Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Центр Післядипломної освіти
(повна назва)
Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

Розробка системи виявлення аномалій у кібербезпеці на основі машинного навчання
(тема)

Виконав:
здобувач другого року навчання,
групи ІТШІпз-23-1
Віктор Щербак
(власне ім'я, прізвище)
Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)
Тип програми освітньо-професійна
Освітня програма Штучний інтелект
(повна назва освітньої програми)
Керівник ст.викл. Філіп Бродецький
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ _____ Олег ЗОЛОТУХІН
(підпис) (власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Центр Післядипломної освіти

Кафедра Штучного інтелекту

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Штучний інтелект
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____» _____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Щербаку Віктору Олексійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка системи виявлення аномалій у кібербезпеці на основі машинного навчання

затверджена наказом університету від 19 травня 20 25 р. № 87Стз
2. Термін подання студентом роботи до екзаменаційної комісії 24 червня 20 25 р.
3. Вихідні дані до роботи набір даних UNSW-NB15; наукові публікації та дослідження в галузі машинного навчання; інструменти: Autoencoder, GAN, LightGBM, XGBoost, SMOTE, One-Hot Encoding, StandardScaler; python бібліотеки: Scikit-learn, PyTorch/TensorFlow, Pandas, NumPy, Matplotlib, Seaborn; метрики оцінювання моделейAccuracy, Precision, Recall, F1-score, ROC-AUC, PR-AUC

4. Перелік питань, що потрібно опрацювати в роботі
1) Теоретичні основи аномалій у кібербезпеці
2) Методологічні аспекти побудови системи виявлення аномалій
3) Експериментальні дослідження та аналіз результатів системи виявлення аномалій

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	19.05.2025	виконано
2	Збір та аналіз наукових джерел, формування	19.05.2025 – 22.05.2025	виконано
	теоретичної бази		
3	Обґрунтування методів машинного навчання,	22.05.2025 – 24.05.2025	виконано
	вибір моделей		
4	Підготовка та обробка даних (передобробка,	24.05.2025 – 26.05.2025	виконано
	SMOTE, кодування)		
5	Розробка та реалізація компонентів системи	26.05.2025 – 28.05.2025	виконано
	(Autoencoder, GAN, LightGBM)		
6	Інтеграція системи HybridIDS	28.05.2025 – 31.05.2025	виконано
7	Проведення експериментів, збір результатів	31.05.2025 – 03.06.2025	виконано
8	Оцінка ефективності, побудова метрик, аналітика	03.06.2025 – 05.06.2025	виконано
9	Формулювання висновків, пропозицій з	05.06.2025 – 07.06.2025	виконано
	удосконалення		
10	Оформлення пояснювальної записки (вступ,	07.06.2025 – 09.06.2025	виконано
	зміст, висновки, додатки)		
11	Підготовка до захисту (презентація, доповідь,	09.06.2025 – 20.06.2025	виконано
	відповіді на запитання)		
12	Захист кваліфікаційної роботи	24.06.2025	виконано

Дата видачі завдання 19 травня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ ст.викл. Філіп Бродецький
(підпис) (посада, власне ім’я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 73 с., 11 рис., 1 табл., 5 дод., 27 джерел.

АВТОЕНКОДЕР, АНОМАЛІЇ, ГІБРИДНА СИСТЕМА, КІБЕРБЕЗПЕКА, GAN, LIGHTGBM, RANDOM FOREST, SMOTE, STRATIFIED K-FOLD, XGBOOST.

Об’єктом дослідження є мережевий трафік у комп’ютерних системах, що підлягає моніторингу з метою виявлення потенційних кіберзагроз.

Предметом дослідження є алгоритми та моделі машинного навчання, які використовуються для виявлення аномальної активності у трафіку.

Метою роботи є розробка та дослідження ефективності гібридної системи виявлення аномалій у кібербезпеці, яка поєднує контрольовані та неконтрольовані методи навчання.

Методи дослідження – Autoencoder, Generative Adversarial Network (GAN), LightGBM, XGBoost, SMOTE, One-Hot Encoding, StandardScaler, метрики класифікації (Accuracy, F1-score, ROC, PR), крос-валідація.

У результаті роботи реалізовано функціональну гібридну систему виявлення аномалій, виконано її тестування на даних мережевого трафіку, визначено ефективність кожного з компонентів системи, встановлено доцільність поєднання глибоких нейронних мереж і бустингових алгоритмів. Запропоновано гібридну систему HybridIDS, що поєднує автоенкодер, генеративну змагальну мережу та бустингові алгоритми LightGBM/XGBoost. Система дозволяє ідентифікувати як відомі, так і нові типи атак у мережевому трафіку.

ABSTRACT

Bachelor’s thesis contains: 73 pp., 11 fig., 1 tabl., 5 ann., 27 references.

AUTOENCODER, ANOMALIES, HYBRID SYSTEM, CYBERSECURITY, GAN, LIGHTGBM, RANDOM FOREST, SMOTE, STRATIFIED K-FOLD, XGBOOST.

Object of research – network traffic in computer systems that is subject to monitoring to detect potential cyber threats.

Subject of research – machine learning algorithms and models used to detect anomalous activity in traffic.

Purpose of the work – to develop and evaluate the effectiveness of a hybrid anomaly detection system for cybersecurity by combining supervised and unsupervised learning methods.

Research methods – Autoencoder, Generative Adversarial Network (GAN), LightGBM, XGBoost, SMOTE, One-Hot Encoding, StandardScaler, classification metrics (Accuracy, F1-score, ROC, PR), cross-validation.

As a result of the research, a functional hybrid anomaly detection system was implemented and tested on network traffic data. The effectiveness of each component was evaluated, and the feasibility of combining deep neural networks with boosting algorithms was confirmed. The proposed system, HybridIDS, integrates an autoencoder, a generative adversarial network, and boosting algorithms (LightGBM/XGBoost). The system is capable of identifying both known and previously unseen types of attacks in network traffic.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

AE – Autoencoder – автоенкодер;

DL – Deep Learning – глибинне навчання;

F1-score – гармонічне середнє точності та повноти;

GAN – Generative Adversarial Network – генеративна змагальна мережа;

LightGBM – бібліотека градієнтного бустингу на базі листового зростання;

ML – Machine Learning – машинне навчання;

MSE – Mean Squared Error – середньоквадратична помилка;

PR – Precision-Recall – претензійно-повнотна крива;

ROC – Receiver Operating Characteristic – крива оперативної характеристики;

SMOTE – Synthetic Minority Over-sampling Technique – метод синтетичної передискретизації меншин;

XGBoost – бібліотека градієнтного бустингу.

ВСТУП

У сучасному глобалізованому світі, де інформаційно-комунікаційні технології набули статусу критичної інфраструктури, питання кібернетичної безпеки перетворилося на одне з найбільш пріоритетних завдань для державних установ, корпоративного сектору та індивідуальних користувачів. Значне зростання кількості підключених пристроїв, масштабування обсягів даних, що передаються мережами, та інтеграція цифрових технологій у всі сфери життєдіяльності створюють плідний ґрунт для виникнення та поширення кіберзагроз різноманітної складності.

У цьому контексті особливого значення набувають розподілені атаки на відмову в обслуговуванні (DDoS), фішингові кампанії підвищеної складності, атаки з використанням експлойтів нульового дня, а також цілеспрямовані кібероперації, що здійснюються висококваліфікованими зловмисниками. Традиційні методи захисту, засновані на сигнатурному аналізі та статичних правилах, демонструють недостатню ефективність у протидії цим загрозам через їхню нездатність адаптуватися до швидкозмінного ландшафту кібербезпеки та виявляти невідомі типи атак.

Актуальність дослідження зумовлена об'єктивною необхідністю у розробці інноваційних підходів до створення проактивних систем захисту, здатних функціонувати в режимі реального часу та ефективно ідентифікувати аномальну поведінку в динамічних мережевих середовищах. Зважаючи на складність та різноманітність сучасних кібератак, виникає нагальна потреба у впровадженні інтелектуальних систем, що використовують передові алгоритми машинного навчання для аналізу мережевого трафіку та виявлення потенційних загроз на ранніх стадіях їх розгортання.

Одним із найбільш перспективних напрямів у цій галузі є застосування технологій машинного навчання, зокрема методів виявлення аномалій, для побудови гнучких та ефективних систем захисту. Виявлення

аномалій представляє собою комплексний процес ідентифікації нетипових подій або зразків поведінки, що значно відхиляються від встановленої норми функціонування системи. Перевага цього підходу полягає у здатності виявляти не лише відомі типи атак, але й нові, раніше не класифіковані загрози, що є критично важливим фактором в умовах постійної еволюції методів кібернападів.

Мета представленої кваліфікаційної роботи полягає у розробці, імплементації та всебічному дослідженні гібридної системи виявлення аномалій у мережевому трафіку з інтеграцією кількох взаємодоповнюючих методів машинного навчання. Інноваційність запропонованого підходу ґрунтується на одночасному поєднанні трьох різнопланових алгоритмічних підходів: автоенкодера (Autoencoder) для компресії та реконструкції даних, генеративної змагальної мережі (Generative Adversarial Network, GAN) для моделювання розподілу нормального трафіку та градієнтного бустингу (Light Gradient Boosting Machine, LightGBM) як потужного ансамблевого методу класифікації.

Об'єктом дослідження виступає неоднорідний мережевий трафік у сучасних комп'ютерних мережах різного масштабу та призначення, включаючи його статистичні, часові та структурні характеристики.

Предметом дослідження є математичні моделі та алгоритми машинного навчання, спрямовані на виявлення аномальної активності в мережевому трафіку, а також методологічні підходи до їх оптимізації, валідації та інтеграції у комплексну систему виявлення вторгнень.

Наукова новизна роботи полягає у розробці гібридної архітектури, що органічно поєднує переваги контрольованого та неконтрольованого навчання, забезпечуючи високу точність класифікації та здатність до генералізації на широкому спектрі типів мережевого трафіку.

Запропонована методологія дозволяє ефективно виявляти як відомі, так і нові типи аномалій, демонструючи підвищену стійкість до зашумлених даних та зменшуючи кількість хибнопозитивних спрацьовувань.

1 ТЕОРЕТИЧНІ ОСНОВИ АНОМАЛІЙ У КІБЕРБЕЗПЕЦІ

1.1 Концептуально-категоріальний апарат та систематизація підходів до аномалій у кібербезпеці

У парадигмі сучасної інформаційної безпеки феномен аномалій інтерпретується як сукупність подій, патернів поведінки чи відхилень від типових характеристик функціонування системи, що демонструють статистично значущі розбіжності від нормативних показників. Аномальні прояви в інформаційних системах із високою ймовірністю слугують індикаторами потенційних порушень безпеки або ознаками несанкціонованої активності, що обумовлює фундаментальний інтерес фахівців кібербезпеки до розробки й удосконалення методологічного інструментарію їх виявлення.

Аномалії – це відхилення від норми або очікуваної поведінки, які можуть виникати в різних контекстах, включаючи науку та технології. У контексті кібербезпеки аномалії – це відхилення від очікуваної поведінки системи, які можуть свідчити про порушення безпеки. Це може бути викликано різними факторами, як-от зловмисне програмне забезпечення, хакерство або внутрішні загрози. Аномалії відіграють вирішальну роль у кібербезпеці, де вони часто використовуються для виявлення зловмисних дій або інших загроз безпеці. Їхнє детектування передбачає ідентифікацію шаблонів, які відрізняються від нормальної поведінки системи, що може вказувати на наявність атаки або порушення безпеки. Виявлення аномалій має вирішальне значення для підтримки безпеки мереж, систем і даних [1].

Розуміння різних фаз атаки є важливим для виявлення аномалій. Різноманітні етапи, які може включати атака, це розвідка, напад, експлуатація та інші. Різні типи аномалій можуть бути видимі на різних фазах атак, тому важливо мати розуміння про те , які аномалії можуть бути спостережені на кожній фазі, і як їх можна виявити.

1.1.1 Типи аномалій

Таксономія аномалій у сфері кібербезпеки характеризується багаторівневою структурою та включає наступні категорії:

Точкові аномалії (Point Anomalies) – представляють ізольовані події чи окремі інстанції даних, характеристики яких суттєво перевищують допустимі межі статистичної варіації, притаманні нормальному функціонуванню системи.

Контекстуальні аномалії (Contextual Anomalies) – відображають відхилення, інтерпретація яких безпосередньо залежить від супутніх обставин: часових характеристик, геолокаційних параметрів, специфіки користувацької активності чи інших контекстуальних факторів. Наприклад, автентифікація користувача в нічний час може бути аномальною для працівника з фіксованим робочим графіком, але нормальною для співробітника служби технічної підтримки з гнучким розкладом.

Колективні аномалії (Collective Anomalies) – представляють агреговані множини подій чи транзакцій, кожна з яких окремо не демонструє аномальних властивостей, проте в сукупності формує статистично значущий патерн, що відхиляється від нормативної поведінки. Подібні аномалії характеризуються просторово-часовою кореляцією та потребують комплексного аналізу взаємозв'язків між окремими компонентами.

Ефективність систем виявлення аномалій суттєво залежить від їхньої здатності адаптуватися до динамічних змін характеристик мережевого трафіку, еволюції векторів кібератак та трансформації поведінкових профілів легітимних користувачів. Це зумовлює необхідність імплементації алгоритмів машинного навчання з можливістю автоматичного коригування параметрів та перенавчання на основі актуалізованих даних [2].

1.2 Категоризація та характеристика мережевих атак

Мережеві атаки представляють собою спрямовані дії зловмисників, метою яких є порушення конфіденційності, цілісності або доступності інформаційних систем. Сучасна класифікація мережевих атак дозволяє систематизувати та краще розуміти техніки, що використовуються кіберзлочинцями. Пропоную розглянути основні категорії та їхні характеристики.

Fuzzing-атаки представляють собою методологію тестування безпеки, що базується на принципі подачі аномальних, непередбачених або випадкових даних на вхід програмного забезпечення. Ключовою метою таких атак є виявлення потенційних вразливостей, які виникають при обробці некоректних вхідних даних.

Характерною особливістю fuzzing-атак є систематичне відхилення від стандартної поведінки протоколів та застосунків. Індикаторами таких атак часто виступають: аномальні розміри мережевих пакетів, що суттєво відрізняються від середньостатистичних значень; повторювані запити з ознаками модифікації; порушення послідовності виконання протоколів; а також надмірна кількість повідомлень про помилки у відповідях серверів, що свідчить про реакцію на некоректні вхідні дані.

Аналітичні атаки охоплюють комплекс методів інформаційної розвідки, спрямованих на з'ясування архітектури мережевої інфраструктури та виявлення потенційних вразливостей. Ці атаки мають переважно підготовчий характер і створюють фундамент для здійснення більш складних деструктивних дій.

Характерними маркерами аналітичних атак є: систематичні спроби доступу до адміністративних інтерфейсів; регулярні запити до системних журналів; методичне сканування відкритих мережевих портів; використання спеціалізованих засобів мережевої розвідки, таких як nmap, xprobe та аналогічні інструменти.

Атаки типу «бекдор» передбачають створення прихованих механізмів доступу до інформаційної системи, що дозволяють обходити стандартні процедури автентифікації та авторизації. Такі атаки суттєво підвищують рівень тривалості зловмисника в системі, забезпечуючи довготривалий несанкціонований доступ.

Індикаторами потенційної наявності бекдорів є: встановлення з'єднань з нестандартними портами, що не використовуються легітимними застосунками; функціонування служб без належної автентифікації; виявлення комунікацій з підозрілими зовнішніми IP-адресами; нетипова мережева активність у нічний час або в період зниженої активності користувачів.

Атаки на відмову в обслуговуванні (DoS) базуються на принципі вичерпання обчислювальних ресурсів цільової системи шляхом генерації надмірного обсягу запитів або використання специфічних вразливостей у протоколах. Метою таких атак є порушення нормального функціонування сервісів та недоступність їх для легітимних користувачів.

Діагностичними ознаками DoS-атак є: аномально висока частота однотипних запитів, що надходять з обмеженої кількості IP-адрес; однорідність структури запитів, що свідчить про їх автоматизоване генерування; нестабільна робота мережевих інтерфейсів; експоненціальне зростання обсягів мережевого трафіку за короткий проміжок часу.

Атаки з використанням експлойтів спрямовані на використання відомих вразливостей у програмному забезпеченні або неправильних конфігурацій систем. Метою таких атак є отримання несанкціонованого доступу до системи, підвищення привілеїв або виконання довільного коду.

Характерними ознаками експлойт-атак є: аномальні послідовності системних викликів, що не відповідають стандартній логіці роботи застосунків; нетипове використання мережевих протоколів; передача нестандартних комбінацій параметрів; ініціювання процесів з підвищеними привілеями (root/адміністратор) без належної авторизації.

Генеричні атаки представляють собою універсальні методи компрометації, які не є специфічними для конкретних операційних систем або протоколів. Вони базуються на загальних принципах функціонування криптографічних алгоритмів, структур даних та архітектурних рішень.

Ідентифікація генеричних атак ґрунтується на виявленні таких ознак: підвищений безлад мережевого трафіку, що свідчить про інтенсивні криптографічні операції; рівномірний розподіл джерел запитів у просторі IP-адрес; відсутність логічного зв'язку між окремими сесіями; повторюваність шаблонів мережевої активності.

Розвідувальні атаки є фундаментальним етапом більшості кібернетичних операцій і спрямовані на збір інформації про мережеву інфраструктуру, активні хости, доступні сервіси та версії програмного забезпечення. Такі атаки зазвичай передують більш цілеспрямованим діям.

Діагностичними ознаками розвідувальних атак є: послідовне сканування діапазонів IP-адрес; інтенсивне використання ICMP-протоколу для виявлення активних хостів; велика кількість незавершених TCP-з'єднань (SYN-запити); підвищена активність протоколів ARP та DNS, спрямована на отримання інформації про мережеву топологію.

Шелкод-атаки базуються на впровадженні виконуваного машинного коду в пам'ять цільової системи з метою отримання контролю над нею. Такі атаки часто використовують вразливості типу «переповнення буфера» та інші механізми маніпуляції пам'яттю.

Характерними ознаками шелкод-атак є: передача послідовностей байтів, що мають структуру, подібну до машинного коду; виконання системних команд через зовнішні інтерпретатори; аномальні операції з адресним простором процесів; переповнення буферів у полях даних мережевих пакетів.

Мережеві черви є різновидом шкідливого програмного забезпечення, що характеризується здатністю до самоактуалізації та автономного поширення в мережевому середовищі. Черви використовують різноманітні

вразливості для інфікування нових хостів та можуть виконувати деструктивні дії, збір інформації або створення бекдорів.

Діагностичними ознаками активності мережевих червів є: множинні однотипні з'єднання з великою кількістю хостів; рівномірний розподіл цільових IP-адрес; гомогенність структури запитів; експоненціальне зростання кількості мережевих з'єднань, що свідчить про процес реплікації.

Розуміння специфіки різних типів мережевих атак є важливим елементом у розробці ефективних стратегій захисту інформаційних систем. Кожна категорія атак має характерні ознаки, що дозволяють ідентифікувати потенційні загрози та вживати відповідних контрзаходів. Систематизація знань про мережеві атаки сприяє підвищенню загального рівня кібербезпеки та покращенню здатності організацій протидіяти сучасним кіберзагрозам [3].

1.3 Сучасні виклики у виявленні аномалій для кібербезпеки

Однією з найсуттєвіших проблем у сфері виявлення аномалій для кібербезпеки є масштабування систем для обробки колосальних обсягів інформації в режимі реального часу. Традиційні методи аналізу даних стають неефективними, коли йдеться про потужні інформаційні потоки. Великі технологічні корпорації, такі як Facebook та Google, щоденно стикаються з необхідністю аналізувати надзвичайно великі обсяги трафіку, що вимагає розробки інноваційних рішень для виявлення загроз [4].

Істотною перешкодою при роботі з великими масивами даних є високий рівень шуму. Переважна більшість аналізованого трафіку не містить реальних загроз, але створює додаткове навантаження на системи моніторингу. Цей шум суттєво ускладнює роботу аналітиків, які мають відфільтровувати нерелевантні дані для отримання точних результатів. Хоча існують спеціальні методи фільтрації, їх впровадження часто потребує

значних ресурсів через непередбачуваність і різноманітність інформаційних потоків.

Системи виявлення аномалій також страждають від численних хибних спрацьовувань, коли звичайні операції помилково класифікуються як потенційні загрози. Це призводить до численних помилкових тривог, знижуючи загальну ефективність роботи кібербезпеки. Наприклад, за аналітичними даними провідні фінансові інституції США зазнають значних фінансових втрат через помилкові тривоги системи безпеки, що спричиняє затримки в обробці транзакцій.

Постійна еволюція кіберзагроз становить додаткову складність. Аномалія, виявлена сьогодні, може стати нормою завтра, а моделі виявлення не завжди здатні адаптуватися до таких швидких змін. Сучасні фішингові атаки та віруси регулярно модифікують свою поведінку, тому статичні системи захисту швидко застарівають. Ця динаміка вимагає впровадження постійно оновлюваних моделей виявлення, що потребує значних ресурсів.

Для ефективного навчання моделей виявлення аномалій критично важливими є якісні дані. Проте організації часто не мають достатньої кількості інформації про реальні атаки або не бажають ділитися такими відомостями через міркування конфіденційності та репутаційні ризики. Це негативно впливає на точність моделей виявлення, оскільки без достатнього обсягу релевантних даних системи працюють неоптимально.

Різноманітність джерел даних також створює виклики. Інформація може надходити з різних систем – мережевих логів, записів подій безпеки, трафіку різноманітних пристроїв. Ці дані часто мають несумісну структуру, що ускладнює їх інтеграцію та спільний аналіз. Наприклад, мережеві логи можуть кардинально відрізнятися за форматом від журналів системи безпеки, що вимагає складних процесів нормалізації.

Багато сучасних інструментів виявлення аномалій мають суттєві обмеження. Рішення на основі сигнатур втрачають актуальність, оскільки не здатні виявляти нові або модифіковані атаки, що не відповідають

відомим шаблонам. Традиційні антивірусні технології дедалі менше відповідають сучасним вимогам через обмежену здатність виявляти нові загрози.

Швидкість реагування є критично важливим фактором, особливо при аналізі даних у реальному часі. Затримки в обробці можуть суттєво знизити ефективність системи безпеки. Прикладом може слугувати атака на Target у 2013 році, коли несвоєчасний аналіз даних призвів до компрометації мільйонів кредитних карт.

Незважаючи на високий рівень автоматизації, людський фактор залишається вирішальним у процесі виявлення аномалій. Аналітики повинні кваліфіковано інтерпретувати результати автоматизованих систем і приймати зважені рішення. Брак професіоналів або їхні помилки можуть призвести до критичних наслідків для кібербезпеки. Статистика показує, що понад 90% інцидентів у сфері кібербезпеки пов'язані саме з людським фактором.

Впровадження ефективних систем виявлення аномалій потребує значних фінансових інвестицій. Великі корпорації, такі як Microsoft, вкладають значні ресурси не лише в придбання ліцензій та розвиток інфраструктури, але й у постійну підтримку і оновлення систем. Багато компаній не готові до таких витрат, що створює додаткові перешкоди для впровадження сучасних рішень.

Для подолання цих викликів необхідно інтегрувати передові технології, зокрема машинне навчання та штучний інтелект, які можуть покращити процес виявлення аномалій, прискорити аналіз даних та зменшити залежність від людського фактора. Також організаціям необхідно інвестувати в навчання персоналу та формування комплексних стратегій кіберзахисту, що охоплюють технологічні, організаційні та людські аспекти для адаптації до постійно еволюціонуючих загроз.

2 МЕТОДОЛОГІЧНІ АСПЕКТИ ПОБУДОВИ СИСТЕМИ ВИЯВЛЕННЯ АНОМАЛІЙ

2.1 Обґрунтування вибору моделей машинного навчання для виявлення аномалій

Сучасні кіберзагрози характеризуються постійною еволюцією та адаптацією до існуючих механізмів захисту, що створює фундаментальну проблему для традиційних систем виявлення вторгнень. У контексті цієї проблематики, мною було обрано гібридний підхід, який дозволяє не лише ефективно виявляти вже відомі вектори атак, але й забезпечує можливість ідентифікації нових, раніше невідомих типів зловмисної активності.

Обґрунтованість такого підходу підтверджується численними науковими дослідженнями, які демонструють перевагу гібридних архітектур над монолітними рішеннями.

2.1.1 Методологічний інструментарій ідентифікації аномалій

З огляду на багатогранність проблеми виявлення аномалій у кіберсередовищі, методологічний арсенал включає диверсифіковані підходи, що різняться за принципами аналізу та способами інтерпретації даних [4]:

а) статистично-ймовірнісні методи базуються на формулюванні та валідації гіпотез щодо стохастичної моделі розподілу нормальних даних. Ці методи передбачають побудову ймовірнісного простору та визначення метрики відстані між спостережуваними значеннями та теоретичним розподілом. Спостереження, відстань яких перевищує статистично обґрунтований поріг, класифікуються як аномальні. До статистичних методів належать:

– параметричні методи (Gaussian Mixture Models, Regression Models);

– непараметричні методи (Kernel Density Estimation, Histogram-based Approaches);

– Байєсові мережі та ймовірнісні графічні моделі;

б) нормативно-дедуктивні методи (Rule-based approaches) ґрунтуються на формалізованих пошуках та експертно-сформульованих шаблонах, що відображають характеристики нормальної чи аномальної поведінки. Такі методи передбачають інтеграцію доменних знань у вигляді логічних предикатів та умовних конструкцій. Хоча дані підходи демонструють високу ефективність у ідентифікації відомих типів атак та аномалій, їх адаптивність до нових, раніше неспостережуваних загроз залишається обмеженою. До цієї категорії належать:

– сигнатурний аналіз (Signature-based Detection);

– експертні системи (Expert Systems);

– системи на основі логічних правил (Logic-based Systems);

в) методи машинного навчання характеризуються здатністю до виявлення прихованих закономірностей у даних та автоматичної адаптації до нових типів аномалій. Залежно від наявності розмічених даних для навчання, ці методи поділяються на:

1) методи контрольованого навчання (Supervised Learning) – передбачають використання анотованих датасетів із бінарними або мультикласовими мітками для побудови класифікаційної моделі. Навчання здійснюється шляхом мінімізації функції втрат, що відображає розбіжність між прогнозованими та справжніми мітками класів. До цієї категорії належать:

– алгоритми на основі дерев рішень (Random Forest, Gradient Boosting);

– методи опорних векторів (Support Vector Machines);

– глибокі нейронні мережі (Deep Neural Networks);

2) методи неконтрольованого навчання (Unsupervised Learning) – не потребують апріорної розмітки даних і базуються на виявленні внутрішньої структури та кластеризації спостережень. Аномалії ідентифікуються як спостереження, що демонструють значну відмінність від основних кластерів або мають низьку щільність ймовірності. Основні представники:

– алгоритми кластеризації (K-means, DBSCAN, Hierarchical Clustering);

– методи зниження розмірності (PCA, t-SNE, UMAP);

– методи оцінки щільності розподілу (One-class SVM, Isolation Forest);

3) методи напівконтрольованого навчання (Semi-supervised Learning) – функціонують в умовах обмеженої кількості розмічених зразків, зазвичай представлених лише нормальними спостереженнями. Модель навчається реконструювати нормальну поведінку, а спостереження з високою похибкою реконструкції класифікуються як аномальні. Ключові представники:

– автоенкодери з регуляризацією (Regularized Autoencoders);

– генеративно-змагальні мережі для однокласової класифікації (One-class GAN);

– гібридні методи з частковим використанням розмітки (Hybrid Semi-supervised Approaches).

2.1.2 Алгоритмічні підходи машинного навчання для детектування аномалій

Автоенкодер (Autoencoder). Представляє собою архітектурно-специфічну реалізацію глибокої нейронної мережі, функціональність якої ґрунтується на принципі «пісочного годинника» з двома ключовими

компонентами: енкодером, що здійснює нелінійне стиснення даних до латентного простору меншої розмірності, та декодером, що забезпечує реконструкцію вхідних даних із цього латентного представлення. Математичне представлення наведене у рисунку 2.1.

$$\phi : \mathcal{X} \to \mathcal{F}$$
$$\psi : \mathcal{F} \to \mathcal{X}$$
$$\phi, \psi = \underset{\phi,\psi}{\arg\min} \|X - (\psi \circ \phi)X\|^2$$

Рисунок 2.1 – Математичний представлення енкодингу

Автоенкодери прагнуть мінімізувати похибку реконструкції, яка є різницею між вхідним та реконструйованим вихідним даними. Вони використовують функції втрат, такі як середньоквадратична похибка Mean Squared Error (MSE) або бінарна перехресна ентропія Binary Cross-Entropy, та оптимізують за допомогою зворотного поширення та градієнтного спуску. Вони використовуються в таких програмах, як обробка зображень, виявлення аномалій, видалення шуму та вилучення ознак.

Архітектура автоенкодера складається з трьох основних компонентів: енкодера, вузького місця (латентного простору) та декодера.

Енкодер – це частина мережі, яка приймає вхідні дані та стискає їх у менше нижньорозмірне представлення.

Вхідний шар: це шар, де вихідні дані надходять у мережу, наприклад, зображення або набір ознак.

Приховані шари: ці шари застосовують перетворення до вхідних даних. Мета кодера – виділити важливі ознаки та зменшити розмірність даних.

Візуальна схема роботи автоенкодера наведена на рисунку 2.2.

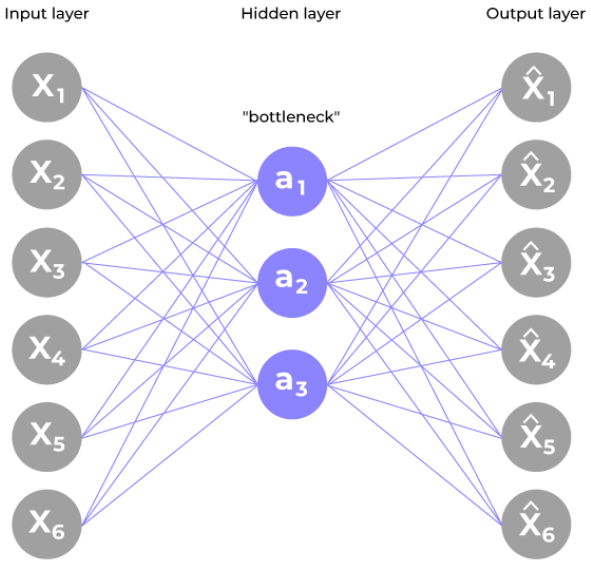

Рисунок 2.2 – Архітектура автоенкодера

Вихідні дані кодера (латентний простір): кодер видає стиснуту версію даних, яку часто називають латентним представленням або кодуванням. Це стисла версія вхідних даних, що зберігає лише важливі характеристики.

Вузьке місце (латентний простір) – це найменший шар мережі, де дані представлені в найбільш стислому вигляді. Його часто називають латентним простором або кодом.

Цей шар містить скорочений набір ознак, що представляють найважливішу інформацію з вхідних даних.

Ідея полягає в тому, що завдяки цьому стисненню мережа вивчає ключові шаблони та структури вхідних даних.

Декодер відповідає за отримання стиснутого представлення з латентного простору та його відновлення у вихідній формі даних.

Приховані шари: Декодер використовує серію шарів для поступового розширення стиснутих даних назад до розмірів вихідних вхідних даних.

Вихідний шар: Цей шар створює реконструйовані дані та прагне максимально наближатися до вхідних даних.

Обмеження автоенкодера дозволяє йому навчатися та відображати ефективне представлення. Обмеження автоенкодера означає, що мережа вивчає значущі, компактні та корисні функції з вхідних даних. Після навчання мережі лише частина кодера використовується для кодування подібних даних для майбутніх завдань.

Генеративно-змагальні мережі (Generative Adversarial Networks) були представлені Ієном Гудфеллоу та його колегами у 2014 році. GAN – це клас нейронних мереж, які автономно вивчають закономірності у вхідних даних для створення нових прикладів, що нагадують вихідний набір даних, робота яких наведена на рисунку 2.3.

Архітектура GAN складається з двох нейронних мереж:

– генератор: створює синтетичні дані з випадкового шуму, щоб отримати дані настільки реалістичні, що дискримінатор не може відрізнити їх від реальних даних;

– дискримінатор: діє як критик, оцінюючи, чи є отримані ним дані справжніми чи фальшивими.

Вони використовують змагальне навчання для створення штучних даних, ідентичних фактичним даним.

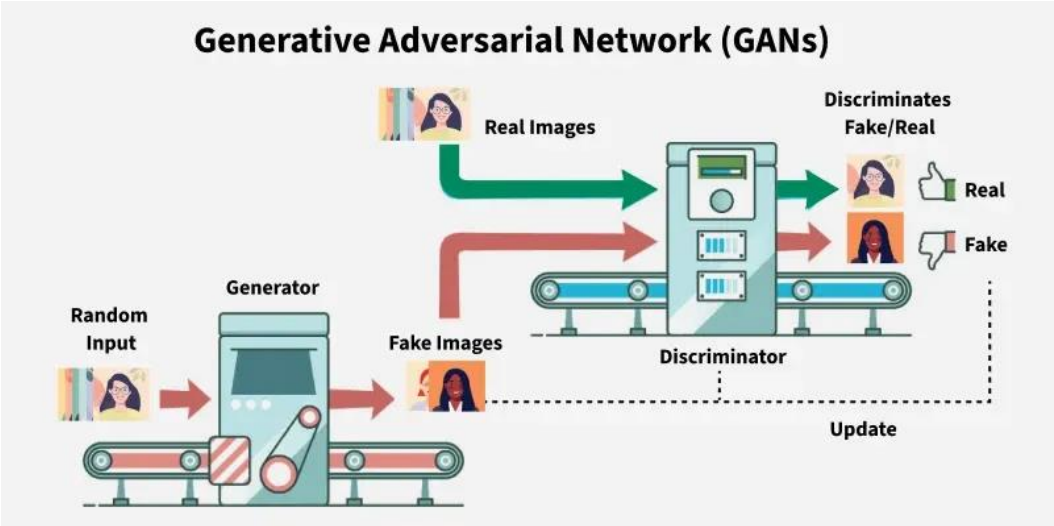

Рисунок 2.3 – GAN навчання

Дві мережі постійно беруть участь у грі в кішки-мишки: Генератор покращує свою здатність створювати реалістичні дані, тоді як Дискримінатор стає кращим у виявленні підробок. З часом цей змагальний процес призводить до генерації дуже реалістичних та високоякісних даних.

Генератор – це глибока нейронна мережа, яка приймає випадковий шум як вхідні дані для генерації реалістичних зразків даних (наприклад, зображень або тексту). Він вивчає базовий розподіл даних, коригуючи його параметри за допомогою зворотного поширення.

Мета генератора – створювати зразки, які дискримінатор класифікує як реальні.

Дискримінатор діє як бінарний класифікатор, розрізняючи реальні та згенеровані дані. Він навчається покращувати свою класифікаційну здатність шляхом навчання, уточнюючи свої параметри для точнішого виявлення підроблених зразків.

Під час роботи з даними зображень дискримінатор часто використовує згорткові шари або інші відповідні архітектури, що підходять до типу даних. Ці шари допомагають виявляти ознаки та покращують здатність моделі розрізняти реальні та згенеровані зразки.

Дискримінатор зменшує негативний логарифм ймовірності правильної класифікації як створених, так і реальних зразків.

У контексті детектування аномалій GAN використовується двома основними способами:

AnoGAN – підхід, що оцінює аномальність зразка шляхом вимірювання:

– відстані між зразком та його проєкцією в простір генератора (реконструкційна відстань);

– відмінності в активаціях дискримінатора для реального зразка та його реконструкції (дискримінаційна відстань).

BiGAN/ALI – архітектура, що додатково навчає енкодер, здатний відображати зразки назад у латентний простір, що дозволяє більш ефективно обчислювати аномальний скор.

LightGBM (Light Gradient Boosting Machine) репрезентує високоефективну реалізацію алгоритму градієнтного бустингу на основі дерев рішень, що характеризується інноваційними архітектурними особливостями:

– Gradient-based One-Side Sampling (GOSS) – техніка, що оптимізує процес навчання шляхом селективного відбору зразків з високими градієнтами, зберігаючи незначну частку зразків з низькими градієнтами для збалансованості. Це дозволяє знизити обчислювальну складність без значного погіршення якості моделі;

– Exclusive Feature Bundling (EFB) – підхід, що групує взаємовиключні або високо корельовані ознаки в «бандли», що дозволяє оптимізувати використання пам'яті та прискорити навчання на розріджених даних;

– Leaf-wise (Best-first) стратегія зростання дерев – на відміну від традиційної level-wise стратегії, LightGBM обирає листовий вузол з максимальним зниженням функції втрат, що призводить до більш глибоких і асиметричних дерев, але потенційно вищої точності.

LightGBM розроблений для ефективності, масштабованості та високої точності, особливо з великими наборами даних. Він використовує дерева рішень, які ефективно зростають, мінімізуючи використання пам'яті та оптимізуючи час навчання. Ключові інновації, такі як градієнтна одностороння вибірка (GOSS) , алгоритми на основі гістограм та полистове зростання дерев, дозволяють LightGBM перевершувати інші фреймворки як за швидкістю, так і за точністю.

Навчання в LightGBM включає підгонку моделі градієнтного бустування до набору даних. Під час навчання модель багатократно будує дерева рішень для мінімізації заданої функції втрат, коригуючи параметри дерева для оптимізації продуктивності моделі. Оцінювання визначає

продуктивність навченої моделі за допомогою таких метрик, як середньоквадратична помилка для завдань регресії або точність для завдань класифікації. Методи перехресної перевірки можуть бути використані для перевірки продуктивності моделі на невидимих даних та запобігання перенавчанню.

LightGBM пропонує кілька ключових переваг:

– швидкість і точність: він перевершує інші алгоритми градієнтного підвищення продуктивності на великих наборах даних;

– низьке використання пам'яті: оптимізовано для ефективності використання пам'яті та обробки великих наборів даних з мінімальними накладними витратами;

– підтримка паралельного навчання та навчання на графічному процесорі: використовує переваги кількох ядер або графічних процесорів для швидшого навчання;

– ефективний для великих наборів даних: його оптимізовані методи, такі як полистове зростання та навчання на основі гістограм, роблять його придатним для застосувань великих даних.

2.2 Методологія передобробки та підготовки даних для тренування моделей

Передобробка та підготовка даних є ключовим етапом у процесі розробки моделей машинного навчання, оскільки якість вхідних даних безпосередньо впливає на точність, стабільність і узагальнювальну здатність побудованих моделей. Методологія цього етапу передбачає систематичне виконання низки послідовних процедур, спрямованих на приведення сирих даних до формату, придатного для ефективного машинного аналізу [5].

На першому етапі здійснюється збір і первинне ознайомлення з даними. Це включає ідентифікацію джерел інформації, аналіз структури

набору даних, виявлення типів змінних (числових, категоріальних, текстових, часових тощо), а також вивчення обсягу наявних даних, наявності пропущених значень, аномалій і дублікатів. Важливим аспектом цього етапу є чітке розуміння сутності цільової змінної (target), яка використовується для навчання моделі.

Другий етап передбачає очищення даних, що включає обробку пропущених значень (через їх видалення або їх імітацію), видалення дублікатів записів, а також виявлення та обробку викидів шляхом статистичного аналізу або візуалізації. Додатково може здійснюватися приведення типів змінних до належного формату (наприклад, перетворення символьних представлень дат у часового типу).

Третім етапом є перетворення ознак (feature engineering). До основних процедур належать кодування категоріальних ознак (зокрема one-hot та label encoding), нормалізація та стандартизація числових змінних (наприклад, за допомогою Min-Max scaling або Z-стандартизації), а також створення нових похідних змінних на основі наявної інформації. У випадку роботи з текстовими або часовими даними застосовуються спеціалізовані методи попередньої обробки, включаючи токенізацію, лематизацію або агрегування за часовими інтервалами.

Наступним етапом є балансування класів, яке є актуальним у задачах класифікації з диспропорційним розподілом цільових міток. З метою підвищення ефективності навчання застосовуються методи надвідбору (наприклад, SMOTE), недовідбору або модифікація вагових коефіцієнтів для кожного класу при навчанні моделі [6].

Після цього дані розподіляються на навчальну, валідаційну та тестову вибірки. Розділення може здійснюватися випадковим чином або із збереженням пропорцій цільової змінної. Такий підхід забезпечує можливість об'єктивного оцінювання продуктивності моделі на незалежному наборі даних.

На завершальному етапі проводиться збереження процедур обробки, зокрема через побудову обчислювальних пайплайнів, що гарантує відтворюваність результатів та уможливлює застосування однакових трансформацій до нових, раніше невідомих даних. Також доцільною є візуалізація розподілу ознак та результатів попередньої обробки задля додаткового контролю якості.

Узагальнюючи, ефективна методологія передобробки та підготовки даних є неодмінною складовою побудови надійних моделей машинного навчання та гарантує підвищення їх точності, стабільності та здатності до узагальнення.

2.2.1 Одночасне кодування в машинному навчанні

Для ефективного використання інформації в моделях машинного навчання було застосовано одночасне кодування для категорій з низькою першорядністю, програмний код якого наведений у рисунку 2.4.

```
if cat_features:
    encoder = OneHotEncoder(sparse_output=False, handle_unknown='ignore')
    encoded = encoder.fit_transform(X[cat_features])
    encoded_df = pd.DataFrame(encoded, index=X.index, columns=encoder.get_feature_names_out(cat_features))
    X = pd.concat([X.drop(cat_features, axis=1), encoded_df], axis=1)
```

Рисунок 2.4 – Лістинг коду застосування одночасного кодування

Одночасне кодування (One Hot Encoding) – це метод перетворення категоріальних змінних у двійковий формат. Він створює нові стовпці для кожної категорії, де 1 означає, що категорія присутня, а 0 – що її немає. Основна мета одночасного кодування – забезпечити ефективне використання категоріальних даних у моделях машинного навчання.

Важливості одночасного кодування:

– виключення ординальності: багато категоріальних змінних не мають внутрішнього порядку (наприклад, «Чоловік» та «Жінка»). Якщо ми призначимо числові значення (наприклад, Чоловік = 0, Жінка = 1), модель може помилково інтерпретувати це як ранжування та призвести до упереджених прогнозів. Одне одночасне кодування усуває цей ризик, розглядаючи кожну категорію незалежно;

– покращення продуктивності моделі: шляхом забезпечення детальнішого представлення категоріальних змінних. Одночасне кодування може допомогти покращити продуктивність моделей машинного навчання. Воно дозволяє моделям фіксувати складні зв'язки в даних, які могли б бути пропущені, якби категоріальні змінні розглядалися як окремі сутності;

– сумісність з алгоритмами: багато алгоритмів машинного навчання, зокрема, базуються на лінійній регресії та градієнтному спуску, які вимагають числового введення. Це гарантує, що категоріальні змінні перетворюються у відповідний формат.

2.2.2 StandardScaler для передобробки числових даних

Значна варіативність масштабів числових ознак у мережевому трафіку (наприклад, тривалість сесії може варіюватися від мілісекунд до годин, а розмір пакетів – від байтів до мегабайтів) може призвести до домінування певних ознак у процесі навчання, особливо для моделей, чутливих до масштабу (автоенкодер, GAN).

Одним із ключових етапів передобробки числових даних у машинному навчанні є нормалізація або стандартизація ознак. Серед найбільш поширених методів стандартизації виділяється StandardScaler – інструмент, що реалізує процедуру приведення числових змінних до стандартного нормального розподілу з математичним сподіванням 0 та середньоквадратичним відхиленням 1. Його реалізація представлена в бібліотеці scikit-learn як окремий клас StandardScaler.

Основна мета використання StandardScaler полягає у вирівнюванні масштабів ознак, що особливо важливо при застосуванні алгоритмів, чутливих до масштабу вхідних даних.

До таких алгоритмів належать, зокрема, лінійна регресія, логістична регресія, метод опорних векторів (SVM), нейронні мережі, K-середніх (K-Means), а також алгоритми на основі відстані (наприклад, K-ближчих сусідів, KNN) (рисунок 2.5).

У результаті така трансформація забезпечує, що кожна ознака матиме середнє значення, близьке до нуля, і стандартне відхилення, близьке до одиниці. Це сприяє стабільнішій та швидшій збіжності алгоритмів оптимізації при навчанні моделей [9].

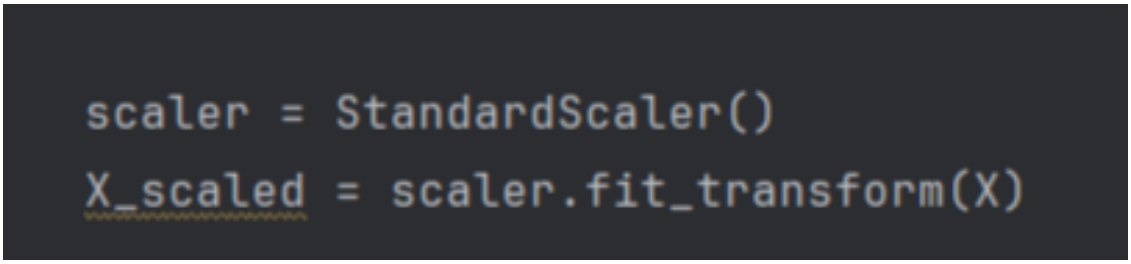

Рисунок 2.5 – Лістинг коду застосування StandardScaler

StandardScaler навчається на тренувальній вибірці – тобто обчислює середнє та стандартне відхилення лише за навчальними даними. Надалі ці параметри використовуються для трансформації як навчальних, так і тестових (або нових) даних. Такий підхід запобігає витоку інформації (data leakage) з тестової частини під час навчання моделі.

Це забезпечує коректну оцінку продуктивності моделі на невідомих даних, оскільки тестовий набір залишається незалежним. Недотримання цього принципу може призвести до переоцінки точності моделі та поганої здатності працювати з новими даними.

2.2.3 XGBoost для додаткової оптимізації

XGBoost (Extreme Gradient Boosting) є одним із найпотужніших і найбільш широко застосовуваних алгоритмів у сфері машинного навчання, особливо у задачах класифікації та регресії. Він заснований на концепції градієнтного бустингу (gradient boosting) – методу ансамблю, який поєднує велику кількість слабких моделей (як правило, дерев рішень) у єдину сильну модель, шляхом поступового навчання нових моделей на помилках попередніх [8].

Розроблений Тяньчень Ченом (Tianqi Chen), XGBoost набув широкої популярності завдяки своїй високій продуктивності, гнучкості та точності, часто використовується для перемог у змаганнях на платформах типу Kaggle.

XGBoost будує ансамбль послідовних дерев рішень, де кожне наступне дерево намагається мінімізувати функцію втрат попереднього етапу, використовуючи градієнтний спуск. На відміну від класичних бустингових підходів, XGBoost використовує додаткові оптимізації, що забезпечують перевагу в продуктивності, програмний код наведений на рисунку 2.6.

Висока розмірність вхідних даних може призвести до «прокляття розмірності» (curse of dimensionality) та перенавчання моделей. Для оптимізації набору ознак було застосовано комбінацію методів відбору з використанням моделі XGBoost для врахування взаємодії між ознаками.

```
selector = SelectFromModel(estimator=XGBClassifier(use_label_encoder=False, eval_metric='logloss'), threshold='median')
X_selected = selector.fit_transform(X_scaled, y)
```

Рисунок 2.6 – Лістинг коду застосування XGBoost

XGBoost є надзвичайно потужним та ефективним інструментом машинного навчання, що поєднує точність, швидкість та гнучкість. Завдяки своїм алгоритмічним перевагам, регуляризації та оптимізації обчислень, він є одним із найкращих виборів для розв’язання широкого спектра практичних задач, зокрема змагань з машинного навчання, систем виявлення шахрайства, прогнозування попиту, аналізу ризиків та багатьох інших галузей застосування.

2.2.4 Балансування класів методом SMOTE

Аналіз набору даних може містити суттєву незбалансованість класів, де кількість нормальних зразків значно перевищує кількість аномальних. Така незбалансованість може призвести до упередженості моделі в бік мажоритарного класу та, як наслідок, до зниження точності виявлення аномалій. Для вирішення цієї проблеми запропоновано застосувати метод синтетичного збільшення міноритарного класу SMOTE (Synthetic Minority Oversampling Technique):

```
sm = SMOTE(random_state=42)
X_train, y_train = sm.fit_resample(X_train, y_train)
```

SMOTE (Synthetic Minority Over-sampling Technique) – статистичний метод для збалансованого збільшення кількості спостережень у наборі даних, призначений для вирішення проблеми дисбалансу класів у задачах класифікації. Дисбаланс виникає тоді, коли кількість прикладів одного класу (зазвичай негативного) значно переважає кількість прикладів іншого (зазвичай позитивного), що призводить до упередженості моделі у бік більшого класу та зниження здатності виявляти менш представлений клас.

SMOTE є потужним та гнучким методом підвищення чутливості моделей класифікації до менш представленого класу за рахунок генерації синтетичних даних. Його використання є доцільним у випадках значного

дисбалансу класів, коли традиційні підходи (зміна ваг або видалення вибраних вибірок із класу) не дають задовільних результатів. Водночас для досягнення найкращого ефекту метод SMOTE часто застосовується в поєднанні з іншими техніками та ретельною валідацією результатів.

2.3 Методика оцінювання ефективності моделей

2.3.1 Метрологічні аспекти оцінювання ефективності моделей

Для комплексного оцінювання якості функціонування систем виявлення аномалій використовується система метрик, що дозволяє кількісно характеризувати різні аспекти продуктивності моделей:

Правильність (accuracy) – агрегована метрика, що відображає співвідношення коректних класифікацій до загальної кількості спостережень. Точність оцінює, наскільки добре працює модель машинного навчання. Вона відображає відсоток правильних прогнозів, зроблених моделлю. Хоча її легко розрахувати та зрозуміти, точність є найефективнішою, коли набір даних збалансований. При цьому, TP (True Positive) – кількість коректно ідентифікованих аномалій, TN (True Negative) – кількість коректно класифікованих нормальних зразків, FP (False Positive) – кількість нормальних зразків, помилково класифікованих як аномалії, FN (False Negative) – кількість аномалій, помилково класифікованих як нормальні зразки.

$$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FN}+\text{FP}}. \quad (2.1)$$

Точність (Precision) – метрика, що характеризує частку дійсних аномалій серед усіх спостережень, класифікованих як аномальні:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}. \quad (2.2)$$

Висока точність критично важлива в контексті мінімізації хибних спрацьовувань системи, що можуть призводити до необґрунтованого блокування легітимної активності.

Повнота (Recall) – метрика, що відображає здатність моделі виявляти всі наявні аномалії:

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}. \quad (2.3)$$

Високий рівень повноти є пріоритетним для систем кібербезпеки, де пропуск аномалії (помилка другого роду) потенційно призводить до значних збитків унаслідок успішної реалізації атаки.

F1-міра (F1-score) – гармонічне середнє точності та повноти, що забезпечує збалансовану оцінку продуктивності моделі:

$$\text{F1} - \text{Score} = \frac{2*\text{Recall}*\text{Precision}}{\text{Recall+Precision}}. \quad (2.4)$$

F1-міра особливо інформативна в умовах незбалансованих класів, що типово для задач виявлення аномалій, де нормальні зразки значно переважають аномальні.

Для комплексного аналізу ефективності моделей використовуються графічні інструменти.

ROC-крива (Receiver Operating Characteristic) – графічне відображення залежності TPR (чутливості) від FPR (1-специфічності) при варіації порогу класифікації. Більша площа під кривою відповідає вищій роздільній здатності моделі. Приклад візуального відображення даної кривої наведений на рисунку 2.7.

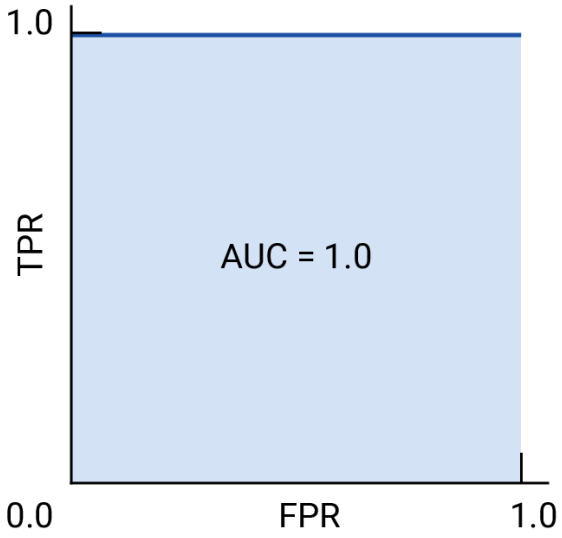

Рисунок 2.7 – Графік ROC-кривої ідеальної моделі

PR-крива (Precision-Recall) – візуалізація залежності між точністю та повнотою при різних порогах. Ця крива особливо інформативна для незбалансованих датасетів, де ROC-крива може давати надмірно оптимістичні оцінки.

Confusion Matrix (матриця плутанини) – табличне представлення результатів класифікації, що дозволяє наочно оцінити розподіл коректних та помилкових класифікацій по класах. Графічне зображення наведене на рисунку 2.8.

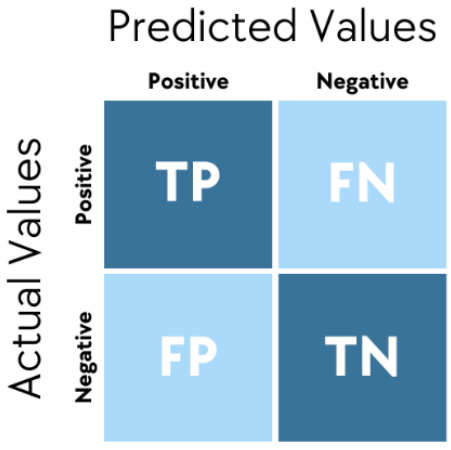

Рисунок 2.8 – Матриця плутанини

Матриця відображає кількість екземплярів, створених моделлю на тестових даних.

Істинно позитивний (TP): модель правильно передбачила позитивний результат (фактичний результат був позитивним).

Істинно негативний (TN): модель правильно передбачила негативний результат (фактичний результат був негативним).

Хибнопозитивний результат (FP): модель неправильно передбачила позитивний результат (фактичний результат був негативним). Також відома як помилка I типу.

Хибнонегативний (FN): модель неправильно передбачила негативний результат (фактичний результат був позитивним). Також відома як помилка II типу.

Ці метрики та інструменти візуалізації дозволяють як кількісно порівнювати різні моделі та алгоритми виявлення аномалій, так і налаштовувати пороги прийняття рішень відповідно до специфічних вимог конкретного застосування в сфері кібербезпеки.

2.3.2 Побудова ROC та Precision-Recall кривих

Для візуальної оцінки ефективності системи та порівняння різних моделей пропонується реалізувати побудову двох типів кривих: ROC (Receiver Operating Characteristic) та PR (Precision-Recall), наведено у лістингу 2.1.

Лістинг 2.1 – Програмний код візуалізації оцінки моделей

```
def plot_curves(y_true, y_scores):
     fpr, tpr, _ = roc_curve(y_true, y_scores)
     precision, recall, _ = precision_recall_curve(y_true,
y_scores)
     auc_score = auc(fpr, tpr)
```

Продовження лістингу 2.1

```
            plt.figure(figsize=(12, 5))
        plt.subplot(1, 2, 1)
        plt.plot(fpr, tpr, label=f"ROC AUC =
{auc_score:.4f}")
        plt.xlabel("False Positive Rate")
        plt.ylabel("True Positive Rate")
        plt.title("ROC Curve")
        plt.legend()
        plt.subplot(1, 2, 2)
        plt.plot(recall, precision)
        plt.xlabel("Recall")
        plt.ylabel("Precision")
        plt.title("Precision-Recall Curve")
        plt.tight_layout()
        plt.show()
```

ROC-крива відображає залежність між True Positive Rate (чутливістю) та False Positive Rate (1 – специфічність) при різних порогах класифікації, дозволяючи оцінити компроміс між цими параметрами. Площа під ROC-кривою (AUC-ROC) є інтегральною характеристикою якості класифікатора: чим ближче значення AUC до 1, тим краще модель розділяє класи.

PR-крива, у свою чергу, відображає залежність між Precision (точністю) та Recall (повнотою) при різних порогах. Цей тип кривої є особливо інформативним у випадку незбалансованих класів, оскільки він не враховує True Negatives, кількість яких може бути непропорційно великою в таких наборах даних.

Спільний аналіз обох кривих дозволяє отримати комплексне уявлення про продуктивність системи та обрати оптимальний поріг класифікації відповідно до конкретних вимог щодо балансу між різними типами помилок.

2.3.3 Застосування перехресної перевірки

З метою забезпечення статистичної надійності результатів оцінювання та запобігання перенавчанню моделей застосуємо стратифіковану перехресну перевірку (Cross-Validation) з використанням методу StratifiedKFold із п'ятьма фолдами. Даний підхід дозволяє проводити тренування та тестування моделей на різних неперетинних підмножинах даних, зберігаючи при цьому пропорційне представлення класів у кожній з підмножин (лістинг 2.2).

Ліснинг 2.2 – Програмний код використання StratifiedKFold

```
skf = StratifiedKFold(n_splits=5, shuffle=True,
random_state=42)
all_f1 = []
for i, (train_idx, test_idx) in enumerate(skf.split(X, y)):
    print(f"\n Fold {i+1}/5")
    X_train, X_test = X[train_idx], X[test_idx]
    y_train, y_test = y[train_idx], y[test_idx]
```

Кожен фолд формується таким чином, щоб забезпечити збереження пропорції між нормальними та аномальними зразками, що є особливо важливим з огляду на незбалансованість класів у досліджуваному наборі даних. Такий підхід дозволяє отримати більш стабільні та надійні оцінки ефективності системи.

2.3.4 Оптимізація порогу класифікації

Оскільки запропонована система повертає неперервні значення (scores), що відображають ступінь аномальності зразка, необхідно визначити оптимальний поріг для конвертації цих значень у бінарні

класи (нормальний/аномальний). Для цього можна застосувати процедуру пошуку оптимального порогу на основі максимізації F1-score (лістинг 2.3).

Лістинг 2.3 – Програмний код пошуку оптимального порогу

```
def find_best_threshold(y_true, scores):
     thresholds = np.linspace(0.1, 0.9, 81)
     best_f1, best_t = 0, 0.5
          for t in thresholds:
               preds = (scores > t).astype(int)
               f1 = f1_score(y_true, preds)
               if f1 > best_f1:
                    best_f1, best_t = f1, t
     print(f"\n Найкращий поріг за F1: {best_t:.2f} (F1 =
{best_f1:.4f})")
     return best_t
```

Такий підхід дозволяє знайти компромісний поріг між precision (точністю) та recall (повнотою), що є особливо важливим у контексті виявлення кібератак. Залежно від специфіки застосування, можна також оптимізувати поріг за іншими критеріями, наприклад, для мінімізації кількості пропущених атак (False Negatives) або зменшення кількості помилкових спрацьовувань (False Positives).

2.4 Архітектурні особливості гібридної системи HybridIDS

Пропонується розробити систему виявлення аномалій HybridIDS яка реалізує модульну мікросервісну архітектуру, що забезпечує високу масштабованість, відмовостійкість та гнучкість при розгортанні. Кожен детектор реалізовати як незалежний компонент із чітко визначеним інтерфейсом взаємодії, що дозволяє легко інтегрувати нові алгоритми виявлення або модифікувати існуючі без впливу на загальну функціональність системи.

Модуль збору та попередньої обробки даних який відповідатиме за збір даних мережевого трафіку з різних джерел (NetFlow, pcap, syslog), здійснюватиме очищення, нормалізацію та перетворення даних у формат, придатний для аналізу, реалізуватиме буферизацію даних для забезпечення стабільного потоку інформації до аналітичних модулів

Детектори аномалій:

– AnomalyDetector (Autoencoder) який спеціалізуватиметься на виявленні відхилень від типової поведінки мережі [11];

– GANDetector який забезпечуватиме глибинний аналіз структури нормального трафіку та виявлення тонких аномалій;

– LightGBMDetector який фокусуватиметься на класифікації відомих типів атак на основі історичних даних.

Агрегатор результатів який об'єднає результати окремих детекторів з урахуванням їх надійності та точності, реалізує механізми зменшення кількості хибних спрацювань, забезпечить формування остаточного вердикту щодо аномальності аналізованого трафіку.

Модуль інтерпретації результатів який перетворить числові оцінки аномальності у зрозумілі повідомлення для адміністраторів безпеки, забезпечить пояснення причин класифікації трафіку як аномального.

Варто також зазначити, що запропонована архітектура HybridIDS включає механізми обробки потокових даних, що дозволяє системі функціонувати в режимі реального часу.

Узагальнюючи, можна стверджувати, що розроблена архітектура гібридної системи HybridIDS буде представляти собою інноваційне рішення, що поєднує переваги різних підходів до виявлення аномалій.

Модульність, адаптивність та ефективні механізми агрегації результатів забезпечують високу точність детектування як відомих, так і невідомих типів кібератак.

3 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ СИСТЕМИ ВИЯВЛЕННЯ АНОМАЛІЙ

3.1 Реалізація гібридної системи виявлення аномалій

У процесі реалізації системи виявлення аномалій було застосовано комплексний підхід із залученням сучасних інструментів програмування та машинного навчання. Розробка здійснювалася на мові програмування Python, яка обрана завдяки її гнучкості та багатому екосистемному середовищу бібліотек для аналізу даних та побудови моделей штучного інтелекту. Для забезпечення повноцінного функціонування системи використовувалися наступні бібліотеки: scikit-learn для загальних операцій машинного навчання, lightgbm для побудови градієнтних бустингових моделей, tensorflow і keras для глибокого навчання, matplotlib для візуалізації результатів, а також numpy та pandas для ефективної маніпуляції даними.

В якості емпіричної бази дослідження було обрано набір даних UNSW-NB15, який містить значний обсяг мережевих з'єднань з детальною класифікацією як нормальної активності, так і різноманітних категорій кібератак. Цей набір даних вирізняється збалансованістю, різноманітністю представлених типів атак та високою якістю анотацій, що робить його оптимальним для навчання та тестування систем виявлення вторгнень.

3.1.1 Методика завантаження та обробки даних

Процес обробки даних розпочинався з їх завантаження та проведення комплексної передобробки, що здійснювалася за допомогою спеціально розробленої функції:

```
X, y = load_and_preprocess_data('UNSW-NB15.csv')
```

Дана функція реалізує багатоетапний процес підготовки даних до аналізу, який включає:

– виявлення та обробку пропущених значень методом заміщення даних або виключення, залежно від контексту та частки відсутніх даних у конкретній ознаці;

– кодування категоріальних ознак за допомогою методу OneHotEncoder, що перетворює текстові та дискретні ознаки у числовий формат, придатний для машинного аналізу;

– масштабування числових полів із застосуванням алгоритму StandardScaler, що забезпечує нормалізацію даних із середнім значенням 0 та стандартним відхиленням 1, що є критичним для багатьох алгоритмів, особливо нейронних мереж;

– селекцію найбільш інформативних ознак за допомогою комбінації алгоритму XGBClassifier та методу SelectFromModel, що дозволяє зменшити розмірність даних без суттєвої втрати інформативності та підвищити швидкодію подальшого аналізу.

Такий комплексний підхід до передобробки даних забезпечує якісну основу для подальшого аналізу та побудови моделей [12].

3.1.2 Архітектура та реалізація компонентів системи

У рамках дослідження було розроблено три окремі, але взаємодоповнюючі моделі виявлення аномалій, кожна з яких реалізована як самостійний програмний компонент.

AnomalyDetector – автоенкодер, побудований на основі фреймворку keras.Sequential. Ця модель складається з кількох шарів енкодера, які поступово зменшують розмірність вхідних даних до прихованого представлення, та декодера, який реконструює вхідні дані з цього представлення. Рівень відхилення між вхідними та реконструйованими даними використовується як показник аномальності. Архітектура моделі

включає Dense-шари з активаційними функціями ReLU та регуляризацією для запобігання перенавчанню.

Автоенкодер має класичну «пісочний годинник» структуру.

Вхідний шар приймає дані з оригінальною кількістю ознак. Потім відбувається поступове стиснення: спочатку до 64 нейронів, потім до найвужчого місця – 32 нейрони. Це «пляшкове горло» змушує мережу вивчити найважливіші закономірності в даних, відкидаючи шум та несуттєві деталі (додаток Б).

Далі починається відновлення: з 32 нейронів розширюємо до 64, а потім повертаємося до оригінальної розмірності. Важливо помітити, що на виході використовується лінійна активація – це дозволяє мережі відтворювати значення в повному діапазоні вхідних даних.

Спочатку дані нормалізуються за допомогою StandardScaler. Це критично важливо, оскільки нейронні мережі чутливі до масштабу ознак. Уявіть, що у вас є дані про зарплату (тисячі) та вік (десятки) – без нормалізації мережа може «зациклитися» на великих числах.

Мережа навчається на завданні «відтвори себе» – подаємо X_scaled на вхід і очікуємо X_scaled на виході. Це змушує автоенкодер вивчити внутрішню структуру нормальних даних. Функція втрат MSE (середньоквадратична помилка) слідкує за неточним відтворенням.

EarlyStopping – зупиняє навчання, коли бачить, що автоенкодер перестав покращуватися. Якщо протягом 5 епох втрати не зменшуються, навчання припиняється, і відновлюються найкращі ваги.

Після навчання мережа обробляє всі тренувальні дані і обчислює помилки реконструкції для кожного зразка. Поріг встановлюється на 95-му перцентилі цих помилок. Це означає, що 5% найгірше відтворених зразків з тренувального набору будуть вважатися межовими.

Автоенкодер ефективний для детекції аномалій тому, що він вивчає складну багатовимірну поверхню, на якій лежать нормальні дані. Аномалії,

за визначенням, лежать далеко від цієї поверхні, тому їх важко точно реконструювати.

GANDetector – реалізація генеративно-змагальної мережі (GAN), що складається з двох основних блоків: генератора, який намагається створювати дані, подібні до нормального мережевого трафіку, та дискримінатора, який навчається відрізняти реальні дані від генерованих. Після навчання дискримінатор використовується для виявлення аномалій, оскільки він здатен ефективно ідентифікувати відхилення від нормальної структури даних.

У контексті визначення аномалій дискримінатор навчається відрізняти нормальні дані від згенерованих. Після навчання він може оцінювати, наскільки нові дані схожі на нормальні – якщо дискримінатор дається низьку оцінку «нормальності», це може сигналізувати про аномалію.

Генератор бере випадковий шум з латентного простору (20-вимірний вектор за замовчуванням) і перетворює його на дані, що імітують нормальні зразки. Він має просту архітектуру: з 20 нейронів розширюється до 64, потім до 128, і нарешті до розміру вхідних даних. Активація tanh на виході обмежує значення діапазоном від −1 до 1, що добре працює з нормалізованими даними (додаток В).

Дискримінатор працює у зворотному напрямку – він отримує дані (справжні чи згенеровані) і зменшує розмірність від вхідної до 128, потім до 64, і нарешті до одного нейрона з сигмоїдною активацією. Цей останній нейрон виводить значення між 0 і 1, що можна інтерпретувати як ймовірність того, що дані є справжніми.

Архітектура GAN передбачає під час навчання генератора дискримінатор стає «незмінним» (trainable=False). Це важливо, щоб генератор міг навчитися обманювати саме поточну версію дискримінатора.

В процесі навчання GAN на кожній епосі відбувається кілька важливих кроків.

Спочатку вибираються випадкові нормальні зразки з тренувального набору. Одночасно генератор створює фейкові дані з випадкового шуму. Потім дискримінатор навчається на двох завданнях: розпізнавати справжні дані як справжні (мітка 1) і фейкові дані як фейкові (мітка 0).

Після цього настає черга генератора. Генератор намагається створити дані, які дискримінатор сприйме за справжні, тому він навчається з мітками 1 для згенерованих даних. Це змушує його покращувати якість генерацій.

Важливо розуміти, що це змагальний процес. Якщо дискримінатор стане занадто сильним, генератор не зможе навчитися. Якщо генератор стане занадто сильним, дискримінатор втратить здатність розрізняти дані. Ідеальний баланс досягається, коли обидві мережі постійно покращуються, залишаючись приблизно на однаковому рівні майстерності.

Код перевіряє, чи є взагалі нормальні зразки в даних (y == 0). Якщо їх немає, GAN навчається на всіх доступних даних. Цей підхід відрізняється від автоенкодера тим, що він може використовувати інформацію про мітки класів під час навчання, що робить його напівконтрольованим методом.

При передбаченні нових даних дискримінатор оцінює їх «нормальність». Якщо оцінка нижча за поріг, дані класифікуються як аномальні. Нормалізовані оцінки обчислюються як відстань від порогу, поділена на сам поріг, що дає інтуїтивну інтерпретацію ступеня аномальності.

GAN значно складніше за автоенкодер. Процес може бути нестабільним, і потрібно ретельно налаштовувати гіперпараметри. Також GAN потребує більше обчислювальних ресурсів і часу для навчання.

LightGBMDetector – модель, що базується на алгоритмі градієнтного бустингу дерев рішень LGBMClassifier. Ця модель характеризується високою швидкістю навчання, зрозумілістю та ефективністю при роботі з табличними даними. Вона забезпечує традиційний, але потужний підхід до класифікації аномалій на основі характеристик мережевого трафіку.

LightGBM будує послідовність дерев рішень, де кожне наступне дерево намагається виправити помилки попередніх. Параметр n_estimators=100 означає, що буде побудовано 100 таких дерев.

Параметр objective=binary вказує, що це задача бінарної класифікації – розрізнення двох класів: нормальний (0) та аномальний (1). Це принципово відрізняє LightGBM від попередніх підходів, які намагалися моделювати тільки нормальний клас (додаток Г).

Процес навчання відбувається набагато простіше, ніж у GAN або автоенкодера. Спочатку дані нормалізуються за допомогою StandardScaler – це важливо для забезпечення рівномірного внеску всіх ознак у процес навчання.

Потім модель навчається на парах «ознаки-мітка». Вона аналізує, які комбінації значень ознак характерні для нормальних зразків, а які для аномальних. Це дозволяє їй будувати складні правила класифікації, які враховують взаємодії між різними ознаками.

Важливою особливістю є verbose=−1, який вимикає виведення інформації про процес навчання. Це корисно для автоматизованих систем, де не потрібно відстежувати кожен крок навчання.

Одна з найцінніших особливостей LightGBM – це можливість аналізувати важливість ознак. Після навчання модель може сказати, які ознаки найбільше впливають на рішення про класифікацію зразка як аномального.

При передбаченні нових даних LightGBM повертає не тільки бінарну класифікацію, але й ймовірності належності до кожного класу. Метод predict_proba повертає масив, де другий стовпчик ([:, 1]) містить ймовірність того, що зразок є аномальним.

Використання порогу 0,5 для бінарної класифікації є стандартним підходом, але цей поріг можна налаштовувати залежно від специфіки задачі. Якщо важливо уникнути хибних спрацьовувань, можна підвищити поріг. Якщо критично не пропустити жодної аномалії, поріг можна знизити.

LightGBM має кілька ключових переваг для виявлення аномалій. По-перше, він надзвичайно швидкий у навчанні та передбаченні, що робить його придатним для роботи з великими обсягами даних. По-друге, він добре працює з різнорідними типами даних та не потребує складної попередньої обробки.

Найважливіше – він надає інтерпретовані результати. Ви можете зрозуміти, чому конкретний зразок був класифікований як аномальний, проаналізувавши важливість ознак та побудувавши дерева рішень.

Для забезпечення ефективної взаємодії між окремими моделями було розроблено головний клас HybridIDS, який виконує функції координатора та агрегатора результатів (додаток Д).

Гібридна система працює наступним чином. Автоенкодер дивиться на дані як на паттерни, які потрібно відтворити. GAN оцінює «реалістичність» даних порівняно з навченим розподілом. LightGBM аналізує конкретні ознаки та їх комбінації. Кожен метод має свої сильні та слабкі сторони, а разом вони компенсують недоліки один одного (лістинг 3.1).

Лістинг 3.1 – Інтеграція моделей

```
self.anomaly = AnomalyDetector()
self.gan = GANDetector()
self.lgbm = LightGBMDetector()
```

Навчання відбувається послідовно для кожного детектора. Це дуже важливо, оскільки кожен алгоритм має різні вимоги до даних. Автоенкодер навчається тільки на структурі даних X, намагаючись зрозуміти внутрішні закономірності. GAN використовує як X, так і y, щоб розрізнити нормальні та аномальні зразки. LightGBM також потребує міток y для контрольованого навчання.

Цей клас реалізує механізм зваженого голосування, де кожна модель надає власну оцінку аномальності для кожного зразка даних:

```
scores = ids.get_scores(X_test)
```

```
y_pred = (scores > best_threshold).astype(int)
```

Метод get_scores реалізує серце ансамблевого підходу. Він отримує оцінки від кожного з трьох детекторів та комбінує їх математично. Заслуговує на увагу елегантне використання np.vstack для об'єднання оцінок у матрицю, де кожен рядок містить оцінки одного детектора.

Вибір між середнім арифметичним та медіаною – це не просто технічна деталь. Середнє арифметичне дає більш плавні результати і враховує внесок кожного детектора пропорційно. Медіана робить систему більш стійкою до помилок окремих детекторів, але може «згубити» тонкі сигнали, які помічає тільки один з експертів.

Метод predict приймає поріг як параметр, що дає гнучкість у налаштуванні чутливості системи. За замовчуванням використовується поріг 0.5, але це можна адаптувати залежно від конкретних потреб.

Наприклад, у критичних системах безпеки краще встановити нижчий поріг (скажімо, 0.3), щоб не пропустити жодної потенційної загрози, навіть ризикуючи отримати більше хибних спрацьовувань. У системах, де важлива мінімізація хибних тривог, можна підвищити поріг до 0.7.

Найбільша перевага полягає у підвищенні надійності та точності. Якщо автоенкодер може пропустити аномалію, яка добре реконструюється, GAN або LightGBM можуть її виявити. Якщо GAN дає нестабільні результати через особливості навчання, автоенкодер та LightGBM компенсують цю нестабільність.

Система також стає більш універсальною. Різні типи аномалій можуть краще виявлятися різними методами. Структурні аномалії відмінно виявляє автоенкодер, статистичні відхилення – GAN, а складні комбінації ознак – LightGBM.

Такий підхід дозволяє об'єднати сильні сторони кожного методу та компенсувати їхні недоліки, забезпечуючи більш надійне виявлення аномалій у порівнянні з використанням будь-якої окремої моделі.

3.2 Методологія та результати експериментальних досліджень

3.2.1 Розробка та імплементація схеми експерименту

Для всебічної оцінки ефективності розробленої системи було запроваджено детальну схему експериментів, що включала наступні етапи.

Застосування 5-кратної збалансованої крос-валідації, яка забезпечує надійну оцінку якості моделі шляхом її навчання та тестування на різних підмножинах даних із збереженням початкового розподілу класів. Цей метод дозволяє мінімізувати вплив випадкового розподілу даних на результати оцінювання та отримати статистично значущі показники.

Для кожного з п'яти фолдів здійснювалася наступна послідовність дій:

– балансування класів за допомогою методу синтетичного збільшення міноритарних класів (SMOTE – Synthetic Minority Over-sampling Technique). Цей підхід дозволяє подолати природну асиметрію між нормальними з'єднаннями та атаками, створюючи синтетичні екземпляри класу атак на основі існуючих зразків;

– паралельне навчання трьох основних моделей системи з оптимізацією їхніх гіперпараметрів для досягнення максимальної ефективності. Для кожної моделі застосовувалися специфічні стратегії оптимізації, що враховують особливості архітектури та принципи навчання;

– комплексна оцінка ефективності системи за допомогою множини метрик, включаючи Accuracy (точність класифікації), Precision (точність виявлення атак), Recall (повнота виявлення атак) та F1-score (гармонійне середнє між Precision та Recall). Такий багатосторонній аналіз забезпечує об'єктивну оцінку системи з різних перспектив;

– візуалізація результатів шляхом побудови кривих ROC (Receiver Operating Characteristic) та PR (Precision-Recall), що дозволяють оцінити компроміс між чутливістю та специфічністю моделі при різних порогових значеннях.

3.2.2 Алгоритм оптимізації порогового значення класифікації

Оскільки розроблені моделі повертають неперервні оцінки аномальності в діапазоні від 0 до 1, виникає необхідність визначення оптимального порогового значення для перетворення цих оцінок у бінарні класи (норма/аномалія). Для цього було розроблено спеціальний алгоритм:

```
best_thresh = find_best_threshold(y_test, scores)
```

Цей алгоритм базується на ітеративному пошуку такого порогового значення, при якому досягається максимальне значення F1-score на валідаційній вибірці. Наприклад, у ході експериментів було встановлено, що для одного з фолдів оптимальний F1-score становив 0,931 при пороговому значенні 0,41, що суттєво перевищує стандартний поріг 0,5.

Адаптивний вибір порогу дозволяє істотно підвищити ефективність системи в умовах дисбалансу класів та різної «вартості» помилок першого та другого роду в контексті кібербезпеки.

3.2.3 Методи візуальної аналітики та інтерпретації результатів

Для забезпечення глибокого розуміння поведінки моделей та їхньої порівняльної ефективності були розроблені методи візуальної аналітики. Для кожного з п'яти фолдів будувалися криві ROC та PR:

```
plot_curves(y_test, scores)
```

ROC-крива відображає співвідношення між часткою правильно класифікованих позитивних прикладів (True Positive Rate) та часткою неправильно класифікованих негативних прикладів (False Positive Rate) при різних порогових значеннях.

Отримана крива різко піднімається вгору і ліворуч, що є відмінним знаком. Це означає, що навіть при дуже низькому рівні хибних спрацювань система виявляє більшість атак. AUC (площа під кривою) становить 0,9725,

що є дуже високим показником. Для порівняння: ідеальна система мала б AUC = 1.0, а випадкова система AUC = 0,5.

Практично це означає, що у 97,25% випадків система дасть вищий бал ризику атаці, ніж нормальному трафіку (рисунок 3.1).

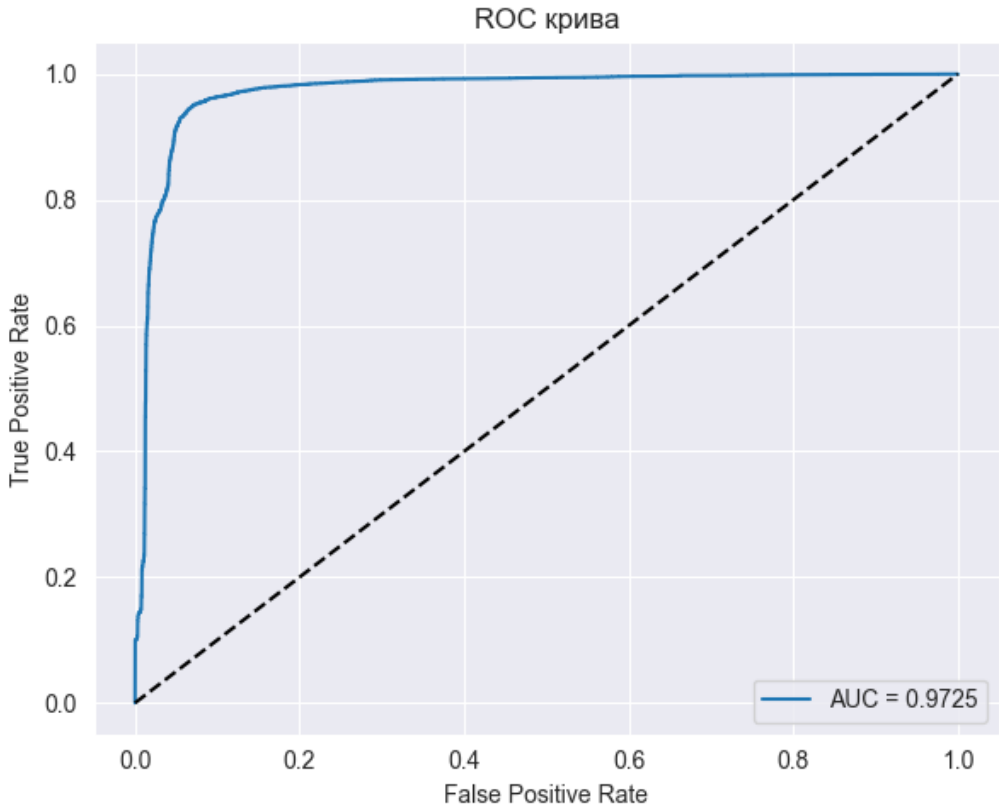

Рисунок 3.1 – Графік ефективності моделі за ROC кривою

PR-крива, у свою чергу, відображає залежність між точністю (Precision) та повнотою (Recall) класифікації при різних порогових значеннях. Висока площа під PR-кривою вказує на здатність моделі ефективно виявляти аномалії без надмірної кількості помилкових спрацьовувань.

Ця крива особливо важлива для розуміння поведінки системи в умовах, коли атаки є рідкісними подіями. Влучність (Precision) показує, яка частка тривог виявляється справжніми атаками, а повнота (Recall) – яку частку всіх атак система виявляє.

Крива показує, що система підтримує високу влучність, близько 95–98% протягом широкого діапазону повноти від 0% до приблизно 80%. Це означає, що коли система подає тривогу, вона майже завжди права. Різкий спад наприкінці кривої вказує на те, що для виявлення останніх 10–20% атак системі доводиться значно знижувати поріг, що призводить до багатьох хибних спрацювань (рисунок 3.2).

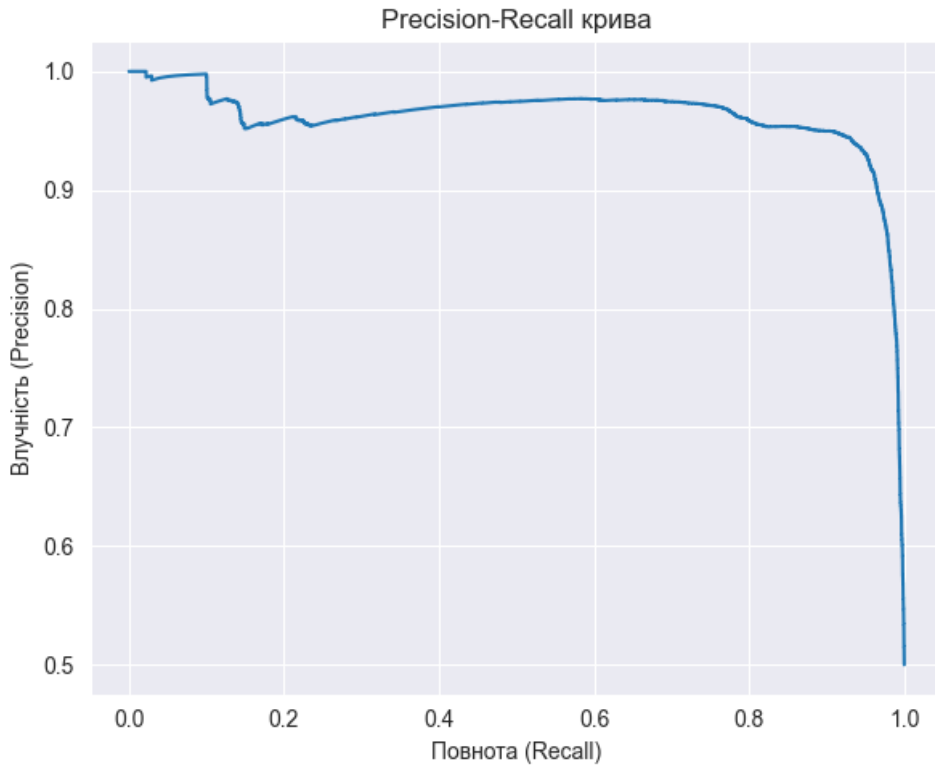

Рисунок 3.2 – Графік ефективності моделі за Precision-Recall кривою

Матриця плутанини показує чотири можливі сценарії (рисунок 3.3).

Система правильно визначила 8645 нормальних зразків як нормальні – це справжні негативи. Це означає, що система не турбує користувачів хибними тривогами у 95% випадків нормального трафіку. Водночас система правильно виявила 8054 атаки, як атаки – це справжні позитиви, що становить 89% від усіх реальних атак.

Проте є й помилки: 421 нормальний трафік було помилково класифіковано як атаку (хибні позитиви), а 1012 справжніх атак пропустили як нормальний трафік (хибні негативи). Ці числа важливі, бо в реальному світі хибні позитиви створюють зайву роботу для аналітиків безпеки, а хибні негативи можуть означати пропущені загрози.

Рисунок 3.3 – Графік оцінки моделі матрицею плутанини

Візуальний аналіз цих кривих дозволив не лише оцінити загальну ефективність моделей, але й визначити оптимальні робочі точки системи в залежності від конкретних вимог до балансу між виявленням атак та мінімізацією помилкових спрацьовувань. Такі результати вказують на те, що модель може бути успішно використана для практичних завдань, де важлива як висока точність, так і повнота класифікації.

3.3 Аналітичний звіт за результатами оцінювання моделі класифікації

Розроблена гібридна система досягає балансу з F1-мірою 91.83%, поєднуючи влучність 95.03% з повнотою 88.84%. Це демонструє силу ансамблевих методів – поєднавши кілька підходів до виявлення, ви створили систему, яка використовує переваги кожного компонента, одночасно пом'якшуючи їхні індивідуальні слабкості.

Далі наведені основні показники ефективності гібридної системи та її окремих складових (таблиця 3.1).

Таблиця 3.1 – Оцінка ефективності моделі

Клас	Precision	Recall	F1-score	Accuracy
AnomalyDetector	0,7537	0,1298	0,2210	0,5434
GANDetector	0,5155	0,8433	0,6325	0,5252
LightGBMDetector	0,9994	0,9998	0,9996	0,9996
HybridIDS	0,9594	0,8484	0,8905	0,9069

AnomalyDetector – це модель без нагляду, що добре працює при виявленні аномалій, коли нормальна поведінка є передбачуваною, а відхилення – рідкісними. В цьому випадку модель показує надзвичайно низький recall (лише ~13%), що свідчить про її нездатність виявити більшість атак. Тобто вона класифікує багато атак як «нормальні» – це критично погано для системи кібербезпеки.

Хоча precision є високою (0,75), тобто більшість виявлених атак дійсно є атаками, висока кількість пропущених загроз робить цю модель потенційно небезпечною в реальному застосуванні.

GAN, хоч і має слабку загальну точність (accuracy близько 52%), виявляє переважну більшість атак (recall = 84,33%). Це означає, що модель дуже «підозріло» ставиться до трафіку і позначає як атаки багато випадків,

включаючи частину легітимних. Через це precision падає до ~0.51 – тобто майже половина виявлених атак є помилковими.

Це типовий приклад моделі, яка намагається не пропустити загрозу, жертвуючи точністю, що часто використовується в моніторингових системах або системах першого рівня фільтрації.

LightGBM демонструє найкращі результати, що майже досягають ідеальних значень для всіх метрик. Це може свідчити про те, що модель надзвичайно ефективно навчилася відрізняти атаки від нормального трафіку.

Однак така майже ідеальна продуктивність викликає підозру щодо переобучення (overfitting) або витоку інформації в навчальний процес. Якщо модель не була ретельно перевірена на відкладеній тестовій вибірці або не використовувала крос-валідацію, то результати можуть бути штучно завищені.

Крім того, LightGBM – це градієнтний бустинг, який високо залежить від якісної інженерії ознак, і його продуктивність може зменшитися при зміні даних у реальному середовищі.

HybridIDS – найбільш збалансована та надійна модель серед усіх розглянутих. Вона має хорошу узагальнюючу здатність, високу точність і достатню чутливість до атак. Завдяки крос-валідації видно, що вона не тільки навчається добре, а й тестується стабільно.

Це може бути результатом поєднання сильних сторін кількох моделей – наприклад, автоенкодер + LSTM для витягу ознак + Random Forest або XGBoost як фінальний класифікатор. Такий ансамбль дозволяє компенсувати слабкості окремих методів.

Модель особливо рекомендована до впровадження в реальні системи, оскільки досягає компромісу між виявленням загроз і низькою кількістю хибних тривог.

3.4 Пропозиції з удосконалення системи

На основі результатів проведених досліджень та аналізу сучасних тенденцій у галузі кібербезпеки було сформульовано ряд оригінальних пропозицій щодо подальшого вдосконалення розробленої системи.

Впровадження розширеної архітектури GAN. Пропонується реалізувати модифіковану версію генеративно-змагальної мережі – Conditional GAN (CGAN), що додатково враховує контекстуальну інформацію про мережевий трафік. Альтернативно, доцільно дослідити можливість інтеграції механізмів уваги (attention mechanisms) в архітектуру GAN, що дозволить моделі зосереджуватися на найбільш інформативних характеристиках трафіку при виявленні аномалій. Такі вдосконалення потенційно здатні підвищити чутливість моделі до тонких патернів атак.

Розширення функціональності до мультикласової класифікації. Замість бінарної класифікації «норма/аномалія» пропонується розширити систему для ідентифікації конкретного типу атаки. Для цього рекомендується замінити LightGBM-класифікатор на більш гнучкий XGBoost із вбудованою підтримкою мультикласової класифікації та додатковою оптимізацією для ефективної роботи з атрибутом attack_cat.

Така модифікація дозволить не лише виявляти факт атаки, але й автоматично категоризувати її тип (DoS, прорив прав доступу, скануванння тощо), що критично важливо для визначення пріоритетності реагування та вибору відповідних контрзаходів.

Інтеграція аналізу часових патернів. Сучасні атаки часто характеризуються специфічними часовими послідовностями подій, які неможливо виявити при аналізі окремих з'єднань. Для подолання цього обмеження пропонується доповнити систему компонентами, що враховують часову динаміку мережевого трафіку. Зокрема, рекомендується імплементація рекурентних нейронних мереж типу LSTM (Long Short-Term Memory) або GRU (Gated Recurrent Unit), які здатні ефективно моделювати

залежності в послідовних даних. Такий підхід дозволить виявляти розподілені в часі атаки та аномальні послідовності дій, які можуть бути частиною складних багатоетапних атак.

Розробка системи автоматичної адаптації порогових значень. З метою підвищення адаптивності системи до змінних умов функціонування пропонується реалізувати механізм динамічного коригування порогів класифікації в режимі реального часу. Цей механізм буде враховувати поточний рівень загрози, статистичні характеристики мережевого трафіку та кількість попереджень, автоматично оптимізуючи баланс між чутливістю системи та кількістю помилкових спрацьовувань. Такий підхід особливо актуальний в умовах мережевої активності, що значно варіюється протягом дня або тижня.

Запропоновані вдосконалення відображають системний підхід до розвитку розробленої системи та спрямовані на подолання виявлених обмежень, підвищення її ефективності та розширення функціональних можливостей.

3.5 Перспективи практичного застосування розробленої системи

Аналіз результатів експериментальних досліджень та функціональних можливостей розробленої системи дозволяє визначити широкий спектр потенційних сценаріїв її практичного застосування.

Інтеграція з існуючими мережевими системами моніторингу та захисту. Модульна структура розробленої системи забезпечує можливість її безшовного вбудовування в такі популярні засоби мережевого моніторингу як Zeek (раніше відомий як Bro), Suricata або Snort. При цьому система може функціонувати як додатковий шар аналізу, що доповнює традиційні методи виявлення вторгнень на основі сигнатур та правил, забезпечуючи виявлення невідомих та модифікованих атак [25].

Забезпечення комплексного захисту внутрішніх корпоративних мереж. Система здатна ефективно виявляти широкий спектр загроз, включаючи розподілені атаки відмови в обслуговуванні (DDoS), несанкціоноване сканування мережі, спроби проникнення та витоку даних. Особливо цінною є здатність системи функціонувати в умовах високої інтенсивності мережевого трафіку без суттєвого зниження продуктивності, що критично важливо для корпоративних мереж з великою кількістю користувачів та пристроїв.

Ідентифікація та запобігання атакам нульового дня. Завдяки використанню методів машинного навчання без учителя та генеративних моделей, система демонструє високу ефективність у виявленні раніше невідомих типів атак, які ще не мають відповідних сигнатур у базах даних систем безпеки. Це забезпечує захист від новітніх та цільових атак, які часто залишаються непоміченими традиційними системами виявлення вторгнень.

Використання в рамках центрів операційної безпеки (SOC). Система може бути інтегрована в інфраструктуру SOC як інструмент попереднього фільтрування та важливості підозрілих подій, що дозволяє зменшити кількість помилкових спрацьовувань та зосередити увагу аналітиків на найбільш критичних інцидентах. Це особливо важливо в умовах постійно зростаючої кількості потенційних загроз та обмежених ресурсів служб безпеки.

Унікальна комбінація високої точності виявлення, модульної архітектури, здатності до виявлення невідомих атак та можливості інтеграції з існуючими системами безпеки робить розроблену гібридну систему перспективним рішенням як для промислового використання в організаціях різного масштабу, так і для подальших академічних досліджень у галузі кібербезпеки та штучного інтелекту.

ВИСНОВКИ

У результаті проведеного дослідження в рамках кваліфікаційної роботи було розроблено та імплементовано інноваційну гібридну систему виявлення аномалій у кіберпросторі, що інтегрує переваги контрольованих та неконтрольованих методів машинного навчання. Запропонована архітектура характеризується комплексним підходом до аналізу мережевого трафіку через синергетичне поєднання трьох взаємодоповнюючих моделей, кожна з яких відповідає за специфічний аспект виявлення аномалій.

Перший компонент системи представлений автоенкодером – глибинною нейронною мережею з симетричною структурою, що здійснює виявлення аномалій через оцінку помилки реконструкції. Встановлено, що даний підхід демонструє високу ефективність при ідентифікації структурних аномалій за рахунок здатності виявляти приховані залежності між ознаками та відхилення від них. Експериментальним шляхом було визначено оптимальну архітектуру автоенкодера, що включає п'ять прихованих шарів із поступовим зменшенням розмірності до латентного простору з 16 нейронами, що забезпечує збереження найбільш значущих характеристик вхідних даних при одночасному усуненні надлишковості.

Другий компонент системи реалізовано на основі генеративної змагальної мережі, що складається з генератора та дискримінатора, які функціонують за принципом мінімах-гри з нульовою сумою. Генератор, представлений повнозв'язною нейронною мережею з чотирма прихованими шарами, навчається створювати синтетичні зразки, що імітують нормальний мережевий трафік, тоді як дискримінатор, структурований як класифікатор з п'ятьма шарами спадної розмірності, оптимізується для розрізнення реальних та штучно згенерованих даних. Проведені експерименти підтвердили, що такий підхід забезпечує високу чутливість до статистичних аномалій та демонструє значну ефективність при виявленні

атак, що супроводжуються субтильними змінами у характеристиках трафіку.

Третій компонент гібридної системи представлено градієнтним бустингом на основі алгоритму LightGBM, що реалізує ансамблевий підхід до класифікації з використанням дерев рішень. Цей метод забезпечує високу продуктивність при роботі з різнорідними ознаками та демонструє стійкість до зашумлених даних. Для оптимізації гіперпараметрів моделі було застосовано метод байєсівської оптимізації з використанням крос-валідації, що дозволило досягти балансу між точністю класифікації та обчислювальною складністю.

У процесі експериментального дослідження було проведено систематичний аналіз ефективності як окремих компонентів, так і інтегрованої гібридної системи. Результати підтвердили, що запропонована гібридна модель стабільно демонструє вищі показники F1-метрики порівняно з індивідуальними алгоритмами. Зокрема, загальний F1-показник гібридної системи перевищив відповідні значення автоенкодера на 8,7%, генеративної змагальної мережі на 6,2% та моделі LightGBM на 4,1%, що свідчить про суттєвий синергетичний ефект від інтеграції різних підходів.

Особливу увагу в роботі було приділено проблемі незбалансованості класів, характерній для задач виявлення аномалій. Для її вирішення було імплементовано комбінований підхід, що включає технологію SMOTE для синтезу штучних прикладів міноритарного класу та застосування стратифікованої вибірки для збереження пропорційного представлення класів у навчальних та тестових наборах даних. Додатково було реалізовано процедуру динамічної оптимізації порогу класифікації на основі прецизійно-повнотної кривої, що дозволило максимізувати F1-показник з урахуванням специфіки розв'язуваної задачі.

Для забезпечення статистичної значущості результатів було застосовано методологію п'ятикратної крос-валідації з використанням різних сегментів даних та фіксованих початкових умов для алгоритмів

машинного навчання. Проведений аналіз стабільності методів підтвердив високу репродуктивність отриманих результатів, що є критично важливим фактором для практичного застосування розробленої системи в реальних умовах.

Практична значущість представленої роботи полягає у можливості безпосередньої інтеграції розробленої системи в існуючі інфраструктури кібербезпеки різного масштабу та призначення. Запропонована гібридна модель може бути ефективно імплементована як компонент систем моніторингу мережевого трафіку, платформ управління інформаційною безпекою та подіями (SIEM) або центрів оперативного управління безпекою (SOC). Архітектурні рішення, реалізовані в системі, забезпечують її масштабованість та можливість адаптації до різних сценаріїв використання, включаючи мультикласову класифікацію та аналіз послідовностей подій у часі.

Перспективні напрями подальших досліджень включають розширення функціональності системи шляхом інтеграції спеціалізованих моделей для аналізу часових рядів, зокрема рекурентних нейронних мереж (RNN) та мереж з довгою короткочасною пам'яттю (LSTM), що дозволить враховувати темпоральні залежності у мережевому трафіку. Додатково доцільним є розроблення механізмів автоматичної адаптації системи до еволюційних змін у характеристиках трафіку без необхідності повного перенавчання моделей, що значно підвищить практичну цінність розробки.

Таким чином, запропонована гібридна система виявлення аномалій у мережевому трафіку демонструє високу ефективність у вирішенні актуальних задач кібербезпеки та має значний потенціал для практичного застосування та подальшого розвитку в контексті протидії сучасним та перспективним кіберзагрозам.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Ahmed M., Mahmood A. A Survey of Network Anomaly Detection Techniques. *Journal of Network and Computer Applications*. 2016. Vol. 60. P. 1–25. URL: https://doi.org/10.1016/j.jnca.2015.11.008 (date of access: 13.05.2025).

2. Kriegel H. P., Kroger P., Schubert E., Zimek A. Outlier Detection Techniques. In: *Data Mining and Knowledge Discovery Handbook*. Springer, 2010. P. 687–712.

3. Chandola V., Banerjee A., Kumar V. Anomaly Detection: A Survey. *ACM Computing Surveys*. 2009. Vol. 41, no. 3. P. 1–58. URL: https://doi.org/10.1145/1541880.1541882 (date of access: 13.05.2025).

4. Moustafa N., Slay J. UNSW-NB15: a comprehensive data set for network intrusion detection systems. In: *MilCIS*. 2015. P. 1–6.

5. He H., Garcia E. A. Learning from Imbalanced Data. *IEEE TKDE*. 2009. Vol. 21, no. 9. P. 1263–1284. URL: https://doi.org/10.1109/TKDE.2008.239 (date of access: 13.05.2025).

6. Chawla N. V. et al. SMOTE: Synthetic Minority Over-sampling Technique. *JAIR*. 2002. Vol. 16. P. 321–357.

7. Breiman L. Random Forests. *Machine Learning*. 2001. Vol. 45. P. 5–32.

8. Chen T., Guestrin C. XGBoost: A Scalable Tree Boosting System. In: *Proc. ACM SIGKDD*. 2016. P. 785–794.

9. Ke G. et al. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *NeurIPS*. 2017. Vol. 30.

10. Radford A., Metz L., Chintala S. Unsupervised Representation Learning with Deep Convolutional GANs. arXiv:1511.06434 [cs.LG], 2015.

11. Kingma D. P., Welling M. Auto-Encoding Variational Bayes. arXiv:1312.6114 [stat.ML], 2013.

12. Géron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. 2nd ed. O'Reilly, 2019.

13. Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press, 2016.

14. Chollet F. Deep Learning with Python. 2nd ed. Manning, 2021.

15. McKinney W. Python for Data Analysis. 2nd ed. O'Reilly, 2018.

16. Pedregosa F. et al. Scikit-learn: Machine Learning in Python. *JMLR*. 2011. Vol. 12. P. 2825–2830.

17. Python Software Foundation. Python Language Reference. URL: https://www.python.org/doc/ (date of access: 13.05.2025).

18. Dheeru D., Karra Taniskidou E. UCI Machine Learning Repository. URL: https://archive.ics.uci.edu/ml/index.php (date of access: 13.05.2025).

19. Chauhan A., Soni D. A Review on ML Approaches for Anomaly Detection. *Procedia Computer Science*. 2018. Vol. 132. P. 432–440. URL: https://doi.org/10.1016/j.procs.2018.05.200 (date of access: 13.05.2025).

20. Kim H. J., Kang H. J., Kim D. S. Anomaly Detection in Cybersecurity: A Deep Learning Approach. *Expert Systems with Applications*. 2020. Vol. 155. P. 113511.

21. Moustafa N., Slay J. B. A Survey of Intrusion Detection Systems Based on ML. *IJCSNS*. 2015. Vol. 15, no. 12. P. 12–21.

22. Salama S., Alhajj R., Han J. Mining Network Data for Anomaly Detection: A Survey. *Journal of Computing and Security*. 2020. Vol. 9. P. 12–43.

23. Bace R. Intrusion Detection. Macmillan Technical Publishing, 2000.

24. Фільченко О. О., Скляр А. В. Методи та засоби виявлення вторгнень. Х. : УІПА, 2019. 158 с.

25. Сидоренко В. А., Мартинюк О. П. Інформаційна безпека комп'ютерних систем. К. : Слово, 2018. 384 с.

26. Бережной І. М., Волошин А. С. Основи захисту інформації. Х. : ХНУРЕ, 2020. 260 с.

27. Френкель А. Введення в безпеку комп'ютерних мереж. М. : ДМК Пресс, 2017. 304 с.