із мета-класифікатором на базі XGBoost. Зазначена конфігурація продемонструвала Accuracy на рівні 0,9807 та F1-score 0,9657 при затримці прогнозування 7,16 мс на один мережевий потік. Застосування комплексного методу попередньої обробки даних (SMOTE, Min-Max нормалізація, PCA-редукція з 80 до 18 компонент) забезпечило приріст Accuracy (найвище значення) на 3,87 % та F1-score (найвище значення) на 5,11 % порівняно з результатами на необробленому датасеті, а найменший час прогнозування скоротився на 76 % – із 27,1 до 6,51 мс.

Додатково було проведено багатокритеріальний відбір оптимальних архітектур із використанням стратегії above-average rule та побудови фронту Парето. Було ідентифіковано три недоміновані конфігурації: перша забезпечує

максимальну точність (Accuracy = 0,9807), друга – збалансовану конфігурацію (Accuracy = 0,9762 при 6,91 мс), третя – мінімальну затримку (6,51 мс). Усі три конфігурації задовольняють вимоги до систем виявлення вторгнень реального часу.

Архітектура програмного рішення

Для практичного впровадження результатів дисертаційного дослідження розроблено архітектуру програмного рішення, яка складається з п’яти функціональних модулів, кожен з яких реалізує окремий етап конвеєра обробки мережевого трафіку [24, 25, 26, 37].

Модуль збору даних відповідає за захоплення мережевого трафіку та формування записів про мережеві потоки. Модуль здійснює агрегацію пакетів та обчислення базових статистичних характеристик кожного потоку. Для високошвидкісних мереж (10 Gbps і більше) рекомендується використання бібліотек захоплення трафіку, що працюють в обхід ядра ОС, зокрема DPDK або PF_RING, що забезпечує обробку пакетів у режимі zero-copy.

Модуль попередньої обробки реалізує комплексний метод оптимізації даних, розроблений та верифікований у підрозділі 3.1. Послідовність операцій включає: очищення від технічних артефактів (NaN, Infinity, від’ємні тривалості потоків); балансування класів засобами алгоритму SMOTE – застосовується виключно під час етапу навчання моделі; Min-Max нормалізацію ознакового простору з використанням параметрів, обчислених на навчальній вибірці; PCA-трансформацію з редукцією до 18 головних компонент, що акумулюють 95 % дисперсії вихідних 80 ознак.

Модуль інженерії ознак реалізує механізми створення додаткових інформативних характеристик, описаних у підрозділі 3.2. Зокрема, це агрегація поведінки трафіку у часових вікнах (1, 5, 10 секунд), обчислення статистичних параметрів (середня кількість пакетів та байтів, дисперсія розмірів, швидкість передачі), а також адаптивне зважування ознак на основі механізмів зворотного зв’язку від класифікатора.

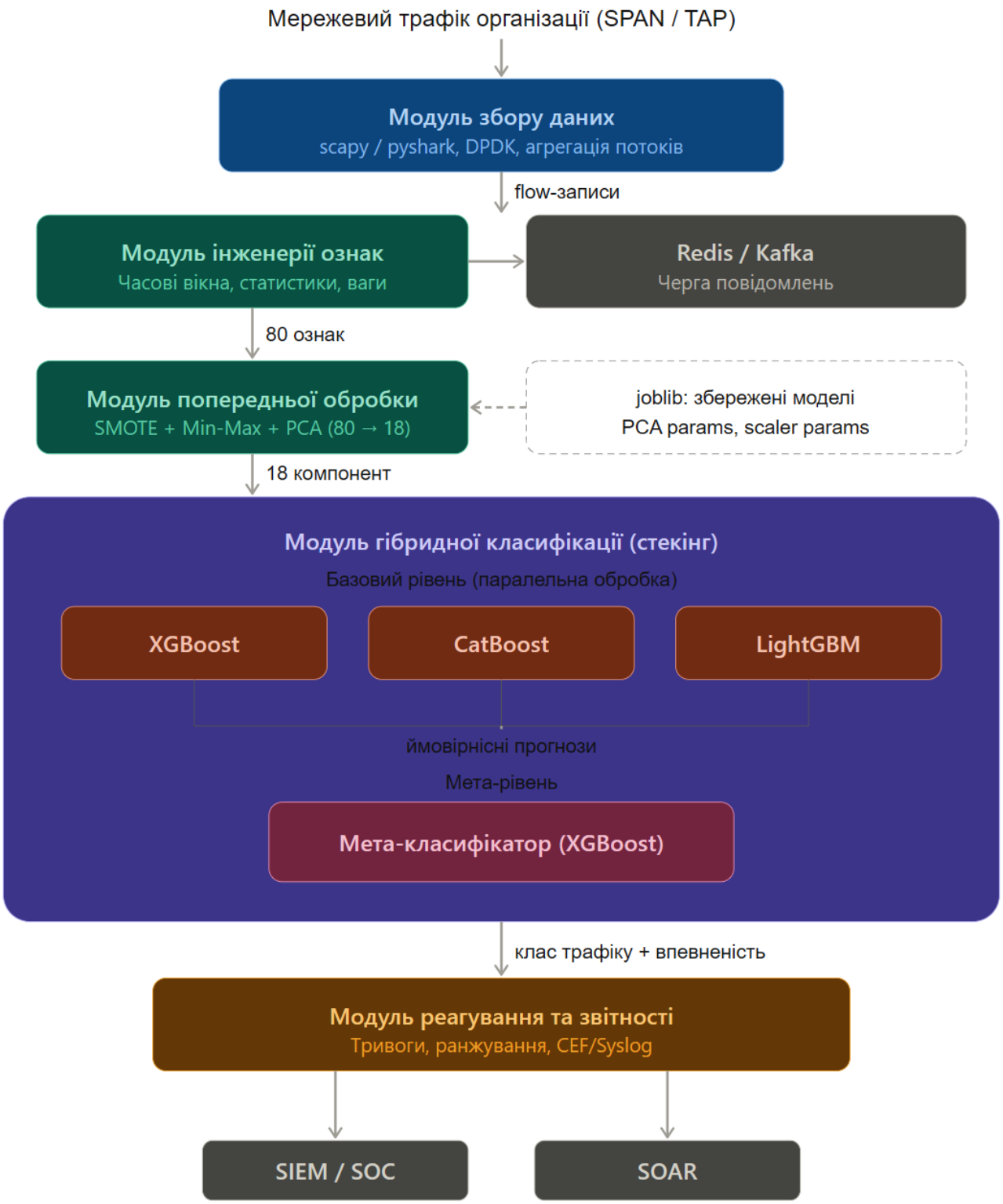

Рис. 4.1. Архітектура програмного рішення для виявлення шкідливої активності на основі гібридної класифікації

Модуль гібридної класифікації є ядром системи та реалізує стекінг-архітектуру, розроблену у другому розділі. Модуль складається з двох рівнів: базового рівня, на якому паралельно функціонують три класифікатори (XGBoost, CatBoost, LightGBM), та мета-рівня, де мета-класифікатор (XGBoost) агрегує

ймовірнісні прогнози базових моделей для формування фінального рішення щодо класу трафіку (легітимний або конкретний тип атаки).

Модуль реагування та звітності забезпечує формування тривожних сповіщень, ранжування інцидентів за рівнем критичності та передачу результатів класифікації до зовнішніх систем – SIEM-платформ, SOC-центрів або автоматизованих систем реагування (SOAR). Модуль також забезпечує логування всіх подій для подальшого аудиту та ретроспективного аналізу.

4.2. Програмна реалізація методу виявлення шкідливої активності в інформаційній системі організації

Програмні залежності та технологічний стек

Програмна реалізація розробленого методу базується на мові програмування Python версії 3.10 або вище, що обумовлено широкою підтримкою бібліотек машинного навчання та аналізу даних. У таблиці 4.1 наведено повний перелік програмних залежностей, необхідних для розгортання системи [37, 47].

Таблиця 4.1.

Програмні залежності для розгортання системи виявлення шкідливої активності

Бібліотека	Версія	Призначення
scikit-learn	≥ 1.3	Реалізація стекінг-ансамблю (StackingClassifier), PCA, Min-Max нормалізації, метрик оцінки якості, крос-валідації
xgboost	≥ 2.0	Базовий та мета-класифікатор XGBoost з підтримкою GPU-прискорення (CUDA)
lightgbm	≥ 4.0	Базовий класифікатор LightGBM, оптимізований для великих обсягів даних

Продовження таблиці 4.1.

Програмні залежності для розгортання системи виявлення шкідливої активності

catboost	≥ 1.2	Базовий класифікатор CatBoost з вбудованою обробкою категоріальних ознак
imbalanced-learn	≥ 0.11	Реалізація алгоритму SMOTE для балансування класів навчальної вибірки
pandas	≥ 2.0	Маніпуляції з табличними даними, агрегація мережевих потоків
numpy	≥ 1.24	Чисельні обчислення, операції з масивами ознак
joblib	≥ 1.3	Серіалізація навчених моделей для збереження та завантаження
flask / fastapi	≥ 2.0 / ≥ .100	REST API для інтеграції модуля класифікації із зовнішніми системами (SIEM, SOAR)
scapy / pyshark	≥ 2.5 /≥ 0.6	Захоплення та парсинг мережевих пакетів, формування flow-записів
docker	≥ 24.0	Контейнеризація компонентів системи для ізольованого розгортання
redis / kafka	≥ 7.0 / ≥ 3.5	Черга повідомлень для буферизації потоків між модулями збору та класифікації

Операційне середовище передбачає використання ОС Ubuntu Server 22.04 LTS або вище, що забезпечує стабільну підтримку всіх зазначених бібліотек та інструментів контейнеризації. Для прискорення етапу навчання моделей рекомендується наявність графічного прискорювача NVIDIA з підтримкою CUDA 11.8+ [103].

Алгоритм функціонування програмного рішення

Алгоритм роботи розробленого програмного рішення в режимі інференсу (класифікації в реальному часі) складається з наступних етапів (рис.4.2):

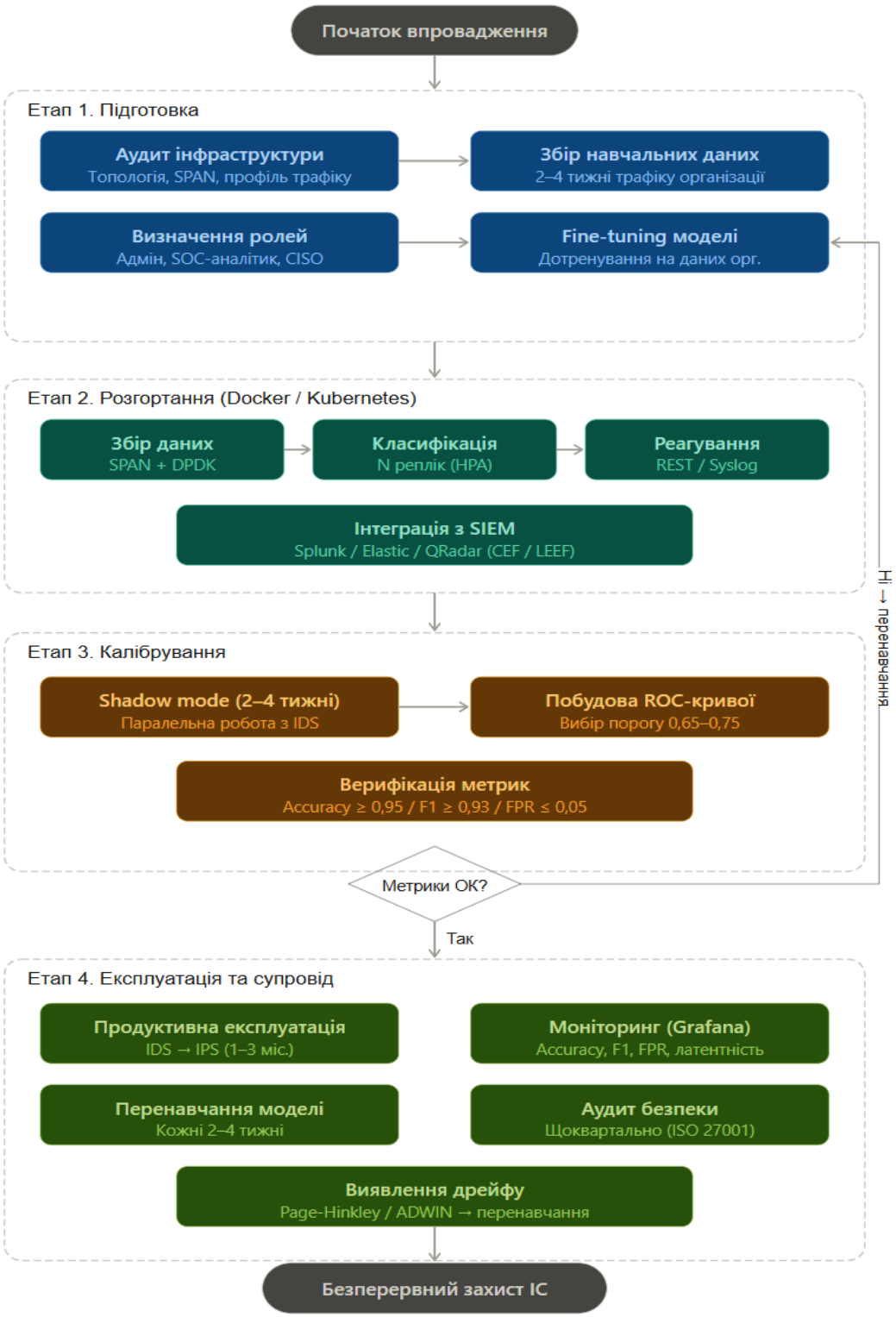

Рис. 4.2. Алгоритм впровадження парограмної реалізації методу виявлення шкідливої активності в інформаційну систему організації

Етап 1. Захоплення трафіку. Модуль збору даних безперервно приймає мережеві пакети з мережевого інтерфейсу та здійснює їх агрегацію у мережеві потоки за п’ятіркою (5-tuple). Сформовані записи потоків надсилаються до черги повідомлень (Redis або Apache Kafka) для подальшої обробки.

Етап 2. Формування вектора ознак. Модуль інженерії ознак вилучає записи з черги та обчислює розширений набір характеристик: статистичні параметри у часових вікнах, темпоральні та поведінкові ознаки. Результатом є вектор з 80 первинних ознак для кожного потоку.

Етап 3. Попередня обробка. Модуль попередньої обробки застосовує Min-Max нормалізацію з використанням збережених параметрів (min та max кожної ознаки, обчислених на етапі навчання) та PCA-трансформацію, що перетворює 80-вимірний вектор у 18-вимірний вектор головних компонент.

Етап 4. Класифікація. Оброблений вектор одночасно подається на три базові класифікатори (XGBoost, CatBoost, LightGBM), які формують ймовірнісні прогнози належності потоку до кожного з класів (Benign, DoS, DDoS, Brute Force, Web Attack, Botnet, Infiltration тощо). Прогнози базових моделей конкатенуються та подаються на вхід мета-класифікатора (XGBoost), який формує фінальне рішення.

Етап 5. Реагування. Якщо мета-класифікатор ідентифікує потік як шкідливий із ймовірністю, що перевищує встановлений поріг (рекомендоване значення 0,65–0,75), система генерує тривожне сповіщення із зазначенням типу атаки, рівня впевненості та контекстної інформації (IP-адреси, порти, час). Сповіщення передається до SIEM-системи або SOC-центру.

Вимоги до апаратного забезпечення

На основі результатів оцінки масштабованості, проведеної у Розділі 3, сформовано мінімальні та рекомендовані вимоги до апаратного забезпечення для організацій різного масштабу. Результати наведено у таблиці 4.2.

Таблиця 4.2.

Вимоги до апаратного забезпечення залежно від масштабу організації

Параметр	Мала організація (до 250 користувачів)	Середня організація (250–1000 користувачів)	Велика організація (1000+ користувачів)
Очікуване навантаження (потоків/с)	до 5 000	5 000–25 000	25 000–50 000+
CPU	16–32 ядра (Intel Xeon / AMD EPYC)	80–120 ядер (кластер)	200–350 ядер (кластер)
RAM	32–64 ГБ	128–256 ГБ	512 ГБ+
GPU (для навчання)	NVIDIA RTX 4070 (8 ГБ)	NVIDIA A30 (24 ГБ)	NVIDIA A100 (40–80 ГБ)
Сховище	500 ГБ SSD	2 ТБ NVMe SSD	10 ТБ+ NVMe SSD (RAID)
Мережевий інтерфейс	1 Gbps	10 Gbps	10–25 Gbps (DPDK)

Зазначені вимоги ґрунтуються на експериментально встановленому показнику затримки 7,16 мс на один потік для оптимальної конфігурації стекінг-моделі. Для паралельної обробки використовується горизонтальне масштабування: кожне CPU-ядро обслуговує окремий екземпляр класифікаційного конвеєра, що забезпечує лінійне зростання пропускної здатності.

Методика оцінки ефективності рішення в умовах реальної експлуатації [102]

Для об’єктивної оцінки ефективності розробленого методу після його розгортання в інформаційній системі організації пропонується методика, що

включає три групи показників: операційні метрики якості детекції, метрики продуктивності та організаційні метрики.

Операційні метрики якості детекції передбачають безперервний моніторинг показників Accuracy, Precision, Recall та F1-score на реальному трафіку організації. Для цього формується валідаційна вибірка шляхом ручної розмітки випадкової підмножини тривожних сповіщень аналітиками SOC-центру. Рекомендується проводити таку розмітку щотижнево для вибірки обсягом не менше 500 записів. Очікувані цільові значення для реального середовища: Accuracy $\geq 0{,}95$; F1-score $\geq 0{,}93$; False Positive Rate $\leq 0{,}05$.

Метрики продуктивності включають вимірювання середньої затримки обробки одного потоку (цільове значення ≤ 10 мс), пропускної здатності системи (кількість потоків на секунду), відсотка втрачених пакетів та рівня завантаження CPU. Збір цих метрик здійснюється автоматизовано за допомогою інструментів моніторингу (Prometheus, Grafana) та інтегрується в єдину панель управління.

Організаційні метрики визначають практичний ефект від впровадження системи та включають: середній час виявлення інциденту (Mean Time to Detect, MTTD); середній час реагування на інцидент (Mean Time to Respond, MTTR); кількість підтверджених інцидентів, виявлених системою за певний період; відсоток скорочення хибних тривог порівняно з попереднім рішенням (якщо таке існувало).

Для порівняння ефективності розробленого методу з існуючими рішеннями рекомендується провести паралельну експлуатацію (shadow mode) протягом 2–4 тижнів, під час якої система працює паралельно з наявними засобами захисту без автоматичного блокування трафіку, а результати класифікації порівнюються з рішеннями існуючих IDS/IPS.

4.3. Розробка рекомендацій щодо застосування методу гібридної класифікації в системах виявлення шкідливої активності

На основі результатів дисертаційного дослідження та розробленої архітектури програмного рішення сформовано комплекс рекомендацій щодо застосування методу гібридної класифікаці виявлення шкідливої активності в інформаційній системі організації. Рекомендації охоплюють етапи підготовки, розгортання, налаштування та подальшої підтримки системи [97].

Eman 1. Підготовка до впровадження

Перед розгортанням системи необхідно провести комплекс підготовчих заходів, що включають аудит поточної мережевої інфраструктури та інвентаризацію існуючих засобів захисту. Рекомендується виконати наступні дії:

Провести інвентаризацію мережевої топології організації, визначити ключові точки збору трафіку (core-комутатори, периметрові маршрутизатори, DMZ-сегменти). Ідентифікувати типовий профіль мережевого трафіку організації: середній та піковий обсяг потоків на секунду, розподіл протоколів, географію трафіку. Ці дані є необхідними для коректного визначення конфігурації апаратного забезпечення відповідно до таблиці 4.2.

Сформувати первинний набір навчальних даних. Для цього рекомендується збирати мережевий трафік організації протягом 2–4 тижнів із забезпеченням покриття всіх типових робочих сценаріїв (робочі дні, вихідні, пікові навантаження). Зібрані дані використовуються для дотренування (fine-tuning) моделі, попередньо навченої на датасеті CSE-CIC-IDS2018, що дозволяє адаптувати класифікатор до специфіки трафіку конкретної організації.

Визначити відповідальних осіб та розподілити ролі: адміністратор системи (розгортання, оновлення моделей), SOC-аналітик (верифікація тривог, ручна розмітка даних), керівник служби інформаційної безпеки (визначення порогових значень та політик реагування).

Eman 2. Розгортання системи

Розгортання рекомендується здійснювати з використанням технології контейнеризації Docker та оркестратора Kubernetes, що забезпечує ізольованість компонентів, зручність масштабування та відмовостійкість. Кожен функціональний модуль системи розгортається як окремий контейнер (мікросервіс):

Контейнер модуля збору даних встановлюється на сервері, підключеному до SPAN-порту (порту дзеркалювання) або мережевого TAP-пристрою ключових сегментів мережі. Для мереж із пропускною здатністю 10 Gbps і вище рекомендується використання DPDK-сумісних мережевих адаптерів (Intel X710, Mellanox ConnectX-5) із конфігурацією hugepages для виділення пам’яті в обхід ядра ОС.

Контейнери модулів попередньої обробки та класифікації розгортаються на обчислювальному кластері. Кількість реплік визначається інтенсивністю трафіку: для навантаження 10 000 потоків/с достатньо 4–6 реплік класифікаційного модуля при використанні 16-ядерного серверного процесора. Kubernetes Horizontal Pod Autoscaler забезпечує автоматичне масштабування кількості реплік залежно від поточного навантаження.

Контейнер модуля реагування інтегрується з існуючою SIEM-платформою організації (Splunk, Elastic SIEM, IBM QRadar) через REST API або протокол Syslog (RFC 5424). Тривожні сповіщення передаються у форматі CEF (Common Event Format) або LEEF, що є стандартними для більшості SIEM-платформ [99].

Eman 3. Налаштування та калібрування

Після розгортання системи необхідно провести її калібрування для конкретного середовища організації. Ключові параметри, що підлягають налаштуванню, наведено у таблиці 4.3.

Калібрування порогу спрацювання рекомендується проводити на основі ROC-кривої, побудованої за результатами класифікації реального трафіку організації протягом першого тижня експлуатації у режимі shadow mode. Оптимальне значення порогу обирається в точці, яка мінімізує загальну вартість

помилок з урахуванням того, що вартість пропущеної атаки (False Negative) для критичних систем суттєво перевищує вартість хибної тривоги (False Positive).

Таблиця 4.3.

Основні параметри калібрування системи

Параметр	Рекомендоване значення	Обґрунтування
Поріг спрацювання мета-класифікатора	0,65–0,75	Забезпечує оптимальний баланс між Precision та Recall. Нижчі значення збільшують кількість хибних тривог, вищі – ризик пропуску атак
Період перенавчання моделі	2–4 тижні	Забезпечує адаптацію до еволюції мережевого трафіку та нових типів загроз без надмірного навантаження на обчислювальні ресурси
Розмір часового вікна агрегації	5–10 с	Оптимальний для виявлення DDoS та Brute Force атак. Менші вікна збільшують навантаження, більші – знижують оперативність виявлення
Частка зразків для активного навчання	5–10 % від тривог	Зразки з найбільшою невизначеністю класифікатора передаються на ручну розмітку аналітикам SOC
Ліміт збереження логів	90–180 днів	Відповідає вимогам стандартів ISO/IEC 27001 та NIST CSF щодо збереження журналів подій безпеки

Етап 4. Підтримка та супровід

Для забезпечення стабільної та ефективної роботи системи в довгостроковій перспективі рекомендується виконувати наступні регулярні процедури.

Перенавчання моделі слід проводити не рідше одного разу на 2–4 тижні з використанням нових розмічених даних. Процедура перенавчання реалізується інкрементально (без повного перенавчання з нуля) шляхом дотренування на оновленій вибірці, що включає нещодавно зібраний та розмічений трафік. Це дозволяє моделі адаптуватися до концептуального дрейфу – зміни статистичних характеристик трафіку внаслідок еволюції як легітимної активності, так і тактик зловмисників.

Моніторинг якості детекції передбачає щотижневий аналіз ключових метрик (Accuracy, F1-score, FPR) на основі розмічених вибірок. У разі падіння F1-score нижче порогового значення 0,93 необхідно ініціювати позачергове перенавчання моделі або перегляд складу ознак. Для автоматичного виявлення деградації якості рекомендується використання статистичних тестів виявлення дрейфу (Page-Hinkley test, ADWIN).

Оновлення програмних компонентів здійснюється відповідно до релізного циклу використовуваних бібліотек. Критичні оновлення безпеки встановлюються невідкладно, планові оновлення – у рамках щомісячного вікна технічного обслуговування. Перед оновленням обов’язково проводиться регресійне тестування моделі на контрольній вибірці.

Аудит системи рекомендується проводити щоквартально із залученням зовнішніх експертів або внутрішньої служби аудиту інформаційної безпеки. Аудит включає перевірку коректності конфігурації, актуальності навчальних даних, відповідності вимогам стандартів ISO/IEC 27001 та NIST Cybersecurity Framework [94, 95].

Схема інтеграції у мережеву інфраструктуру організації

Розроблене програмне рішення рекомендується розташовувати після мережевого екрану організації, що забезпечує аналіз трафіку, який вже пройшов первинну фільтрацію. Типова схема інтеграції передбачає підключення модуля

збору даних до SPAN-порту core-комутатора, що дозволяє отримувати копію всього внутрішнього трафіку без впливу на продуктивність мережі. Для організацій із розподіленою мережевою інфраструктурою (філії, хмарні сегменти) рекомендується розгортання окремих екземплярів модуля збору даних у кожному сегменті з централізованою агрегацією результатів класифікації [96].

Режим інтеграції визначається відповідно до потреб організації. У пасивному режимі система здійснює моніторинг копії трафіку та формує тривожні сповіщення без впливу на мережевий потік. В активному режимі система встановлюється в розрив мережевого каналу та має змогу автоматично блокувати підозрілі потоки. Пасивний режим рекомендується на етапі впровадження та калібрування (перші 1–3 місяці), після чого можливий перехід до активного режима для критичних сегментів мережі.

Єфект від впровадження

На основі результатів експериментального дослідження та аналізу практик впровадження систем виявлення шкідливої активності у корпоративних середовищах, очікуваний ефект від впровадження розробленого методу включає:

Зменшення середнього часу виявлення інцидентів безпеки (MTTD) на 40–60 % порівняно з ручним моніторингом або сигнатурними системами, що обумовлено автоматизацією аналізу та здатністю методу виявляти раніше невідомі типи атак.

Зниження навантаження на аналітиків SOC-центру через скорочення кількості хибних тривог на 20–35 %, що підтверджується результатами порівняння F1-score окремих класифікаторів (0,82–0,94) та гібридної моделі (0,9657).

Підвищення оперативності реагування на складні багатоетапні атаки (APT, multistage intrusions) завдяки здатності стекінг-архітектури враховувати складні нелінійні залежності між ознаками трафіку, які не піддаються виявленню окремими класифікаторами.

Забезпечення обробки трафіку в режимі, наближеному до реального часу, із затримкою не більше 7,16 мс на потік, що дозволяє ефективно захищати мережі з пропускною здатністю до 10 Gbps без помітного впливу на продуктивність.

Висновки до розділу 4

У четвертому розділі дисертаційної роботи проведено систематизацію результатів експериментального дослідження методу виявлення шкідливої активності на основі гібридної класифікації та розроблено комплекс рекомендацій щодо його практичного впровадження в інформаційну систему організації.

1. Систематизовано результати експериментальної верифікації гібридного методу та представлено їх у вигляді цілісного програмного рішення. Оптимальна конфігурація стекінг-ансамблю (XGBoost + CatBoost + LightGBM з мета-класифікатором XGBoost) яка була обрана за допомогою послідовного застосування стратегії фільтрації за середніми значеннями (*above-average rule*) та побудови фронту Парето та забезпечує Accuracy 0,9807, F1-score 0,9657 та затримку прогнозування 7,16 мс, що підтверджує придатність методу для застосування в режимі реального часу.

2. Розроблено п’ятимодульну архітектуру програмного рішення, що включає модулі збору даних, попередньої обробки, інженерії ознак, гібридної класифікації та реагування. Визначено повний перелік програмних залежностей scikit-learn, xgboost, lightgbm, catboost, imbalanced-learn та сформовано диференційовані вимоги до апаратного забезпечення для організації.

3. Розроблено рекомендації оцінки ефективності розробленого методу в умовах реальної експлуатації, що включає операційні метрики якості детекції, метрики продуктивності та організаційні метрики (MTTD, MTTR). Рекомендовано проведення паралельної експлуатації протягом 2–4 тижнів для об’єктивного порівняння з існуючими засобами захисту.

4. Розроблено комплекс рекомендацій щодо впровадження, що охоплює чотири етапи: підготовку інфраструктури та формування навчальних даних, контейнеризоване розгортання з використанням Docker/Kubernetes, калібрування параметрів (поріг спрацювання 0,65–0,75, період перенавчання 2–4 тижні), процедури довгострокової підтримки та супроводу.

ВИСНОВКИ

У дисертаційній роботі вирішено важливе науково-прикладне завдання щодо підвищення ефективності захисту інформаційних систем організацій шляхом розробки та впровадження нових методів виявлення шкідливої активності на основі гібридної класифікації. Отримані результати свідчать про досягнення поставленої мети та дозволяють зробити такі висновки:

1. На підставі проведеного комплексного аналізу встановлено, що сучасний стан розвитку інформаційно-комунікаційних технологій супроводжується експоненційним зростанням кількості та складності кіберзагроз, спрямованих на інформаційні системи організацій. За результатами дослідження аналітичних звітів (Verizon DBIR 2024, IBM X-Force 2025, ENISA Threat Landscape 2024) встановлено, що середній час виявлення та стримування витоку даних становить 204-241 день, що засвідчує неспроможність значної частини організацій своєчасно реагувати на сучасні загрози. Традиційні методи виявлення шкідливої активності – сигнатурні системи, поведінкові евристики та окремі класифікатори машинного навчання – не забезпечують необхідного рівня точності класифікації загроз у реальному часі при допустимому рівні хибних спрацювань. Виявлені протиріччя між необхідністю виявлення нових типів атак у режимі реального часу та статичною природою існуючих засобів захисту, а також між точністю детектування та швидкістю прийняття рішень, обумовлюють актуальність розробки методів виявлення шкідливої активності на основі гібридної класифікації, що поєднують переваги різних підходів.

2. Вперше розроблено метод виявлення шкідливої активності в інформаційній системі організації на основі гібридної класифікації, який базується на використанні гетерогенного набору базових класифікаторів та мета-класифікаторі на основі XGBoost для дворівневої стекінг-архітектури, що дає можливість застосовувати різнорідні ансамблеві методи машинного навчання у системах виявлення шкідливої активності в режимі реального часу з можливістю динамічного переналаштування класифікаторів.

3. Удосконалено метод комплексної оптимізації вхідного набору даних методу виявлення шкідливої активності в інформаційній системі, який, на відміну від існуючих, поєднує балансування класів (SMOTE), нормалізацію (Min-Max) та зниження розмірності (PCA), що дозволило зменшити обчислювальне навантаження та підвищити точність методу гібридної класифікації.

4. Набув подальшого розвитку метод багатокритеріального вибору оптимальної архітектури системи виявлення шкідливої активності у режимі реального часу, в якому, за рахунок використання послідовного застосування стратегії фільтрації за середніми значеннями та побудови фронту Парето забезпечується баланс між максимальною точністю виявлення шкідливої активності та мінімальними витратами обчислювальних ресурсів.

5. У результаті проведення експериментального дослідження розробленого методу на датасеті CSE-CIC-IDS2018 встановлено його високу ефективність у виявленні різнорідних типів атак. Порівняльний аналіз показав, що гібридний метод перевершує окремі класифікатори: приріст Accuracy складає 3,87%, F1-score – 5,11%, при одночасному скороченні часу прогнозування на 76%. Методика багатокритеріального відбору дозволила ідентифікувати три недоміновані конфігурації на фронту Парето, що задовольняють вимоги до систем виявлення вторгнень реального часу з максимальною точністю на рівні 0,9807 та мінімальною затримкою 7,16 мс.

6. Розроблено комплекс рекомендацій щодо впровадження методу виявлення шкідливої активності в інформаційну систему організації. Рекомендовано чотирьох-етапну схему впровадження: підготовка інфраструктури та формування навчальних даних; контейнеризоване розгортання з використанням Docker/Kubernetes; калібрування параметрів системи з оптимальним поріг спрацювання 0,65–0,75 та періодом перенавчання 2–4 тижні; процедури довгострокової підтримки та супроводу з періодичним оновленням моделей. Визначено диференційовані вимоги до апаратного забезпечення для організацій різного масштабу та описано алгоритм функціонування розробленого програмного рішення.

7. Основні результати дисертаційного дослідження успішно впроваджено в діяльність ТОВ «Євротелеком» та ТОВ «АРВІОМ», де вони використовуються для удосконалення систем моніторингу безпеки та виявлення шкідливої активності в реальному часі. Результати також знайшли застосування у навчальному процесі Державного університету інформаційно-комунікаційних технологій в межах освітніх компонент з організації наукових досліджень та штучного інтелекту.

8. Таким чином, поставлена мета дослідження – підвищення точності виявлення шкідливої активності в інформаційній системі організації на основі гібридної класифікації – досягнута повністю. Усі часткові завдання вирішено комплексно та послідовно, розроблені методи мають чітке наукове обґрунтування та підтверджену практичну цінність. Отримані результати становлять корисний внесок у розвиток науково-методичних основ, технологій та засобів виявлення кіберзагроз в інформаційних системах організацій.

9. Напрями подальших досліджень у зазначеній галузі мають ґрунтуватися на: розширенні методів машинного навчання та впровадженні архітектур глибокого навчання для виявлення нових, раніше невідомих типів атак; розробці механізмів адаптивного навчання для автоматичної адаптації моделей до концептуального дрейфу мережевого трафіку; впровадженні методів пояснюваного штучного інтелекту для підвищення довіри до системи та спрощення розслідування інцидентів; оптимізації архітектури стекінгу для забезпечення масштабованості на гіперконвергованих хмарних платформах; розробці методів активного навчання для мінімізації витрат на ручну розмітку даних експертами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Verizon. 2024 Data Breach Investigations Report. Verizon Business, 2024. URL: https://www.verizon.com/business/resources/reports/dbir/.

2. IBM Security. X-Force Threat Intelligence Index 2024. IBM Corporation, 2024. URL: https://www.ibm.com/reports/threat-intelligence.

3. ENISA. ENISA Threat Landscape 2024. European Union Agency for Cybersecurity, 2024. URL: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2024.

4. Ho C.-Y., Lin Y.-D., Lai Y.-C., Chen I.-W., Wang F.-Y., Tai W.-H. False Positives and Negatives from Real Traffic with Intrusion Detection/Prevention Systems. International Journal of Future Computer and Communication, 2012, vol. 1, no. 2, pp. 87–90. DOI: 10.7763/IJFCC.2012.V1.23.

5. Sarker I.H., Kayes A.S.M., Badsha S., Alqahtani H., Watters P., Ng A. Cybersecurity data science: an overview from machine learning perspective. Journal of Big Data, 2020, vol. 7, no. 1, pp. 1–29. DOI: 10.1186/s40537-020-00318-5.

6. Bertino E., Islam N. Botnets and Internet of Things Security. Computer, 2017, vol. 50, no. 2, pp. 76–79. DOI: 10.1109/MC.2017.62.

7. IBM Security. Cost of a Data Breach Report 2024. IBM Corporation, 2024. URL: https://www.ibm.com/reports/data-breach.

8. Wolpert D.H. Stacked Generalization. Neural Networks, 1992, vol. 5, no. 2, pp. 241–259. DOI: 10.1016/S0893-6080(05)80023-1.

9. Buczak A.L., Guven E. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. IEEE Communications Surveys & Tutorials, 2016, vol. 18, no. 2, pp. 1153–1176. DOI: 10.1109/COMST.2015.2494502.

10. Khraisat A., Gondal I., Vamplew P., Kamruzzaman J. Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity, 2019, vol. 2, no. 1, pp. 1–22. DOI: 10.1186/s42400-019-0038-7.

11. Denning D.E. An Intrusion-Detection Model. IEEE Transactions on Software Engineering, 1987, vol. SE-13, no. 2, pp. 222–232. DOI: 10.1109/TSE.1987.232894.

12. Anderson J.P. Computer Security Threat Monitoring and Surveillance. Technical Report, James P. Anderson Co., Fort Washington, PA, 1980.

13. Lunt T.F. A survey of intrusion detection techniques. Computers & Security, 1993, vol. 12, no. 4, pp. 405–418. DOI: 10.1016/0167-4048(93)90029-5.

14. Axelsson S. Intrusion Detection Systems: A Survey and Taxonomy. Technical Report 99-15, Chalmers University of Technology, 2000.

15. Garcia-Teodoro P., Diaz-Verdejo J., Macia-Fernandez G., Vazquez E. Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers & Security, 2009, vol. 28, no. 1–2, pp. 18–28. DOI: 10.1016/j.cose.2008.08.003.

16. Sommer R., Paxson V. Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. Proceedings of the 2010 IEEE Symposium on Security and Privacy, 2010, pp. 305–316. DOI: 10.1109/SP.2010.25.

17. Paxson V. Bro: A System for Detecting Network Intruders in Real-Time. Computer Networks, 1999, vol. 31, no. 23–24, pp. 2435–2463. DOI: 10.1016/S1389-1286(99)00112-7.

18. Roesch M. Snort – Lightweight Intrusion Detection for Networks. Proceedings of the 13th USENIX Conference on System Administration, 1999, pp. 229–238.

19. Liao H.-J., Lin C.-H.R., Lin Y.-C., Tung K.-Y. Intrusion detection system: A comprehensive review. Journal of Network and Computer Applications, 2013, vol. 36, no. 1, pp. 16–24. DOI: 10.1016/j.jnca.2012.09.004.

20. Chandola V., Banerjee A., Kumar V. Anomaly detection: A survey. ACM Computing Surveys, 2009, vol. 41, no. 3, pp. 1–58. DOI: 10.1145/1541880.1541882.

21. Cortes C., Vapnik V. Support-vector networks. Machine Learning, 1995, vol. 20, no. 3, pp. 273–297. DOI: 10.1007/BF00994018.

22. Breiman L. Random Forests. Machine Learning, 2001, vol. 45, no. 1, pp. 5–32. DOI: 10.1023/A:1010933404324.

23. Cover T.M., Hart P.E. Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 1967, vol. 13, no. 1, pp. 21–27. DOI: 10.1109/TIT.1967.1053964.

24. Chen T., Guestrin C. XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794. DOI: 10.1145/2939672.2939785.

25. Ke G., Meng Q., Finley T., Wang T., Chen W., Ma W., Ye Q., Liu T.-Y. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. Advances in Neural Information Processing Systems 30, 2017, pp. 3146–3154.

26. Prokhorenkova L., Gusev G., Vorobev A., Dorogush A.V., Gulin A. CatBoost: unbiased boosting with categorical features. Advances in Neural Information Processing Systems 31, 2018, pp. 6639–6649.

27. Friedman J.H. Greedy Function Approximation: A Gradient Boosting Machine. The Annals of Statistics, 2001, vol. 29, no. 5, pp. 1189–1232. DOI: 10.1214/aos/1013203451.

28. Breiman L. Bagging Predictors. Machine Learning, 1996, vol. 24, no. 2, pp. 123–140. DOI: 10.1007/BF00058655.

29. Freund Y., Schapire R.E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. Journal of Computer and System Sciences, 1997, vol. 55, no. 1, pp. 119–139. DOI: 10.1006/jcss.1997.1504.

30. Hosmer D.W., Lemeshow S., Sturdivant R.X. Applied Logistic Regression, 3rd ed. Hoboken, NJ: John Wiley & Sons, 2013. 528 p. ISBN: 978-0-470-58247-3.

31. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning, 2nd ed. New York: Springer, 2009. 745 p. DOI: 10.1007/978-0-387-84858-7.

32. Bishop C.M. Pattern Recognition and Machine Learning. New York: Springer, 2006. 738 p. ISBN: 978-0-387-31073-2.

33. Goodfellow I., Bengio Y., Courville A. Deep Learning. Cambridge, MA: MIT Press, 2016. 800 p. ISBN: 978-0-262-03561-3.

34. Zhou Z.-H. Ensemble Methods: Foundations and Algorithms. Boca Raton: CRC Press, 2012. 236 p. DOI: 10.1201/b12207.

35. Rokach L. Ensemble-based classifiers. Artificial Intelligence Review, 2010, vol. 33, no. 1–2, pp. 1–39. DOI: 10.1007/s10462-009-9124-7.

36. Dietterich T.G. Ensemble Methods in Machine Learning. Lecture Notes in Computer Science, 2000, vol. 1857, pp. 1–15. DOI: 10.1007/3-540-45014-9_1.

37. Pedregosa F., Varoquaux G., Gramfort A. et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 2011, vol. 12, pp. 2825–2830.

38. Ting K.M., Witten I.H. Issues in Stacked Generalization. Journal of Artificial Intelligence Research, 1999, vol. 10, pp. 271–289. DOI: 10.1613/jair.594.

39. Džeroski S., Ženko B. Is Combining Classifiers with Stacking Better than Selecting the Best One? Machine Learning, 2004, vol. 54, no. 3, pp. 255–273. DOI: 10.1023/B:MACH.0000015878.60765.42.

40. Quinlan J.R. Induction of Decision Trees. Machine Learning, 1986, vol. 1, no. 1, pp. 81–106. DOI: 10.1007/BF00116251.

41. Chawla N.V., Bowyer K.W., Hall L.O., Kegelmeyer W.P. SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 2002, vol. 16, pp. 321–357. DOI: 10.1613/jair.953.

42. Jolliffe I.T. Principal Component Analysis, 2nd ed. New York: Springer, 2002. 487 p. DOI: 10.1007/b98835.

43. He H., Garcia E.A. Learning from Imbalanced Data. IEEE Transactions on Knowledge and Data Engineering, 2009, vol. 21, no. 9, pp. 1263–1284. DOI: 10.1109/TKDE.2008.239.

44. Fernández A., García S., Galar M., Prati R.C., Krawczyk B., Herrera F. Learning from Imbalanced Data Sets. Cham: Springer, 2018. 377 p. DOI: 10.1007/978-3-319-98074-4.

45. Han J., Kamber M., Pei J. Data Mining: Concepts and Techniques, 3rd ed. Waltham, MA: Morgan Kaufmann, 2012. 744 p. ISBN: 978-0-12-381479-1.

46. Shlens J. A Tutorial on Principal Component Analysis. arXiv preprint arXiv:1404.1100, 2014.

47. Lemaître G., Nogueira F., Aridas C.K. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. Journal of Machine Learning Research, 2017, vol. 18, no. 17, pp. 1–5.

48. Japkowicz N., Stephen S. The class imbalance problem: A systematic study. Intelligent Data Analysis, 2002, vol. 6, no. 5, pp. 429–449. DOI: 10.3233/IDA-2002-6504.

49. García S., Luengo J., Herrera F. Data Preprocessing in Data Mining. Cham: Springer, 2015. 320 p. DOI: 10.1007/978-3-319-10247-4.

50. Abdi H., Williams L.J. Principal component analysis. Wiley Interdisciplinary Reviews: Computational Statistics, 2010, vol. 2, no. 4, pp. 433–459. DOI: 10.1002/wics.101.

51. Guyon I., Elisseeff A. An Introduction to Variable and Feature Selection. Journal of Machine Learning Research, 2003, vol. 3, pp. 1157–1182.

52. Van der Maaten L., Hinton G. Visualizing data using t-SNE. Journal of Machine Learning Research, 2008, vol. 9, pp. 2579–2605.

53. Kotsiantis S., Kanellopoulos D., Pintelas P. Data preprocessing for supervised learning. International Journal of Computer Science, 2006, vol. 1, no. 2, pp. 111–117.

54. Prati R.C., Batista G.E., Monard M.C. Data mining with imbalanced class distributions: concepts and methods. Proceedings of the Indian International Conference on Artificial Intelligence, 2009, pp. 359–376.

55. Brownlee J. Data Preparation for Machine Learning. Machine Learning Mastery, 2020. ISBN: 979-8-6489-1854-9.

56. Sharafaldin I., Lashkari A.H., Ghorbani A.A. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP), 2018, pp. 108–116. DOI: 10.5220/0006639801080116.

57. Communications Security Establishment (CSE), Canadian Institute for Cybersecurity (CIC). CSE-CIC-IDS2018 Dataset. 2018. URL: https://www.unb.ca/cic/datasets/ids-2018.html.

58. Tavallaee M., Bagheri E., Lu W., Ghorbani A.A. A Detailed Analysis of the KDD CUP 99 Data Set. Proceedings of the 2009 IEEE Symposium on CISDA, 2009, pp. 1–6. DOI: 10.1109/CISDA.2009.5356528.

59. Moustafa N., Slay J. UNSW-NB15: a comprehensive data set for network intrusion detection systems. Military Communications and Information Systems Conference (MilCIS), 2015, pp. 1–6. DOI: 10.1109/MilCIS.2015.7348942.

60. Leevy J.L., Khoshgoftaar T.M. A survey and analysis of intrusion detection models based on CSE-CIC-IDS2018 Big Data. Journal of Big Data, 2020, vol. 7, no. 104, pp. 1–19. DOI: 10.1186/s40537-020-00382-x.

61. Ring M., Wunderlich S., Scheuring D., Landes D., Hotho A. A survey of network-based intrusion detection data sets. Computers & Security, 2019, vol. 86, pp. 147–167. DOI: 10.1016/j.cose.2019.06.005.

62. Sharafaldin I., Lashkari A.H., Ghorbani A.A. A Detailed Analysis of the CICIDS2017 Dataset. Proceedings of the International Conference on Information Systems Security and Privacy, 2018.

63. Göcs L., Johanyák Z.C. Identifying relevant features of CSE-CIC-IDS2018 dataset for the development of an intrusion detection system. Intelligent Data Analysis, 2023, vol. 27, no. 6, pp. 1639–1662. DOI: 10.3233/IDA-230264.

64. Panigrahi R., Borah S. A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems. International Journal of Engineering and Technology, 2018, vol. 7, no. 3.24, pp. 479–482.

65. Dhaliwal S.S., Nahid A.A., Abbas R. Effective Intrusion Detection System Using XGBoost. Information, 2018, vol. 9, no. 7, p. 149. DOI: 10.3390/info9070149.

66. Ahmad I., Basheri M., Iqbal M.J., Rahim A. Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion

Detection. IEEE Access, 2018, vol. 6, pp. 33789–33795. DOI: 10.1109/ACCESS.2018.2841987.

67. Resende P.A.A., Drummond A.C. A Survey of Random Forest Based Methods for Intrusion Detection Systems. ACM Computing Surveys, 2018, vol. 51, no. 3, pp. 1–36. DOI: 10.1145/3178582.

68. Belouch M., El Hadaj S., Idhammad M. Performance evaluation of intrusion detection based on machine learning using Apache Spark. Procedia Computer Science, 2018, vol. 127, pp. 1–6. DOI: 10.1016/j.procs.2018.01.091.

69. Yin C., Zhu Y., Fei J., He X. A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks. IEEE Access, 2017, vol. 5, pp. 21954–21961. DOI: 10.1109/ACCESS.2017.2762418.

70. Vinayakumar R., Alazab M., Soman K.P. et al. Deep Learning Approach for Intelligent Intrusion Detection System. IEEE Access, 2019, vol. 7, pp. 41525–41550. DOI: 10.1109/ACCESS.2019.2895334.

71. Thockchom N., Singh M.M., Nandi U. A novel ensemble learning-based model for network intrusion detection. Complex & Intelligent Systems, 2023, vol. 9, no. 5, pp. 5693–5714. DOI: 10.1007/s40747-023-01013-7.

72. Kim J., Kim J., Kim H., Shim M., Choi E. CNN-Based Network Intrusion Detection against Denial-of-Service Attacks. Electronics, 2020, vol. 9, no. 6, p. 916. DOI: 10.3390/electronics9060916.

73. Karatas G., Demir O., Sahingoz O.K. Increasing the Performance of Machine Learning-Based IDSs on an Imbalanced and Up-to-Date Dataset. IEEE Access, 2020, vol. 8, pp. 32150–32162. DOI: 10.1109/ACCESS.2020.2973219.

74. Li X., Chen W., Zhang Q., Wu L. Building Auto-Encoder Intrusion Detection System based on random forest feature selection. Computers & Security, 2020, vol. 95, p. 101851. DOI: 10.1016/j.cose.2020.101851.

75. Abdallah E.E., Eleisah W., Otoom A.F. Intrusion Detection Systems using Supervised Machine Learning Techniques: A survey. Procedia Computer Science, 2022, vol. 201, pp. 205–212. DOI: 10.1016/j.procs.2022.03.029.

76. Yang Z., Liu X., Li T. et al. A systematic literature review of methods and datasets for anomaly-based network intrusion detection. Computers & Security, 2022, vol. 116, p. 102675. DOI: 10.1016/j.cose.2022.102675.

77. Ferrag M.A., Maglaras L., Moschoyiannis S., Janicke H. Deep learning for cyber security intrusion detection. Journal of Information Security and Applications, 2020, vol. 50, p. 102419. DOI: 10.1016/j.jisa.2019.102419.

78. Kanimozhi V., Jacob T.P. Artificial Intelligence outflanks all other machine learning classifiers in Network Intrusion Detection System on the realistic cyber dataset CSE-CIC-IDS2018. ICT Express, 2021, vol. 7, no. 3, pp. 366–370. DOI: 10.1016/j.icte.2020.12.004.

79. Le T.T.H., Kim H., Kang H., Kim H. Classification and Explanation for Intrusion Detection System Based on Ensemble Trees and SHAP Method. Sensors, 2022, vol. 22, no. 3, p. 1154. DOI: 10.3390/s22031154.

80. Seth S., Chahal K.K., Grover G. Improving intrusion detection using LightGBM with optimized feature selection and continuous data balancing. Computing and Informatics, 2021, vol. 40, no. 5, pp. 893–912.

81. Farhan R.I., Maolood A.T., Hassan N.F. Performance analysis of flow-based attacks detection on CSE-CIC-IDS2018 dataset using deep learning. Indonesian J. of EE and CS, 2020, vol. 20, no. 3, pp. 1413–1418.

82. Momand A., Jan S.U., Ramzan N. A Systematic and Comprehensive Survey of Recent Advances in Intrusion Detection Systems Using Machine Learning. Journal of Sensors, 2023, vol. 2023, p. 6048087. DOI: 10.1155/2023/6048087.

83. Mohamed A.A. et al. Network Intrusion Detection using Machine Learning Algorithms on CSE-CIC-IDS2018 Dataset. SSRG IJECE, 2024, vol. 11, no. 3, pp. 195–208.

84. Lawal W. et al. A Comparative Analysis of Machine Learning Models for Network Intrusion Detection Using CSE-CIC-IDS2018 Dataset. NIPES JSTR, 2025, vol. 7, pp. 850–858.

85. Gulzar Q., Mustafa K. Resilient cybersecurity: ensemble deep learning and reinforcement learning for Next-Gen IDS. Iran Journal of Computer Science, 2026. DOI: 10.1007/s42044-025-00364-3.

86. Sokolova M., Lapalme G. A systematic analysis of performance measures for classification tasks. Information Processing & Management, 2009, vol. 45, no. 4, pp. 427–437. DOI: 10.1016/j.ipm.2009.03.002.

87. Powers D.M.W. Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation. Journal of Machine Learning Technologies, 2011, vol. 2, no. 1, pp. 37–63.

88. Chicco D., Jurman G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics, 2020, vol. 21, no. 6, pp. 1–13. DOI: 10.1186/s12864-019-6413-7.

89. Davis J., Goadrich M. The Relationship Between Precision-Recall and ROC Curves. Proceedings of ICML, 2006, pp. 233–240. DOI: 10.1145/1143844.1143874.

90. Fawcett T. An introduction to ROC analysis. Pattern Recognition Letters, 2006, vol. 27, no. 8, pp. 861–874. DOI: 10.1016/j.patrec.2005.10.010.

91. Arp D. et al. Dos and Don'ts of Machine Learning in Computer Security. Proceedings of the 31st USENIX Security Symposium, 2022, pp. 3971–3988.

92. Pendlebury F. et al. TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time. Proceedings of the 28th USENIX Security Symposium, 2019, pp. 729–746.

93. Apruzzese G. et al. On the effectiveness of machine and deep learning for cyber security. Proceedings of the 10th International Conference on Cyber Conflict, 2018, pp. 371–390. DOI: 10.23919/CYCON.2018.8405026.

94. NIST. Framework for Improving Critical Infrastructure Cybersecurity, Version 1.1. National Institute of Standards and Technology, 2018. DOI: 10.6028/NIST.CSWP.04162018.

95. ISO/IEC 27001:2022. Information security, cybersecurity and privacy protection – Information security management systems – Requirements. ISO, 2022.

96. Chuvakin A., Schmidt K., Phillips C. Logging and Log Management: The Authoritative Guide. Waltham, MA: Syngress, 2013. 438 p. ISBN: 978-1-59749-635-3.

97. Bhatt S., Manadhata P.K., Zomlot L. The Operational Role of SIEM Systems. IEEE Security & Privacy, 2014, vol. 12, no. 5, pp. 35–41. DOI: 10.1109/MSP.2014.103.

98. Sundaramurthy S.C. et al. A Human Capital Model for Mitigating Security Analyst Burnout. Proceedings of SOUPS, 2015, pp. 347–359.

99. Gama J., Žliobaitė I., Bifet A. et al. A survey on concept drift adaptation. ACM Computing Surveys, 2014, vol. 46, no. 4, pp. 1–37. DOI: 10.1145/2523813.

100. Page E.S. Continuous Inspection Schemes. Biometrika, 1954, vol. 41, no. 1–2, pp. 100–115. DOI: 10.1093/biomet/41.1-2.100.

101. Bifet A., Gavaldà R. Learning from Time-Changing Data with Adaptive Windowing. Proceedings of SIAM International Conference on Data Mining, 2007, pp. 443–448. DOI: 10.1137/1.9781611972771.42.

102. Burns B. et al. Kubernetes: Up and Running, 3rd ed. Sebastopol, CA: O'Reilly Media, 2022. 326 p. ISBN: 978-1-098-11020-8.

103. Merkel D. Docker: lightweight Linux containers for consistent development and deployment. Linux Journal, 2014, vol. 2014, no. 239, p. 2.

104. Leevy J.L., Khoshgoftaar T.M. Detecting cybersecurity attacks across different network features and learners. Journal of Big Data, 2021, vol. 8, no. 38, pp. 1–29. DOI: 10.1186/s40537-021-00426-w.

105. Bouidaine A.B. et al. Deep Learning-Based Anomaly and Intrusion Detection Using the CSE-CIC-IDS2018 Dataset. Eng. Technol. Appl. Sci. Res., 2025, vol. 15, no. 4, pp. 24782–24787.

106. Elhanashi A. et al. Machine Learning Techniques for Anomaly-Based Detection System on CSE-CIC-IDS2018 Dataset. LNEE, 2023, vol. 1036, pp. 131–140. DOI: 10.1007/978-3-031-30333-3_17.

107. Talukder M.A. et al. A dependable hybrid machine learning model for network intrusion detection. JISA, 2023, vol. 72, p. 103405. DOI: 10.1016/j.jisa.2022.103405.

108. Hasan M.A.M. et al. Feature Selection for Intrusion Detection Using Random Forest. Journal of Information Security, 2016, vol. 7, no. 3, pp. 129–140. DOI: 10.4236/jis.2016.73009.

109. Khan F.A. et al. A Novel Two-Stage Deep Learning Model for Efficient Network Intrusion Detection. IEEE Access, 2019, vol. 7, pp. 30373–30385. DOI: 10.1109/ACCESS.2019.2899721.

110. Pham N.T. et al. Improving performance of intrusion detection system using ensemble methods and feature selection. Proceedings of ACSW, 2018, pp. 1–6. DOI: 10.1145/3167918.3167951.

111. Çavuşoğlu Ü. Intrusion Detection on CSE-CIC-IDS2018 Dataset Using Machine Learning Methods. AI Theory and Applications, 2024, vol. 4, no. 2, pp. 145–155.

112. Najafi Mohsenabad H., Tut M.A. Optimizing Cybersecurity Attack Detection: A Comparative Analysis Using the CSE-CIC-IDS 2018 Dataset. Applied Sciences, 2024, vol. 14, no. 3, p. 1044. DOI: 10.3390/app14031044.

113. Гайдур Г.І., Гамза Д.Є. Гібридний метод виявлення шкідливої активності на основі стекінг-ансамблю класифікаторів. Сучасний захист інформації, 2025, № 3(63), С. 20–26. DOI: 10.31673/2409-7292.2025.030315.

114. Савченко В.А., Смолєв Є.С., Гамза Д.Є. Методика виявлення аномалій взаємодії користувачів з інформаційними ресурсами організації. Сучасний захист інформації, 2023, № 4, С. 6–12. DOI: 10.31673/2409-7292.2023.030101.

115. Haidur H., Gakhov S., Hamza D. Using support vectors to build a rule-based system for detecting malicious processes in an organisation's network traffic. IAPGOS, 2024, vol. 14, no. 4, pp. 90–96. DOI: 10.35784/iapgos.6366.

116. Волошко Д.С., Гамза Д.Є., Смолєв Є.С. Технологія виявлення простих вірусів у програмному коді. Сучасний захист інформації, 2023, № 2(54), С. 41–49. DOI: 10.31673/2409-7292.2023.020006.

117. Гамза, Д. (2025). Вплив оптимізації датасету CSE–CIC–IDS2018 на ефективність гібридної стекінгової моделі виявлення мережевих вторгнень.

Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка», 2(30), 766–777. https://doi.org/10.28925/2663-4023.2025.30.963.

118. Friedman J.H. Stochastic gradient boosting. Computational Statistics & Data Analysis, 2002, vol. 38, no. 4, pp. 367–378. DOI: 10.1016/S0167-9473(01)00065-2.

119. Bergstra J., Bengio Y. Random Search for Hyper-Parameter Optimization. Journal of Machine Learning Research, 2012, vol. 13, pp. 281–305.

120. Sarhan M. et al. NetFlow Datasets for Machine Learning-Based Network Intrusion Detection Systems. LNCS, 2021, vol. 12812, pp. 117–135. DOI: 10.1007/978-3-030-72802-1_9.

121. Oliveira N., Praça I., Maia E., Sousa O. False Positive Identification in Intrusion Detection Using XAI. IEEE Access, 2023, vol. 11, pp. 74855–74868. DOI: 10.1109/ACCESS.2023.3292970.

122. Hadi H.J. et al. Reducing False Positives in Intrusion Detection System Alerts: A Novel Aggregation and Correlation Model. In: Goel S., Uzun E., Xie M., Sarkar S. (eds) Digital Forensics and Cyber Crime. ICDF2C 2024. LNICST, vol. 613. Springer, Cham, 2025. DOI: 10.1007/978-3-031-89363-6_9.

123. Гайдур Г.І., Гахов С.О., Гамза Д.Є. Модель виявлення шкідливої активності в інформаційній системі організації на основі гібридної класифікації. Сучасний захист інформації. 2024. № 4(60). С. 30–38. DOI: 10.31673/2409-7292.2024.040003.

124. Haydur H., Hamza D. Hybrid method for malicious activity detection in information systems. Digital Transformation: Strengthening the Cybersecurity Capacities in the Modern World, November 4–5, 2025. P. 39–41.

125. Смолєв Є.С., Гамза Д.Є. Метод оптимізації даних для тренування моделі виявлення вторгнень на основі SVM. Цифрова трансформація кібербезпеки: збірник тез наук.-практ. конф. Київ: РВЦ ДУІКТ, 2024. С. 220–223.

126. Haydur, H., & Hamza, D., Zybin S. (2026, 29 квітня) Multi-criteria selection of optimal hybrid stacking ensemble model for intrusion detection systems: pareto front and above-average rule filtering. Цифрова трансформація кібербезпеки: збірник тез наук.-практ. конф. Київ: РВЦ ДУІКТ. 2026. С. 27-30.

ДОДАТОК А

ARVIOM

ТОВАРИСТВО З ОБМЕЖЕНОЮ ВІДПОВІДАЛЬНІСТЮ «АРВІОМ»

(код ЄДРПОУ 45815341, місцезнаходження: 01013, місто Київ, вулиця Баренбойма, будинок 1)

АКТ

впровадження результатів дисертаційної роботи

«20» квітня 2026 р. м. Київ

Ми, що нижче підписалися, представники ТОВ «АРВІОМ», у складі:

- директора Данилюка Павла Вікторовича;
- керівника проєктів та програм Фесенка Віталія Сергійовича;
- інженера-програміста Розума Олексія Сергійовича

склали цей акт про те, що результати дисертаційної роботи **Гамзи Дмитра Євгенійовича** на тему **«Методи виявлення шкідливої активності в інформаційній системі організації на основі гібридної класифікації»**, поданої на здобуття ступеня доктора філософії за спеціальністю 125 «Кібербезпека», було впроваджено в діяльність ТОВ «АРВІОМ».

1. Суть впровадження

У межах діяльності підприємства впроваджено програмно-алгоритмічне рішення для виявлення шкідливої активності в інформаційній інфраструктурі, що базується на методах гібридної класифікації та ансамблевого машинного навчання.

Впроваджене рішення інтегровано у процеси моніторингу інформаційної безпеки та використовується для аналізу мережевого трафіку і виявлення кіберзагроз у реальному часі.

2. Впроваджені наукові результати

У процесі впровадження використано такі результати дисертаційного дослідження:

1. **Гібридний метод виявлення шкідливої активності на основі стекінг-архітектури**, що поєднує прогнози різнорідних моделей із використанням мета-класифікатора на базі XGBoost.
2. **Метод комплексної попередньої обробки даних**, який включає: балансування класів (SMOTE), нормалізацію (Min-Max) та зниження розмірності (PCA).
3. **Метод багатокритеріального відбору оптимальної конфігурації моделей**, що базується на фільтрації за середніми значеннями (above-average rule) та побудові фронту Парето.
4. **Програмна реалізація системи**, що включає модулі: збору даних, інженерії ознак, попередньої обробки, гібридної класифікації та реагування на інциденти.

3. Результати впровадження

За результатами дослідної та промислової експлуатації встановлено, що впроваджене рішення забезпечує підвищення точності виявлення шкідливої активності до 98 %;

- зниження рівня хибнопозитивних спрацювань на 20–35%;
- скорочення середнього часу виявлення інцидентів (MTTD) на 40–60%;
- забезпечення обробки мережевого трафіку із затримкою не більше 7,16 мс на потік;
- зменшення навантаження на фахівців служби інформаційної безпеки (SOC).

4. Практичне значення

Впроваджені результати дозволили:

- підвищити рівень захищеності інформаційної інфраструктури підприємства;
- покращити ефективність процесів моніторингу кібербезпеки;
- забезпечити виявлення раніше невідомих атак (zero-day);
- оптимізувати використання обчислювальних ресурсів.

Рішення рекомендовано до подальшого використання та масштабування в інформаційних системах підприємства.

5. Висновок

Результати дисертаційної роботи Гамзи Д.Є. є науково обґрунтованими, практично цінними та придатними для використання в реальних умовах функціонування інформаційних систем. Впровадження підтверджує їх ефективність та доцільність застосування у сфері кібербезпеки.

Підписи

від ТОВ «АРВІОМ»:

_____ /Данилюк П.В./

_____ /Фесенко В.С./

_____ /Розум О.С./

М.П.

ДОДАТОК Б

eurotelecom

ТОВАРИСТВО З ОБМЕЖЕНОЮ ВІДПОВІДАЛЬНІСТЮ «ЄВРОТЕЛЕКОМ» 34837389 | +38 099 915-915 9 office@etele.com.ua | 03151, Україна, м.Київ, вул.Волинська, буд.10

№ 24042026-62 від 24.04.2026

АКТ

про впровадження результатів дисертаційної роботи

Комісія у складі: голови комісії директора ТОВ «ЄВРОТЕЛЕКОМ» Булавіна Павла Георгійовича та членів комісії: заступника директора ТОВ «ЄВРОТЕЛЕКОМ» Сабадира Юрія Альбертовича; головного фахівця з програмного забезпечення відділу програмних і апаратних рішень ТОВ «ЄВРОТЕЛЕКОМ» Чистіліної Тетяни Миколаївної, підтверджує, що результати дисертаційного дослідження аспіранта кафедри систем та технологій кібербезпеки Державного університету інформаційно-комунікаційних технологій **Гамзи Дмитра Євгенійовича** на тему **«Методи виявлення шкідливої активності в інформаційній системі організації на основі гібридної класифікації»**, поданого на здобуття наукового ступеня доктора філософії за спеціальністю **125 «Кібербезпека»**, були впроваджені в практичну діяльність ТОВ «ЄВРОТЕЛЕКОМ».

Зокрема, на підприємстві інтегровано розроблене програмне забезпечення, яке базується на запропонованому автором гібридному методі класифікації, що дозволило суттєво підвищити ефективність виявлення шкідливої активності в інформаційній системі підприємства, а саме:

середній час виявлення інцидентів зменшився на 40–60 %;

навантаження на аналітиків SOC знизилося на 20–35 % завдяки скороченню кількості хибнопозитивних спрацювань;

знижено час обробки мережевого трафіку з метою виявлення вторгнень із затримкою, що не перевищує 7,16 мс.

Використання результатів дисертаційної роботи Гамзи Д. Є. дозволило підвищити оперативність реагування на інциденти безпеки в цілому.

Голова комісії: Павло БУЛАВІН

Члени комісії: Юрій Сабадир

Тетяна Чистіліна

ДОДАТОК В

ЗАТВЕРДЖУЮ
Перший проректор
Державного університету інформаційно-комунікаційних технологій
_____ Олександр КОРЧЕНКО
«20» ____ 2026 р.

АКТ

впровадження наукових результатів **Гамзи Дмитра Євгенійовича**, одержаних під час проведення дисертаційної роботи на здобуття наукового ступеня доктора філософії за спеціальністю 125 Кібербезпека, галузі знань 12 Інформаційні технології на тему: «Методи виявлення шкідливої активності в інформаційній системі організації на основі гібридної класифікації» в освітній процес Державного університету інформаційно-комунікаційних технологій

Ми, що нижче підписалися, склали цей акт про те, що результати дисертаційної роботи здобувача наукового ступеня доктора філософії Гамзи Дмитра Євгенійовича впроваджуються та використовуються у навчальному процесі Навчально-наукового інституту кібербезпеки та захисту інформації Державного інституту інформаційно-комунікаційних технологій.

В процесі виконання дисертаційної роботи автором розроблено методи та їх програмну реалізацію, яка базується на мові програмування Python для виявлення шкідливої активності в інформаційній системі організації.

Розроблені методи базуються на застосуванні системного аналізу, методах штучного інтелекту, методах обробки інформації та забезпечують:

– підвищення точності виявлення кіберзагроз (до 98,07 %) та F1-міри (до 96,57 %) за рахунок застосування розробленого гібридного методу на основі архітектури стекінгу, що динамічно поєднує прогнози різнорідних алгоритмів через мета-класифікатор на базі XGBoost;

– зниження кількості хибних спрацювань системи захисту при виявленні шкідливої активності порівняно з існуючими моно-алгоритмами;

– зменшення обчислювального навантаження та скорочення середнього часу прогнозування на 76 % (з 29,37 до 7,28 мс). Це стало можливим завдяки комплексному методу оптимізації набору даних, який поєднує балансування класів (SMOTE), Min-Max нормалізацію та зниження розмірності ознакового простору методом головних компонент.

Запропоновані методи дозволяють зменшити середній час виявлення інцидентів на 40–60 %, знизити навантаження на аналітиків SOC-центру через скорочення хибних тривог на 20–35 %, а також забезпечити обробку мережевого трафіку із затримкою не більше 7,16 мс на потік.

Результати наукових досліджень, зокрема архітектура розробленого методу та принципи оптимізації даних для систем виявлення вторгнень,

успішно впроваджені в навчальний процес на кафедрі систем та технологій кібербезпеки. Вони використовуються під час викладання та виконання практичних робіт з освітніх компонентів «Організація проведення наукових досліджень в кібербезпеці» та «Штучний інтелект».

Використання наукових та практичних результатів дисертаційної роботи Гамзи Д.Є. дозволило суттєво підвищити якість підготовки здобувачів вищої освіти, що сприяло поглибленню теоретичних знань щодо застосування ансамблевих методів машинного навчання та вдосконаленню практичних навичок з розробки й оптимізації сучасних систем виявлення та реагування на інциденти інформаційної безпеки.

Голова комісії:
Директор Навчально-наукового інституту кібербезпеки та захисту інформації
д.т.н., професор Євгенія ІВАНЧЕНКО

Члени комісії:
Завідувач кафедри управління кібербезпекою та захистом інформації,
д.е.н., професор Світлана ЛЕГОМІНОВА

Професор кафедри систем та технологій кібербезпеки,
д.т.н., професор Світлана КАЗМІРЧУК

Доцент кафедри систем та технологій кібербезпеки,
к.т.н., доцент Юрій БОРСУКОВСЬКИЙ

ДОДАТОК Г

```
import pandas as pd
import numpy as np
import time

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.decomposition import PCA
from imblearn.over_sampling import SMOTE

from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier, ExtraTreesClassifier,
GradientBoostingClassifier, StackingClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from xgboost import XGBClassifier
from lightgbm import LGBMClassifier
from catboost import CatBoostClassifier

from sklearn.metrics import accuracy_score, f1_score

# ========================================
# ЕТАП 1: Завантаження та очищення даних
# ========================================
def load_and_clean_data(filepath):
    df = pd.read_csv(filepath)

    df = df.dropna()
    df = df.replace([np.inf, -np.inf], np.nan).dropna()

    if 'Flow Duration' in df.columns:
        df = df[df['Flow Duration'] >= 0]

    X = df.drop('Label', axis=1)
    y = df['Label']

    from sklearn.preprocessing import LabelEncoder
    le = LabelEncoder()
    y = le.fit_transform(y)

    return X, y
```

```
# ========================================
# ЕТАП 2: Комплексна оптимізація датасету
# ========================================
def optimize_dataset(X_train, y_train, X_test):
    smote = SMOTE(random_state=42, n_jobs=-1)
    X_train_resampled, y_train_resampled = smote.fit_resample(X_train, y_train)

    scaler = MinMaxScaler()
    X_train_scaled = scaler.fit_transform(X_train_resampled)
    X_test_scaled = scaler.transform(X_test)

    pca = PCA(n_components=18, random_state=42)
    X_train_pca = pca.fit_transform(X_train_scaled)
    X_test_pca = pca.transform(X_test_scaled)

    return X_train_pca, y_train_resampled, X_test_pca

# ========================================
# ЕТАП 3: Оцінка моделей
# ========================================
def evaluate_model(model, X_train, y_train, X_test, y_test, model_name):
    model.fit(X_train, y_train)

    start_time = time.time()
    y_pred = model.predict(X_test)
    end_time = time.time()

    num_samples = X_test.shape[0]
    total_inference_time_ms = (end_time - start_time) * 1000
    time_per_prediction_ms = total_inference_time_ms / num_samples

    acc = accuracy_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred, average='weighted')

    print(f"{model_name: <30} | Acc: {acc:.4f} | F1: {f1:.4f} | Час (мс):
{time_per_prediction_ms:.4f}")
    return acc, f1, time_per_prediction_ms

# ========================================
# ГОЛОВНИЙ БЛОК ВИКОНАННЯ
# ========================================
if __name__ == "__main__":
    from sklearn.datasets import make_classification
```

```
    X, y = make_classification(n_samples=10000, n_features=80, n_classes=5,
                               weights=[0.85, 0.05, 0.05, 0.02, 0.03], random_state=42)

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, stratify=y,
random_state=42)

    print("\n" + "="*50)
    print("ТЕСТУВАННЯ НА НЕОБРОБЛЕНИХ ДАНИХ")
    print("="*50)

    base_models = {
        "SVM": SVC(kernel='rbf', random_state=42),
        "Random Forest": RandomForestClassifier(n_estimators=100, random_state=42,
n_jobs=-1),
        "KNN": KNeighborsClassifier(n_neighbors=5, n_jobs=-1),
        "XGBoost": XGBClassifier(use_label_encoder=False, eval_metric='mlogloss',
random_state=42, n_jobs=-1),
        "LightGBM": LGBMClassifier(random_state=42, n_jobs=-1, verbose=-1),
        "CatBoost": CatBoostClassifier(iterations=100, random_state=42, verbose=0,
thread_count=-1),
        "Extra Trees": ExtraTreesClassifier(n_estimators=100, random_state=42, n_jobs=-
1),
        "MLP": MLPClassifier(random_state=42),
        "Logistic Regression": LogisticRegression(max_iter=1000, random_state=42,
n_jobs=-1),
        "Naive Bayes": GaussianNB()
    }

    for name, model in base_models.items():
        evaluate_model(model, X_train, y_train, X_test, y_test, name)

    X_train_opt, y_train_opt, X_test_opt = optimize_dataset(X_train, y_train, X_test)

    print("\n" + "="*50)
    print("ТЕСТУВАННЯ НА ОПТИМІЗОВАНИХ ДАНИХ")
    print("="*50)

    for name, model in base_models.items():
        evaluate_model(model, X_train_opt, y_train_opt, X_test_opt, y_test, name)

    estimators_opt1 = [
        ('xgb', XGBClassifier(use_label_encoder=False, eval_metric='mlogloss',
random_state=42, n_jobs=-1)),
        ('cat', CatBoostClassifier(iterations=100, random_state=42, verbose=0,
thread_count=-1)),
```

```
        ('lgb', LGBMClassifier(random_state=42, n_jobs=-1, verbose=-1))
    ]

    estimators_opt2 = [
        ('cat', CatBoostClassifier(iterations=100, random_state=42, verbose=0,
thread_count=-1)),
        ('lgb', LGBMClassifier(random_state=42, n_jobs=-1, verbose=-1)),
        ('et', ExtraTreesClassifier(n_estimators=100, random_state=42, n_jobs=-1))
    ]

    meta_xgb = XGBClassifier(use_label_encoder=False, eval_metric='mlogloss',
random_state=42, n_jobs=-1)
    meta_lr = LogisticRegression(max_iter=1000, random_state=42, n_jobs=-1)
    meta_gb = GradientBoostingClassifier(random_state=42)

    stacking_models = {
        "Stack: XGB+Cat+LGB (Meta: XGB)":
StackingClassifier(estimators=estimators_opt1, final_estimator=meta_xgb, cv=5,
n_jobs=-1),
        "Stack: XGB+Cat+LGB (Meta: LR)":
StackingClassifier(estimators=estimators_opt1, final_estimator=meta_lr, cv=5,
n_jobs=-1),
        "Stack: Cat+LGB+ET (Meta: XGB)":
StackingClassifier(estimators=estimators_opt2, final_estimator=meta_xgb, cv=5,
n_jobs=-1),
        "Stack: Cat+LGB+ET (Meta: GB)":
StackingClassifier(estimators=estimators_opt2, final_estimator=meta_gb, cv=5,
n_jobs=-1)
    }

    for name, model in stacking_models.items():
        evaluate_model(model, X_train_opt, y_train_opt, X_test_opt, y_test, name)
```