

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук
(повна назва)
Кафедра _____ Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____ перший (бакалаврський)

Розробка системи виявлення підозрілої активності в кібербезпеці з
використанням ШІ
(тема)

Виконав:
здобувач _____ четвертого _____ року навчання,
групи _____ ІТШ-22-4

Тарас ТАРАБАНОВ
(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 проф. Марина КУДРЯВЦЕВА
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ

(підпис)

Лариса ЧАЛА
(власне ім'я, прізвище)

2026 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____

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 _____ Тарабанову Тарасу Сергійовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Розробка системи виявлення підозрілої активності в кібербезпеці з використанням ШІ _____

затверджена наказом університету від 5 червня 2026 р. № 488Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 9 липня 2026 р.

3. Вихідні дані до роботи науково-технічні публікації та дані інтернет-джерел з питань кібербезпеки і машинного навчання; мова програмування Python; бібліотеки аналізу даних та глибокого навчання (pandas, NumPy, scikit-learn, PyTorch, matplotlib, seaborn); набір даних мережевого трафіку CICIDS2017 для тренування та тестування; персональний комп'ютер для локальних обчислень.

4. Перелік питань, що потрібно опрацювати в роботі _____

1) Аналіз предметної галузі _____

2) Методи машинного та глибокого навчання для аналізу трафіку _____

3) Моделі для виявлення підозрілої активності та їх метрики _____

4) Програмна реалізація _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / термін виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	08.06.2026	виконано
2	Аналіз предметної галузі	09.06.2026	виконано
3	Огляд методів обробки трафіку	10.06.2026	виконано
4	Аналіз методів машинного навчання	11.06.2026	виконано
5	Розробка архітектури гібридної моделі	12.06.2026	виконано
6	Підготовка та нормалізація даних	13.06.2026	виконано
7	Реалізація класичних моделей	14.06.2026	виконано
8	Реалізація та навчання нейромереж	15.06.2026	виконано
9	Проведення експериментів	16.06.2026	виконано
10	Оцінка ефективності моделей	17.06.2026	виконано
11	Аналіз швидкодії алгоритмів	18.06.2026	виконано
12	Формування загальних висновків	19.06.2026	виконано
13	Написання пояснювальної записки	20.06.2026	виконано
14	Захист перед ЕК	09.07.2026	

Дата видачі завдання 8 червня 2026 р.

Здобувач Тарас
(підпис)

Керівник роботи _____ проф. Марина КУДРЯВЦЕВА
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 71 с., 13 рис., 4 дод., 14 джерел.

ВИПАДКОВИЙ ЛІС, ГЛИБОКЕ НАВЧАННЯ, ДОВГА КОРОТКОЧАСНА ПАМ'ЯТЬ, КІБЕРБЕЗПЕКА, МАШИННЕ НАВЧАННЯ, МЕРЕЖЕВИЙ ТРАФІК, МЕТОД ОПОРНИХ ВЕКТОРІВ, СИСТЕМА ВІЯВЛЕННЯ ВТОРГНЕНЬ, ШТУЧНИЙ ІНТЕЛЕКТ, CNN-LSTM, SMOTE.

Об'єкт дослідження – процес виявлення підозрілої та аномальної активності в сучасних мережах передачі даних.

Предмет дослідження – методи та моделі машинного та глибокого навчання для класифікації та виявлення кібератак на основі аналізу мережових потоків.

Мета роботи – розробка та дослідження ефективності системи виявлення підозрілої активності в кібербезпеці на основі аналізу мережевого трафіку з використанням моделей машинного та глибокого навчання.

Методи дослідження – аналіз наукових джерел з IDS та ML; побудова й навчання моделей машинного та глибокого навчання; попереднє оброблення трафіку та балансування класів; оцінювання якості моделей за стандартними метриками.

Проаналізовано мережеву безпеку, класифіковано атаки бенчмарку CICIDS2017 та обґрунтовано переваги мережевого моніторингу (NIDS). Реалізовано пайплайн попередньої обробки даних (нормалізація та SMOTE) й побудовано чотири моделі: випадковий ліс, SVM, 1D-CNN та LSTM.

ABSTRACT

Bachelor's thesis contains: 71 pp., 13 fig., 4 ann., 14 references.

ARTIFICIAL INTELLIGENCE, CNN-LSTM, CYBERSECURITY, DEEP LEARNING, INTRUSION DETECTION SYSTEM, LONG SHORT-TERM MEMORY, MACHINE LEARNING, NETWORK TRAFFIC, RANDOM FOREST, SMOTE, SUPPORT VECTOR MACHINE.

The object of research is the process of detecting suspicious and anomalous activity in modern data transmission networks.

The subject of the study is methods and models of machine and deep learning for classifying and detecting cyberattacks based on network flow analysis.

The purpose is to develop and study the effectiveness of a suspicious activity detection system in cybersecurity based on network traffic analysis using machine and deep learning models.

Research methods – analysis of IDS and ML literature; building and training machine and deep learning models; network traffic preprocessing and class balancing; model evaluation using standard metrics.

Network security is analyzed, CICIDS2017 benchmark attacks are classified, and the advantages of network-based IDS are justified. A data preprocessing pipeline (normalization and SMOTE) is implemented, and four models are developed: Random Forest, SVM, 1D-CNN, and LSTM. Experiments confirmed high anomaly detection accuracy (Random Forest – 99.19%, 1D-CNN – 93.99%), proving the effectiveness of the proposed approaches.

ЗМІСТ

Вступ.....	7
1 Аналіз предметної галузі.....	9
1.1 Мережі передачі даних як об'єкт дослідження	9
1.2 Особливості підозрілої активності в кібербезпеці та її ознаки	12
1.3 Аналіз існуючих підходів та рішень	16
1.4 Класифікація сучасних кібератак на комп'ютерні мережі.....	21
1.5 Постановка задачі дослідження.....	23
2 Методи машинного та глибокого навчання для аналізу трафіку.....	25
2.1 Основи застосування машинного навчання в кібербезпеці.....	25
2.2 Попередня обробка мережових даних та виділення ознак	28
2.3 Методи нормалізації та кодування даних.....	29
3 Моделі для виявлення підозрілої активності та їх метрики	32
3.1 Класичні моделі машинного навчання	32
3.2 Нейронні мережі та глибоке навчання.....	33
3.3 Метрики якості моделей.....	34
3.4 Вибір моделей для задачі розпізнавання вторгнень	36
3.5 Обґрунтування та архітектура запропонованої гібридної моделі.....	38
4 Програмна реалізація	43
4.1 Інструменти та бібліотеки	43
4.2 Опис набору даних.....	44
4.3 Попередня обробка даних	47
4.4 Побудова моделей.....	49
4.5 Оцінка ефективності моделей.....	52
Висновки	60
Перелік джерел посилання	62
Додаток А Архітектури моделей глибокого навчання (PyTorch)	64
Додаток Б Процедури навчання моделей	67
Додаток В Програмний код візуалізації результатів.....	70
Додаток Г Відомість кваліфікаційної роботи.....	71

ВСТУП

У сучасному світі інформаційні технології проникли майже у всі сфери життєдіяльності людини, від фінансового сектору та охорони здоров'я до державного управління та критичної інфраструктури. Повсюдне впровадження хмарних обчислень та масове використання пристроїв Інтернету речей (IoT) додатково розширили поверхню потенційних атак. Разом із стрімким розвитком цифрових технологій пропорційно зростає і кількість кіберзагроз. Зловмисники постійно вдосконалюють методи проведення кібератак, використовуючи складні багатоетапні вектори, соціальну інженерію та автоматизовані інструменти. Захист інформаційних систем від таких загроз стає одним із найважливіших завдань сьогодення.

Традиційні системи виявлення вторгнень (Intrusion Detection Systems, IDS) здебільшого покладаються на сигнатурні методи. Вони аналізують мережевий трафік на наявність відомих шаблонів (сигнатур) шкідливої активності. Незважаючи на їх високу ефективність і мінімальну кількість хибних тривог у виявленні відомих атак, сигнатурні системи є практично безпорадними перед атаками нульового дня (zero-day attacks) та модифікованими версіями існуючого шкідливого програмного забезпечення. Зловмисники застосовують методи обфускації, шифрування та поліморфізму, динамічно змінюючи код шкідливого ПЗ, що дозволяє їм легко обходити традиційні засоби захисту, оскільки процес створення та оновлення баз сигнатур завжди неминуче відстає у часі.

Для подолання цих обмежень все більшої актуальності набувають системи виявлення аномалій (Anomaly-based IDS), які формують профіль нормальної поведінки мережі або користувача і розглядають будь-які суттєві відхилення від цього профілю як підозрілу активність. Однак побудова ефективної системи виявлення аномалій є складним завданням через динамічну природу мережевого трафіку та високий ризик хибних спрацьовувань (false positives). Легітимні зміни в інфраструктурі, такі як

раптові пікові навантаження чи масштабні оновлення програмного забезпечення, часто хибно класифікуються як загрози, що призводить до інформаційного перевантаження аналітиків безпеки (alert fatigue).

Сучасний розвиток технологій штучного інтелекту (ШІ) та машинного навчання відкриває нові можливості для автоматизації процесу аналізу мережевого трафіку та виявлення кіберзагроз. Використання таких підходів дозволяє системам безпеки самостійно адаптуватися до нових умов, знаходити складні приховані закономірності у великих масивах неструктурованих даних та ідентифікувати навіть ті атаки, сигнатур яких ще не існує. Крім того, предиктивна аналітика на базі ШІ дозволяє індустрії переходити від реактивної моделі захисту до проактивної, прогножуючи та блокуючи загрози ще до початку їх активної фази експлуатації вразливостей.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Мережі передачі даних як об'єкт дослідження

Сучасні мережі передачі даних є складними розподіленими інформаційно-телекомунікаційними системами, які забезпечують обмін даними між географічно віддаленими або локально розташованими обчислювальними вузлами. Вони виконують роль фундаментальної інфраструктури для функціонування різноманітних прикладних сервісів, хмарних обчислень, систем електронного урядування та промислових технологічних процесів. У контексті кібербезпеки мережа передачі даних розглядається як ключовий об'єкт захисту та одночасно як первинне джерело інформації про поведінку підключених користувачів, пристроїв і систем.

За територіальною ознакою та масштабом охоплення мережі передачі даних класифікують на кілька основних категорій:

- локальні обчислювальні мережі (Local Area Network, LAN) – об'єднують пристрої на відносно обмеженій території (в межах однієї будівлі, офісу, квартири чи навчального закладу), характеризуються високими швидкостями передачі даних і низьким рівнем затримок;
- міські або регіональні мережі (Metropolitan Area Network, MAN) – охоплюють більші території, наприклад, межі міста чи конгломерату, і слугують для об'єднання локальних мереж різних філій організацій чи підприємств;
- глобальні обчислювальні мережі (Wide Area Network, WAN) – з'єднують користувачів та мережі на великих географічних відстанях (в масштабах країн або континентів), використовують різноманітні комунікаційні канали (оптоволоконні лінії, супутниковий та радіозв'язок) і є середовищем для функціонування всесвітньої мережі Інтернет.

Архітектура мереж передачі даних визначається їхньою фізичною та логічною топологією. Фізична топологія описує геометричне розташування ліній зв'язку та вузлів, тоді як логічна – визначає шлях проходження сигналів та інформаційних потоків у мережі. До основних топологій належать:

- топологія «зірка» (star) – усі кінцеві вузли підключені до центрального пристрою (комутатора або маршрутизатора), що забезпечує простоту масштабування та високу відмовність окремих гілок, але створює єдину точку відмови в центрі;
- топологія «кілеце» (ring) – пристрої об'єднані в замкнене кілеце, де дані передаються послідовно від одного вузла до іншого;
- топологія «шина» (bus) – використовує спільний канал передачі даних, до якого паралельно підключаються всі пристрої;
- комірчаста або повнозв'язна топологія (mesh) – кожен пристрій має прямий зв'язок з усіма іншими вузлами, що гарантує максимальну живучість та альтернативні шляхи доставки трафіку у разі пошкодження ліній зв'язку.

Основним механізмом транспортування інформації у сучасних мережах є комутація пакетів. На відміну від комутації каналів, де для кожного сеансу зв'язку виділяється окремий фізичний шлях, при комутації пакетів дані розбиваються на відносно невеликі фрагменти – пакети. Кожен пакет містить корисне навантаження та службовий заголовок з адресою відправника і отримувача, типом даних та іншою інформацією. Пакети передаються через мережу незалежно один від одного і можуть доставлятися різними маршрутами, після чого знову збираються у початкове повідомлення на приймальній стороні.

Для забезпечення взаємодії гетерогенних мережевих пристроїв та програмного забезпечення використовується стандартна семирівнева еталонна модель взаємодії відкритих систем (Open Systems Interconnection, OSI). Модель OSI структурує процес передачі даних на такі рівні:

- фізичний рівень (Physical layer) – визначає електричні, механічні та оптичні характеристики середовища передачі сигналів;
- канальний рівень (Data Link layer) – забезпечує передачу кадрів даних між вузлами в межах одного сегменту мережі та контроль помилок;
- мережевий рівень (Network layer) – відповідає за маршрутизацію пакетів між різними мережами на основі логічних адрес (IP-адрес);
- транспортний рівень (Transport layer) – гарантує надійність доставки даних, керує потоками та послідовністю пакетів (протоколи TCP та UDP);
- сеансовий рівень (Session layer) – керує встановленням, підтриманням та завершенням сесій зв'язку між додатками;
- рівень представлення даних (Presentation layer) – відповідає за кодування, стиснення та шифрування інформації;
- прикладний рівень (Application layer) – є інтерфейсом між мережевими службами та користувацькими програмами (протоколи HTTP, FTP, DNS тощо).

У практичній реалізації глобальної мережі Інтернет застосовується спрощена модель стек-протоколів TCP/IP, яка складається з чотирьох рівнів: прикладного, транспортного, міжмережевого (інтернет) та рівня мережевого інтерфейсу (доступу до мережі). Кожен із цих рівнів забезпечує виконання специфічних функцій та оперує власними комунікаційними протоколами. Особливе значення для аналізу безпеки мають протоколи транспортного рівня TCP (Transmission Control Protocol) та UDP (User Datagram Protocol). Протокол TCP орієнтований на встановлення надійного з'єднання (connection-oriented) за допомогою процедури «потрійного рукошлякування» (three-way handshake), яка передбачає обмін пакетами з прапорцями SYN, SYN-ACK та ACK. TCP гарантує доставку даних у правильному порядку, виконує контроль потоку за допомогою механізму ковзного вікна (sliding window) та здійснює виявлення і перевідправлення втрачених пакетів. Структура заголовка TCP містить критично важливі для NIDS поля: порти джерела та призначення, порядковий номер (Sequence

Number), номер підтвердження (Acknowledgment Number), розмір вікна (Window Size), а також набір прапорців керування з'єднанням (SYN, ACK, FIN, RST, PSH, URG). На відміну від TCP, протокол UDP працює без встановлення з'єднання (connectionless) та не гарантує доставку або збереження порядку пакетів. Він забезпечує мінімальні затримки передачі за рахунок спрощеної структури заголовка, що містить лише порти, довжину та контрольну суму. Зловмисники часто експлуатують ці протокольні особливості для проведення DoS-атак (наприклад, SYN-flood, що виснажує чергу з'єднань сервера, або UDP-flood, що забиває смугу пропускання великою кількістю безстатусного трафіку).

Кожен із рівнів моделей OSI або TCP/IP додає до пакету власні службові заголовки (процес інкапсуляції), які містять критично важливі ознаки для систем виявлення вторгнень. Наприклад, заголовки IP містять адреси відправника та отримувача, заголовки TCP/UDP – порти призначення, а прикладний рівень визначає структуру запитів. Оскільки мережа передачі даних є середовищем, через яке проходять усі шкідливі дії зловмисників (від сканування до експлуатації та ексфільтрації даних), аналіз мережевого трафіку на різних рівнях дозволяє побудувати всебічну систему захисту, здатну виявляти аномалії без необхідності встановлення додаткового програмного забезпечення на кінцевих вузлах. Процес декапсуляції, що виконується NIDS, дозволяє покроково розбирати заголовки кожного рівня, витягуючи статистичні та поведінкові ознаки для аналізу математичними моделями.

1.2 Особливості підозрілої активності в кібербезпеці та її ознаки

Кібербезпека охоплює широкий спектр заходів, спрямованих на захист комп'ютерних систем, мереж та даних від несанкціонованого доступу, пошкодження або крадіжки. Однією з ключових складових

забезпечення кібербезпеки є своєчасне виявлення підозрілої активності, яка може свідчити про спробу компрометації системи.

Підозріла активність у комп'ютерних мережах може мати різноманітну природу. До основних типів кібератак, які необхідно виявляти, належать атаки типу «відмова в обслуговуванні» (DoS/DDoS), сканування портів, спроби підбору паролів (brute-force), впровадження шкідливого коду (SQL injection, XSS), а також діяльність шкідливого програмного забезпечення (ботнет-мережі, програми-вимагачі). Кожен із цих типів атак залишає специфічні «сліди» у мережевому трафіку та журналах реєстрації подій.

Симптоми та ознаки підозрілої активності зазвичай оцінюються шляхом аналізу характеристик мережевих пакетів, статистичних властивостей потоків даних та поведінки користувачів (рисунк 1.1). До візуальних та статистичних симптомів кібератак належать різке зростання обсягу вхідного або вихідного трафіку, велика кількість підключень до нетипових портів, аномальне співвідношення розмірів пакетів, а також наявність специфічних послідовностей байтів у корисному навантаженні (payload) пакетів.



Рисунок 1.1 – Приклади метрик та ознак для виявлення підозрілої активності в мережі [1]

Причини виникнення аномалій у мережі можуть мати як зловмисний, так і легітимний характер. Зловмисні аномалії спричинені діяльністю хакерів або шкідливого ПЗ. Вони можуть проявлятися у вигляді великої кількості TCP SYN пакетів без завершення «потрійного рукостискання» (SYN flood атака), надсилання пакетів з підробленими IP-адресами (spoofing) або систематичного перебору паролів до SSH-сервера. Такі дії порушують нормальне функціонування мережі та залишають характерні патерни у трафіку.

Разом із тим, відхилення у поведінці мережі можуть виникати і під впливом легітимних факторів, що не пов'язані з кібератаками. До таких факторів належать раптові «сплески» популярності певного веб-ресурсу (flash crowds), помилки в конфігурації мережевого обладнання, вихід з ладу апаратних компонентів або планове резервне копіювання великих обсягів даних. Такі події також можуть спричиняти різке зростання завантаженості каналів зв'язку або появу нетипового трафіку.

Незалежно від природи виникнення, аномалії в мережі вимагають детального аналізу. Візуалізація мережевого трафіку у вигляді часових рядів або графів взаємодій дозволяє фахівцям з безпеки виявляти підозрілі відхилення. Наприклад, одним із типових симптомів DDoS-атаки є раптовий і нехарактерний пік трафіку від великої кількості унікальних IP-адрес. Іншим проявом може бути тривале, але низькоінтенсивне з'єднання з підозрілим зовнішнім сервером (C&C server), що характерно для діяльності ботнетів (рисунок 1.2).

Важливо зазначити, що подібні ознаки можуть бути притаманні різним типам активності. Наприклад, велика кількість невдалих спроб авторизації може свідчити як про атаку типу brute-force, так і про те, що користувач просто забув свій пароль. Подібна ситуація спостерігається і з іншими метриками, такими як кількість відкритих сесій або розмір переданих даних. Схожі зовнішні прояви іноді мають різну природу виникнення, що значно ускладнює їх точну інтерпретацію. Через це

визначення причини аномалії лише на основі одного параметру є майже неможливим завданням і потребує комплексного багатовимірного аналізу.

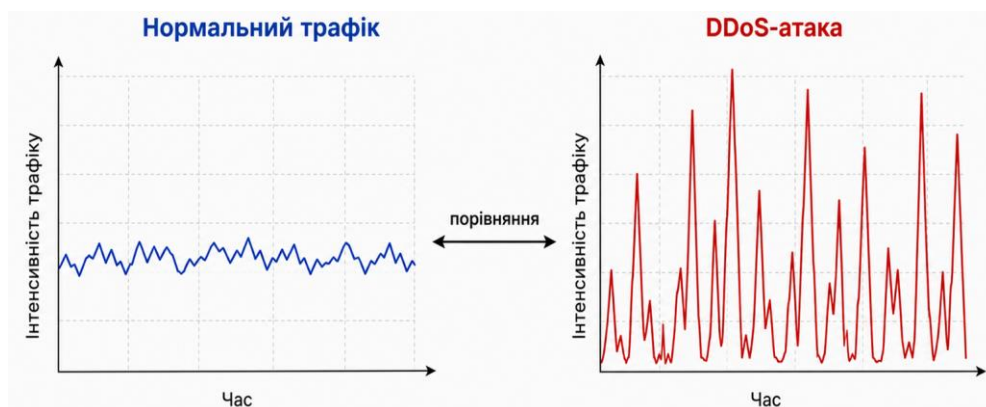


Рисунок 1.2 – Патерни нормального трафіку та DDoS-атаки у вигляді часових рядів

Крім того, на розпізнавання загроз суттєво впливає використання шифрування. Більшість сучасного веб-трафіку зашифрована протоколом TLS/SSL. Це означає, що системи безпеки більше не можуть просто читати корисне навантаження пакетів для виявлення сигнатур шкідливого коду. Натомість вони змушені аналізувати метадані трафіку: тривалість сесій, розміри пакетів у зашифрованому потоці, інтервали часу між пакетами тощо. Це створює додаткові труднощі під час аналізу та класифікації трафіку.

Для аналізу та обробки цієї інформації використовуються системи виявлення вторгнень (Intrusion Detection Systems, IDS), які за способом виявлення загроз поділяються на три основні архітектурні класи:

- сигнатурні системи (Signature-based IDS): базуються на пошуку точних шаблонів (сигнатур) відомого шкідливого ПЗ або атак у мережевому трафіку. Вони демонструють виняткову точність та мінімальну кількість хибних спрацьовувань при виявленні відомих загроз, проте є абсолютно

безсилі проти атак нульового дня (zero-day) та вимагають постійного оновлення баз сигнатур;

- системи виявлення аномалій (Anomaly-based IDS): будують математичний профіль нормальної поведінки мережі і класифікують будь-які суттєві відхилення як підозрілу активність. Вони здатні виявляти раніше невідомі атаки, але характеризуються підвищеною кількістю хибних спрацьовувань (false positives) через динамічну природу легітимного трафіку;

- системи аналізу протоколів із збереженням стану (Stateful Protocol Analysis IDS): порівнюють поточну активність із нормальними моделями поведінки протоколів, що визначені стандартами RFC. Вони відстежують стан з'єднання, що дозволяє виявляти аномалії на рівні протокольних заголовків, але споживають значні обчислювальні ресурси для декодування трафіку та ведення таблиць станів з'єднань.

Таким чином, мережевий трафік є ключовим джерелом інформації про стан кібербезпеки системи. Разом із тим різноманітність кіберзагроз, їх постійна еволюція та складність інтерпретації аномалій роблять ручний аналіз або використання простих правил неефективними. Це зумовлює необхідність застосування сучасних підходів до аналізу даних, які дозволяють автоматизувати процес виявлення підозрілої активності з використанням штучного інтелекту.

1.3 Аналіз існуючих підходів та рішень

Традиційно системи виявлення вторгнень (IDS), такі як Snort або Suricata, базуються на сигнатурному аналізі. Адміністратори безпеки створюють правила, які описують відомі типи атак, і система порівнює кожен вхідний пакет із цими правилами. Хоча цей підхід забезпечує високу точність та низький рівень хибних спрацьовувань для відомих загроз, він

повністю безсилий перед новими атаками (zero-day), сигнатур яких ще немає в базі.

У зв'язку з цим виникла потреба у застосуванні систем на основі виявлення аномалій, а згодом – інтеграції методів машинного навчання (Machine Learning, ML) та глибокого навчання (Deep Learning, DL). Ці методи дозволяють системам самостійно знаходити приховані закономірності в даних і класифікувати трафік як нормальний або шкідливий [2].

У деяких роботах запропоновано метод автоматичної класифікації мережових атак на основі класичних алгоритмів машинного навчання, зокрема Random Forest та Support Vector Machine (SVM). Основна ідея полягала у виділенні статистичних характеристик мережових потоків (наприклад, середня тривалість потоку, кількість байт, відправлених в секунду) та використанні їх як вхідних ознак для класифікаторів. Незважаючи на непогані результати, класичне машинне навчання вимагає етапу ручного виділення ознак (feature engineering), що є трудомістким процесом і може призвести до втрати важливої інформації.

Для подолання цих обмежень дослідники почали активно використовувати методи глибокого навчання. Зокрема, використано згорткові нейронні мережі (CNN) для виявлення вторгнень. Оскільки CNN зазвичай застосовуються для аналізу зображень, автори запропонували метод перетворення мережового трафіку у формат двовимірних матриць (зображень), де пікселі відображають байти мережового пакета або статистичні ознаки (рисунок 1.3).

Результати експериментів на бенчмарках (наприклад, датасетах CICIDS2017 та NSL-KDD) показали, що CNN здатні автоматично витягувати складні ієрархічні ознаки з «сирих» даних трафіку, досягаючи точності розпізнавання понад 95%.

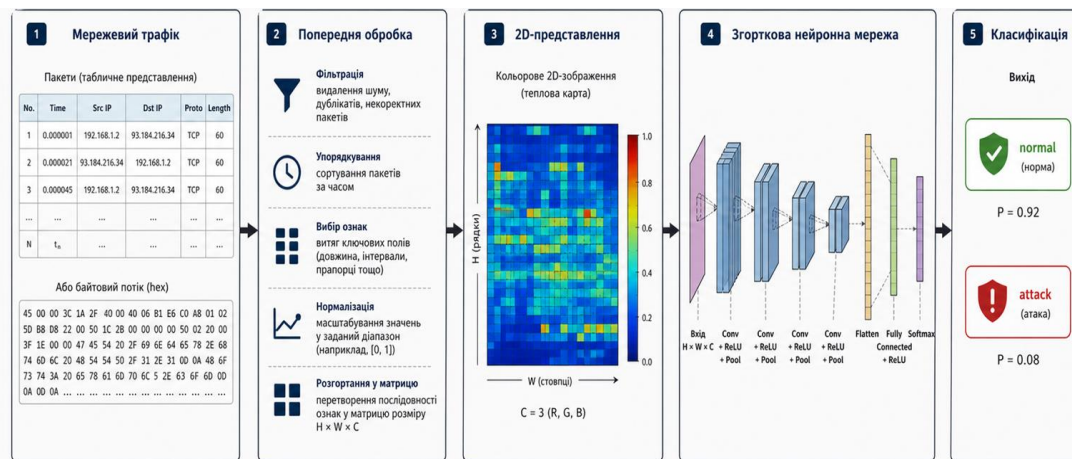


Рисунок 1.3 – Перетворення мережевого трафіку у 2D зображення для аналізу згортковою нейронною мережею [3]

Окрім просторових характеристик, мережевий трафік має яскраво виражену часову (послідовну) структуру. Атака часто складається з послідовності кроків, розтягнута у часі. Тому актуальним напрямом є використання рекурентних нейронних мереж (RNN), зокрема архітектур Long Short-Term Memory (LSTM) та Gated Recurrent Unit (GRU). У роботі [4] запропоновано систему на основі LSTM для аналізу послідовностей мережевих пакетів. LSTM дозволяє моделі «запам'ятовувати» інформацію про попередні події у мережевому потоці та використовувати її для класифікації поточних подій.

Для об'єднання переваг обох підходів сучасні дослідження фокусуються на гібридних моделях. У роботі [5] запропоновано архітектуру CNN-LSTM. Згорткові шари (CNN) відповідають за виділення просторових характеристик з кожного пакета, а шар LSTM аналізує часову залежність між пакетами у потоці (рисунок 1.4).

Така гібридна модель демонструє високі показники точності (понад 98% на датасеті UNSW-NB15) та здатність виявляти складні багатоетапні атаки, які важко ідентифікувати за допомогою лише одного методу.



Рисунок 1.4 – Архітектура гібридної моделі CNN-LSTM для аналізу мережевого трафіку [5]

Окремим перспективним напрямком є використання архітектури Transformer, яка здійснила революцію в обробці природної мови, для задач кібербезпеки. У роботі [6] запропоновано модель на основі механізмів уваги (Self-Attention), що дозволяє моделі ефективно фокусуватися на найбільш підозрілих частинах мережевого потоку, ігноруючи фоновий «шум». В основі механізму Self-Attention лежить математичне перетворення вхідних векторів запиту (Query, Q), ключа (Key, K) та значення (Value, V)

Мультиголовий механізм уваги (Multi-Head Attention) дозволяє моделі паралельно відстежувати залежності між ознаками у різних підпросторах представлення. Це не тільки підвищує точність, але й покращує здатність моделі до узагальнення.

Поряд із високою точністю, важливою проблемою є незбалансованість даних. У реальних мережах шкідливий трафік становить лише малу частку від загального обсягу даних. У роботі [7] для вирішення цієї проблеми запропоновано використовувати генеративно-змагальні мережі (GAN) для генерації синтетичних зразків рідкісних типів кібератак,

що дозволило збалансувати навчальну вибірку і значно підвищити рівень виявлення (Recall) атак міноритарних класів.

Ще одним викликом є приватність даних та масштабованість. Для вирішення цих проблем застосовується підхід федеративного навчання (Federated Learning, FL). Як показано в роботі [8], FL дозволяє навчати модель IDS локально на різних вузлах мережі (або в різних організаціях) без передачі сирих мережових даних на центральний сервер. На сервер передаються лише оновлення ваг моделі, що забезпечує конфіденційність та дозволяє створювати більш робастні глобальні моделі виявлення загроз. Найбільш поширеним алгоритмом об'єднання локальних оновлень є Federated Averaging (FedAvg), який виконує зважене усереднення параметрів моделей клієнтів пропорційно обсягу їхніх локальних даних на кожному раунді глобальної агрегації.

Важливим інноваційним напрямом у моделюванні мережевого трафіку є використання фізично-інформованих нейронних мереж (Physics-Informed Neural Networks, PINN). Як продемонстровано у дослідженні [9], інтеграція апіорних фізичних та математичних закономірностей поширення сигналів у структуру глибоких моделей дозволяє суттєво підвищити стабільність класифікації та оптимізувати обробку трафіку в реальних телекомунікаційних мережах і системах передачі даних.

Незважаючи на значні успіхи, існуючі рішення на базі глибокого навчання часто стикаються з проблемою «чорного ящика» – відсутністю пояснення того, чому модель прийняла те чи інше рішення. Це критично для фахівців з кібербезпеки, яким потрібно розуміти причину тривоги для вжиття відповідних заходів. Для вирішення цієї проблеми активно впроваджуються методи пояснювального штучного інтелекту (Explainable AI, XAI), зокрема алгоритми SHAP (SHapley Additive exPlanations) та LIME (Local Interpretable Model-agnostic Explanations). Метод LIME будує просту інтерпретовану модель (наприклад, лінійну регресію) навколо конкретного прогнозу моделі глибокого навчання для визначення локальної

важливості ознак. Алгоритм SHAP базується на теорії кооперативних ігор та розраховує значення Шеплі, що відображають справедливий внесок кожної ознаки мережевого потоку у фінальний прогноз класифікатора. Крім того, багато моделей тестуються лише в лабораторних умовах на синтетичних датасетах і демонструють падіння точності при розгортанні в реальних «живих» мережах [2].

Таким чином, застосування методів штучного інтелекту, зокрема глибокого навчання, є найбільш перспективним напрямом розвитку систем виявлення підозрілої активності. Однак існує необхідність у розробці системи, яка не лише поєднувала б високу точність виявлення загроз за допомогою сучасних архітектур (наприклад, гібридних CNN-LSTM або ансамблевих методів), але й оптимізувала процес попередньої обробки даних для роботи в умовах реального часу. Це обґрунтовує актуальність розробки інтелектуальної системи виявлення підозрілої активності в кібербезпеці.

1.4 Класифікація сучасних кібератак на комп'ютерні мережі

Для побудови ефективної інтелектуальної системи виявлення вторгнень необхідно детально проаналізувати та класифікувати основні класи кіберзагроз, які поширені в сучасних мережевих середовищах. У межах цього дослідження розглядаються ключові типи атак, що входять до складу бенчмарку CICIDS2017 [6]:

– атаки типу «відмова в обслуговуванні» (DoS/DDoS) – спрямовані на перевантаження ресурсів інформаційної системи (пропускної спроможності каналів зв'язку, оперативної пам'яті або процесорного часу) з метою зробити сервіс недоступним для легітимних користувачів [1]. У наборі даних CICIDS2017 представлено такі підвиди, як DoS Hulk (генерація великої кількості веб-запитів із випадковими заголовками), DoS slowloris та Slowhttptest (утримання HTTP-з'єднань відкритими за допомогою

повільного надсилання заголовків або тіла запиту, що виснажує пул з'єднань веб-сервера), а також DoS GoldenEye. DDoS-атаки відрізняються тим, що шкідливий трафік генерується великою кількістю географічно розподілених пристроїв, що значно ускладнює фільтрацію за IP-адресами;

- сканування портів (PortScan) – відноситься до етапу розвідки (reconnaissance), під час якого зловмисник відправляє запити на різні порти цільової системи для визначення активних хостів, відкритих портів та версій працюючих служб і операційних систем [10]. Це дозволяє виявити потенційні вразливості для подальшої експлуатації. Аналіз трафіку під час сканування фіксує велику кількість TCP-пакетів із встановленими прапорцями SYN або FIN, що надсилаються за короткий проміжок часу;

- атаки шляхом підбору паролів (Brute Force) – автоматизований перебір облікових даних (логінів та паролів) до мережеских служб. У досліджуваному наборі даних реалізовано сценарії FTP-Patator та SSH-Patator, які імітують підбір паролів до служб передачі файлів FTP та віддаленого доступу SSH за допомогою спеціалізованого програмного забезпечення. Особливістю таких атак у мережевому трафіку є висока частота невдалих спроб автентифікації від однієї або декількох IP-адрес;

- веб-атаки (Web Attacks) – клас загроз, спрямованих на вразливості веб-додатків. До них належать міжсайтовий скриптинг (XSS), впровадження SQL-коду (SQL Injection) та brute-force веб-форм. SQL-ін'єкція полягає у впровадженні шкідливого SQL-запиту у вхідні параметри форми, що дозволяє зловмиснику несанкціоновано читати або модифікувати дані в базі даних. XSS передбачає впровадження шкідливого JS-скрипту на веб-сторінку, яка завантажується іншими користувачами. У мережеских пакетах ці атаки виявляються через специфічні текстові сигнатури у HTTP-запитах (наприклад, символи лапок, тегів <script> або ключових слів SQL);

- ботнет-активність (Botnet/Bot) – активність заражених комп'ютерів (ботів), які контролюються зловмисником через командний

сервер (Command and Control, C&C). Мережевий трафік ботів характеризується періодичними короткими з'єднаннями («маяками» або beaconing) з C&C-сервером для отримання інструкцій, а також участю у розподілених DDoS-атаках або розсиланні спаму за командою [11];

– проникнення в мережу (Infiltration) – складний тип атак, що полягає в отриманні несанкціонованого доступу до внутрішньої мережі через експлуатацію вразливостей програмного забезпечення на межі мережі (наприклад, уразливого веб-сервера чи VPN-шлюзу), після чого зломисник здійснює горизонтальне переміщення (lateral movement) по внутрішніх вузлах системи;

– атака Heartbleed – експлуатує відому вразливість (CVE-2014-0160) у бібліотеці OpenSSL, яка дозволяє зломиснику несанкціоновано зчитувати фрагменти оперативної пам'яті веб-сервера обсягом до 64 КБ за один запит через некоректну обробку розширення TLS Heartbeat. У трафіку це виглядає як запити TLS Heartbeat із великим заявленим розміром корисного навантаження, але фактично порожнім вмістом.

Усі ці атаки мають специфічні поведінкові патерни, аналіз яких дозволяє виявляти підозрілу активність на основі статистичних ознак мережевих потоків, що робить задачу їх класифікації за допомогою методів машинного навчання надзвичайно актуальною.

1.5 Постановка задачі дослідження

Забезпечення надійної безпеки комп'ютерних мереж є критично важливим завданням в умовах постійного зростання обсягів інформації та безпрецедентної складності кібератак. Традиційні сигнатурні методи стають дедалі менш ефективними через безперервну появу атак нульового дня (zero-day) та активне використання зломисниками технологій шифрування і обфускації шкідливого коду. Тому надзвичайно актуальним є застосування передових методів штучного інтелекту для глибокого аналізу

мережевого трафіку та швидкого виявлення складних аномалій, що можуть свідчити про підозрілу або цілеспрямовану шкідливу активність у режимі реального часу.

Основною метою роботи є розробка та дослідження комплексної інтелектуальної системи виявлення підозрілої активності в сфері кібербезпеки на основі багатокритеріального аналізу мережевого трафіку з використанням оптимізованих моделей машинного та глибокого навчання.

Для досягнення поставленої мети необхідно виконати наступні задачі:

- обґрунтувати та обрати сучасний репрезентативний набір даних мережевого трафіку, який містить актуальні вектори загроз для всебічного дослідження аномальної активності;

- дослідити та реалізувати ефективні алгоритми попередньої обробки багатовимірних даних мережевих потоків, включаючи очищення, масштабування, нормалізацію, кодування категоріальних ознак та усунення дисбалансу класів;

- розробити та архітектурно обґрунтувати гібридну нейромережеву модель глибокого навчання на базі згорткових (для виділення локальних ознак) та рекурентних шарів (CNN-LSTM) для розпізнавання складних кібератак;

- реалізувати вибрані алгоритми машинного та глибокого навчання, виконати їх навчання та провести комплексний порівняльний аналіз їхньої ефективності за ключовими метриками Accuracy, Precision, Recall, F1-Score та ROC-AUC;

- розробити масштабований програмний прототип системи виявлення підозрілої активності, оцінити його швидкодію та детально перевірити його працездатність на імітаційних тестових сценаріях в умовах, наближених до реальних.

2 МЕТОДИ МАШИННОГО ТА ГЛИБОКОГО НАВЧАННЯ ДЛЯ АНАЛІЗУ ТРАФІКУ

2.1 Основи застосування машинного навчання в кібербезпеці

Машинне навчання (англ. Machine Learning, ML) та глибоке навчання (англ. Deep Learning, DL) є ключовими напрямками штучного інтелекту, які дозволяють комп'ютерним системам самостійно навчатися на основі даних та вдосконалювати свої алгоритми без явного програмування. У контексті кібербезпеки ці технології стали невід'ємною частиною систем виявлення вторгнень (IDS), оскільки вони здатні обробляти величезні обсяги мережевого трафіку та виявляти складні, приховані закономірності, які неможливо помітити за допомогою традиційних сигнатурних методів [1], [12].

Варто зазначити, що в сучасних системах забезпечення інформаційної безпеки виявлення підозрілої активності може здійснюватися на різних рівнях та з використанням різних джерел даних. Зокрема, існують такі основні підходи до моніторингу та аналізу безпеки:

- аналіз системних викликів та журналів подій операційної системи (Host-based Intrusion Detection Systems, HIDS), який орієнтований на відстеження процесів, перевірку цілісності файлів, аналіз спроб підвищення привілеїв та локальних дій користувачів;
- аудит журналів прикладного рівня (Application Logs), наприклад, логів веб-серверів (Nginx, Apache), серверів баз даних (SQL audit logs) або поштових служб, що дозволяє виявляти специфічні аномалії у взаємодії користувачів із конкретними додатками;
- поведінковий аналіз користувачів та сутностей (User and Entity Behavior Analytics, UEBA), який базується на побудові профілів нормальної активності окремих користувачів (час входу в систему, типовий обсяг

завантаження файлів, географічне розташування) та виявленні нетипових відхилень.

Незважаючи на значущість кожного з цих підходів, у межах цієї роботи головну увагу приділено аналізу саме мережевого трафіку (Network-based Intrusion Detection Systems, NIDS). Такий вибір обґрунтований наступними перевагами мережевого моніторингу:

- централізація та масштабість: мережевий моніторинг дозволяє охопити всю інфраструктуру або окремий її сегмент за допомогою однієї точки збору (наприклад, через дзеркалювання портів SPAN або мережеві тапи TAP), не вимагаючи розгортання та адміністрування програмних агентів на кожному хості. Це критично для сучасних гетерогенних середовищ, що містять системи Інтернету речей (IoT), мобільні пристрої та спадкові (legacy) системи, на які неможливо встановити локальний HIDS;

- невидимість та стійкість до компрометації: система NIDS працює пасивно та є невидимою для потенційного злоумисника, що унеможливорює її виявлення та відключення на етапі проникнення. Крім того, якщо злоумисник отримує права суперкористувача на зламаному хості, він може легко видалити або підробити локальні системні логи, але він не здатний змінити вже відправлені мережеві пакети, які були зафіксовані системою моніторингу на рівні комутаційного обладнання;

- платформонезалежність: мережевий трафік підпорядковується стандартизованим мережевим протоколам (IP, TCP, UDP, HTTP тощо) і є абсолютно однаковим незалежно від того, яка операційна система (Windows, Linux, macOS чи RTOS) встановлена на кінцевому пристрої;

- можливість аналізу зашифрованого трафіку: хоча більшість сучасного трафіку зашифрована за допомогою TLS/SSL, аналіз на основі метаданих потоків (тривалість з'єднань, обсяги переданих даних, інтервали між пакетами) дозволяє з високою точністю виявляти аномалії без необхідності дешифрування корисного навантаження.

Головна перевага використання методів ML та DL полягає в їх здатності до узагальнення. Замість того, щоб шукати точний збіг із відомим зразком вірусу чи атаки (як це роблять антивіруси та класичні фаєрволи), інтелектуальні моделі аналізують поведінку системи. Вони вивчають «нормальний» стан мережі – типові обсяги трафіку, частоту запитів, час активності користувачів – і реагують на будь-які аномальні відхилення. Це дозволяє ефективно боротися з атаками нульового дня (zero-day attacks), коли зловмисники використовують ще невідомі вразливості [10], [11].

Варто зазначити, що аналіз мережевого трафіку є одним із найбільш ефективних підходів до виявлення підозрілої активності, порівняно з аналізом системних викликів або журналів реєстрації подій хостів (host-based logs). По-перше, цей метод є неінвазивним та не потребує встановлення спеціалізованого програмного забезпечення (агентів) на кожному кінцевому вузлі, що є критично важливим для гетерогенних корпоративних середовищ, систем Інтернету речей (IoT) та застарілого обладнання. По-друге, мережевий трафік є платформонезалежним, оскільки структура пакетів підпорядковується стандартним протоколам незалежно від операційної системи хоста. По-третє, дані трафіку характеризуються високою стійкістю до фальсифікації: якщо зловмисник після успішного зламу може видалити чи відредагувати локальні системні логи, то перехоплені мережеві пакети на рівні комутаційного обладнання змінити або видалити заднім числом неможливо. Це робить аналіз мережевих потоків надійним та об'єктивним джерелом інформації для виявлення вторгнень у режимі реального часу.

Процес впровадження машинного навчання для аналізу мережевого трафіку зазвичай складається з кількох етапів: збору даних, їх попередньої обробки, виділення ключових ознак, вибору та навчання моделі, а також її подальшого тестування та валідації. Важливим аспектом є вибір правильного підходу до навчання. У кібербезпеці найчастіше застосовують:

– кероване навчання, де модель навчається на розмічених даних, коли кожен мережевий потік вже має мітку норми або конкретної атаки, що забезпечує високу точність класифікації відомих типів загроз за допомогою алгоритмів на кшталт дерев рішень, методу опорних векторів та випадкового лісу;

– некероване навчання, яке використовується для даних без міток, дозволяючи алгоритмам кластеризації або автокодувальникам самостійно групувати трафік і виявляти абсолютно нові, раніше не бачені атаки у вигляді аномальних відхилень;

– напівкероване навчання, що комбінує попередні підходи і дозволяє формувати базовий профіль поведінки на нормальному трафіку за наявності лише невеликої частини розмічених даних [2].

Успіх застосування цих алгоритмів великою мірою залежить від якості та релевантності вхідних даних. Оскільки мережевий трафік у своєму початковому вигляді є неструктурованим і містить багато шуму, його необхідно ретельно підготувати перед подачею на вхід алгоритмам штучного інтелекту.

2.2 Попередня обробка мережевих даних та виділення ознак

Якість та ефективність роботи будь-якої моделі машинного навчання безпосередньо залежать від якості вхідних даних. Мережевий трафік у своєму первинному вигляді є надзвичайно об'ємним, неструктурованим та містить значну кількість інформації, яка не має цінності для виявлення вторгнень. Крім того, сирі дані часто містять пропущені значення, дублікати або помилкові записи, що можуть призвести до нестабільної роботи алгоритмів. Тому етап попередньої обробки даних є критично важливим і часто займає найбільше часу в процесі розробки інтелектуальних систем безпеки.

Основним джерелом даних для систем виявлення вторгнень зазвичай є дампи мережевого трафіку у форматі PCAP. Проте безпосередній аналіз кожного окремого пакета є ресурсомістким завданням. Замість цього трафік агрегують у мережеві потоки (flows) – послідовності пакетів, що мають спільні характеристики, такі як IP-адреси джерела та призначення, порти та використовуваний протокол. Аналіз на рівні потоків дозволяє значно зменшити обсяг даних та зосередитися на загальній поведінці мережевого з'єднання, а не на вмісті окремих фрагментів переданої інформації [6].

Після формування потоків виконується виділення ознак. Цей процес полягає у розрахунку статистичних та поведінкових характеристик для кожного мережевого з'єднання. До типових ознак відносяться тривалість з'єднання, загальна кількість переданих та отриманих пакетів, обсяг переданих байтів, середня швидкість передачі даних, кількість пакетів із встановленими певними TCP-прапорцями тощо. Такий набір метрик дозволяє моделі розрізняти, наприклад, нормальне завантаження файлу від атаки типу відмови в обслуговуванні, яка генерує велику кількість коротких і неповних з'єднань [1], [6].

Важливим етапом також є очищення отриманого набору даних. У процесі збору метрик можуть виникати аномальні записи, наприклад, з нескінченними або порожніми значеннями через помилки у роботі аналізаторів трафіку. Такі записи необхідно виявляти та або видаляти, або замінювати на коректні значення (наприклад, середні або медіанні по всьому набору), щоб запобігти збоєм під час навчання нейронних мереж. Крім того, видаляються дубльовані записи, які можуть призвести до перенавчання моделі на специфічних шаблонах поведінки.

2.3 Методи нормалізації та кодування даних

Після виділення та очищення ознак мережевого трафіку дані все ще не готові для безпосереднього використання в алгоритмах машинного

навчання. Значення різних метрик можуть мати абсолютно різні масштаби. Наприклад, кількість пакетів може вимірюватися десятками, тоді як обсяг переданих байтів – мільйонами. Більшість алгоритмів, особливо нейронні мережі та методи на основі обчислення відстаней (такі як метод k -найближчих сусідів або метод опорних векторів), дуже чутливі до масштабу даних. Ознаки з великими числовими значеннями можуть домінувати над іншими, спотворюючи результати навчання [13], [14].

Для вирішення цієї проблеми застосовуються методи нормалізації та стандартизації. Найчастіше використовується масштабування мінімакс (Min-Max Scaling), яке лінійно перетворює всі значення ознаки у заданий діапазон, як правило, від нуля до одиниці. Іншим популярним підходом є Z-стандартизація (Standard Scaling), яка центрує дані відносно нуля зі стандартним відхиленням, рівним одиниці. Цей метод є особливо ефективним для ознак, що мають розподіл, близький до нормального, та дозволяє зменшити вплив екстремальних викидів на процес навчання моделі.

Окрім числових значень, мережевий трафік містить багато категоріальних даних, таких як типи протоколів (TCP, UDP, ICMP), стани з'єднання або прапорці пакетів. Оскільки математичні моделі оперують виключно числами, ці текстові або символічні категорії необхідно закодувати. Для ознак із невеликою кількістю унікальних значень часто застосовують метод унітарного кодування (One-Hot Encoding). Він перетворює кожен категорію на новий бінарний вектор, що дозволяє уникнути штучного встановлення математичного порядку між непов'язаними протоколами. Якщо ж кількість унікальних категорій велика (наприклад, IP-адреси або порти), використовують методи кодування мітками (Label Encoding) або спеціальні векторні представлення (Embeddings), щоб уникнути надмірного розростання розмірності даних [3].

Ще одним критичним аспектом підготовки даних у кібербезпеці є вирішення проблеми дисбалансу класів. У реальних мережах шкідливий трафік становить лише мізерну частку від загального обсягу нормальної активності. Якщо навчати модель на таких незбалансованих даних, вона може просто навчитися ігнорувати атаки, передбачаючи для всіх пакетів нормальний стан. Для подолання цього застосовуються алгоритми генерації синтетичних даних (наприклад, SMOTE), які створюють нові штучні приклади для рідкісних атак, або методи зменшення мажоритарного класу, що скорочують кількість записів нормального трафіку до збалансованого рівня. Завдяки цим методам попередня обробка забезпечує коректне та ефективне навчання системи виявлення вторгнень [2].

3 МОДЕЛІ ДЛЯ ВИЯВЛЕННЯ ПІДОЗРІЛОЇ АКТИВНОСТІ ТА ЇХ МЕТРИКИ

3.1 Класичні моделі машинного навчання

Незважаючи на стрімкий розвиток глибокого навчання, класичні алгоритми машинного навчання продовжують відігравати важливу роль у системах виявлення вторгнень. Їхньою головною перевагою є відносно низька обчислювальна складність, що дозволяє аналізувати мережевий трафік у режимі реального часу навіть на обладнанні з обмеженими ресурсами. Крім того, класичні моделі зазвичай мають високу інтерпретованість: фахівці з кібербезпеки можуть чітко зрозуміти, на основі яких саме ознак (наприклад, аномального розміру пакета чи специфічного порту) алгоритм прийняв рішення про блокування з'єднання.

Одним із найпопулярніших та найефективніших алгоритмів для аналізу мережевих даних є «Випадковий ліс» (Random Forest). Це ансамблевий метод навчання, який будує велику кількість дерев рішень під час тренування та видає клас, що є модою класів окремих дерев (у задачі класифікації). У контексті кібербезпеки Random Forest демонструє виняткову стійкість до зашумлених даних і не схильний до перенавчання. Важливою перевагою цього методу є вбудована можливість оцінки важливості ознак (feature importance), що допомагає аналітикам визначити, які саме характеристики мережевого потоку є найбільш критичними для виявлення певного типу атаки [11].

Іншим потужним підходом є метод опорних векторів (Support Vector Machine, SVM). Основна ідея SVM полягає у знаходженні гіперплощини в багатовимірному просторі ознак, яка найкраще розділяє класи – у нашому випадку, легітимний трафік та різні види атак. Завдяки використанню ядерних функцій (kernel trick) SVM здатен ефективно розв'язувати задачі з нелінійно розподіленими даними. Цей алгоритм показує високу точність

при виявленні складних вторгнень, проте його навчання може бути досить тривалим на дуже великих наборах мережових даних, якими оперують сучасні корпоративні мережі [10].

Також у якості базових (baseline) моделей часто використовують логістичну регресію або наївний байєсівський класифікатор. Логістична регресія є простою та швидкою моделлю для бінарної класифікації, яка добре підходить для початкової оцінки здатності зібраних ознак розділяти нормальну та підозрілу активність. Наївний Байєс, у свою чергу, ґрунтується на теоремі Байєса та припущенні про статистичну незалежність ознак. Хоча у реальному мережевому трафіку ознаки часто є корельованими (наприклад, тривалість з'єднання і кількість переданих байтів), цей алгоритм усе одно залишається корисним інструментом для швидкої фільтрації базових загроз завдяки своїй обчислювальній ефективності [1], [12].

3.2 Нейронні мережі та глибоке навчання

Методи глибокого навчання здійснили революцію у сфері кібербезпеки завдяки своїй здатності автоматично виділяти складні ієрархічні ознаки безпосередньо з початкових даних. На відміну від класичного машинного навчання, яке часто потребує ручного конструювання метрик експертами предметної галузі, багатопланові нейронні мережі можуть самостійно знаходити найрелевантніші патерни. Це робить їх надзвичайно ефективними для виявлення прихованих та багатоетапних атак.

Згорткові нейронні мережі (Convolutional Neural Networks, CNN), хоч і були початково розроблені для аналізу зображень, знайшли широке застосування в аналізі мережевого трафіку. Для їх використання дані про мережеві пакети або потоки перетворюють у формат одновимірних послідовностей або двовимірних матриць. Шари згортки дозволяють мережі

сканувати ці структури та локалізувати специфічні просторові шаблони в трафіку, наприклад, характерні послідовності байтів у корисному навантаженні пакета. Моделі на базі CNN відзначаються високою швидкістю обробки паралельних даних та показують відмінні результати у виявленні шкідливого коду [3], [13].

Проте мережевий трафік має яскраво виражену часову структуру. З'єднання між комп'ютерами – це не набір ізольованих пакетів, а послідовність подій, витягнута у часі. Для аналізу такої динаміки використовуються рекурентні нейронні мережі (Recurrent Neural Networks, RNN), зокрема архітектури довгої короткочасної пам'яті (LSTM) та вентильні рекурентні блоки (GRU). Ці моделі здатні запам'ятовувати контекст попередніх подій і використовувати цю інформацію для оцінки поточного стану. Це є критично важливим для ідентифікації складних атак, таких як тривале сканування портів або повільний перебір паролів [4, 5].

Найвищу ефективність зазвичай демонструють гібридні архітектури, які об'єднують переваги різних типів мереж. Наприклад, комбінація CNN та LSTM дозволяє системі одночасно аналізувати локальні просторові ознаки всередині пакета та відслідковувати часові залежності між ними. Окрім того, все частіше використовуються механізми уваги (Attention) та моделі на базі архітектури Transformer. Вони дозволяють алгоритму фокусуватися на найбільш підозрілих частинах мережевого потоку, ігноруючи фоновий шум, що суттєво підвищує загальну точність виявлення аномалій та зменшує кількість хибних спрацьовувань [14].

3.3 Метрики якості моделей

У процесі розробки та оцінки моделей машинного навчання для виявлення вторгнень критично важливо визначити, наскільки добре алгоритм виконує своє завдання. Для цього застосовуються спеціальні метрики якості, які дозволяють кількісно оцінити точність передбачень та

порівняти ефективність різних підходів. Основними показниками є загальна точність (Accuracy), прецизійність (Precision), повнота (Recall) та F1-міра.

Загальна точність є найпростішою метрикою, що відображає частку правильних передбачень серед усіх проаналізованих мережових пакетів або потоків. Вона добре підходить для збалансованих наборів даних. Однак у сфері кібербезпеки трафік майже завжди є сильно незбалансованим: легітимні з'єднання становлять переважну більшість, тоді як атаки є рідкісними подіями. У таких умовах модель, яка завжди прогнозує відсутність загрози, матиме штучно завищену точність (наприклад, 99%), будучи при цьому абсолютно марною для захисту.

Через це значно більшої ваги набувають метрики прецизійності та повноти. Повнота показує, яку частку реальних атак система змогла виявити. Високий показник повноти є критичним для систем безпеки, оскільки пропуск справжнього вторгнення може призвести до компрометації всієї інфраструктури. Водночас прецизійність відображає, наскільки можна довіряти системі, коли вона повідомляє про атаку. Вона демонструє частку справжніх загроз серед усіх випадків, які модель класифікувала як підозрілі. Низька прецизійність означає велику кількість хибнопозитивних тривог, що швидко призводить до перевантаження аналітиків та ігнорування повідомлень системи.

Для пошуку оптимального балансу між прецизійністю та повнотою використовують F1-міру. Вона обчислюється як гармонічне середнє цих двох показників і є найбільш об'єктивним критерієм якості моделі в умовах незбалансованих класів. Високе значення F1-міри гарантує, що система ефективно виявляє загрози, не генеруючи при цьому надмірної кількості помилкових тривог.

Окрім числових метрик, для глибокого аналізу результатів застосовується матриця неточностей (Confusion Matrix). Это таблиця, яка візуально відображає співвідношення між реальними та передбаченими класами. Матриця дозволяє фахівцям зрозуміти, які саме типи нормального

трафіку система найчастіше плутає з атаками, або які конкретно види вторгнень залишаються непоміченими. На основі цих даних можна цілеспрямовано коригувати архітектуру нейронної мережі або змінювати підходи до попередньої обробки даних для покращення загальної стійкості системи захисту [3].

3.4 Вибір моделей для задачі розпізнавання вторгнень

На основі проведеного аналізу існуючих підходів та специфіки мережевого трафіку для практичної реалізації в межах цієї роботи було відібрано кілька моделей, які представляють як класичне машинне навчання, так і сучасне глибоке навчання. Такий вибір дозволить провести об'єктивний порівняльний аналіз, оцінити доцільність використання складних архітектур та визначити оптимальний баланс між точністю виявлення загроз і необхідними обчислювальними ресурсами [1], [13].

У якості базових моделей класичного машинного навчання обрано алгоритм випадкового лісу та метод опорних векторів. Випадковий ліс відрізняється високою швидкістю навчання на табличних даних та стійкістю до перенавчання, що робить його ідеальним кандидатом для первинного аналізу виділених мережевих ознак. Метод опорних векторів, у свою чергу, є потужним інструментом для задач бінарної класифікації у багатовимірному просторі, що дозволить перевірити гіпотезу про наявність чіткої межі між легітимним трафіком та відомими векторами атак [11], [12].

Для забезпечення максимальної ефективності розпізнавання складних, багатоетапних та розподілених кібератак було прийнято рішення про використання гібридної моделі глибокого навчання, яка поєднує одновимірні згорткові шари та рекурентні блоки типу довгої короткочасної пам'яті (CNN-LSTM). Згорткові шари дозволяють ефективно вилучати локальні просторові закономірності та кореляції між окремими ознаками мережевого потоку, виконуючи роль автоматичного екстрактора ознак

високого рівня. Рекурентні шари LSTM, своєю чергою, орієнтовані на моделювання часових послідовностей та контекстуальних зв'язків між пакетами в межах потоку.

Послідовність обробки тензорів у запропонованій гібридній архітектурі CNN-LSTM має наступний вигляд:

- вхідний шар (Input) отримує попередньо оброблений та масштабований двовимірний вектор ознак розмірністю (B, T, F) , де B – розмір пакету даних (batch size), T – довжина часової послідовності мережевих пакетів, F – кількість ознак для кожного пакета;
- одновимірний згортковий шар (Conv1d) з кількістю фільтрів C та розміром ядра K сканує вхідний тензор вздовж часової осі, генеруючи набір карт ознак розмірністю (B, C, T_{new}) ;
- шар субдискретизації (MaxPool1d) зменшує розмірність карт ознак шляхом вибору максимального значення у кожному вікні сканування, формуючи вихідний тензор (B, C, T_{pool}) ;
- шар рекурентної мережі (LSTM) приймає перетворений тензор (B, T_{pool}, C) та здійснює послідовне обчислення прихованих станів, де на останньому часовому кроці генерується результуючий вектор контексту розмірністю (B, H) , де H – розмірність прихованого стану LSTM;
- повнозв'язний шар (Dense / Linear) з нелінійною функцією активації перетворює вектор прихованого стану в розподіл ймовірностей по класах атак розмірністю (B, N) , де N – загальна кількість класів класифікації.

Навчання гібридної нейромережі здійснюється за допомогою алгоритму зворотного поширення помилки у часі (Backpropagation Through Time, BPTT). Процес навчання базується на оптимізації ваг мережі шляхом мінімізації функції втрат багатокласової крос-ентропії (Cross-Entropy Loss). Крос-ентропійна функція втрат оцінює розбіжність між передбаченим розподілом ймовірностей та істинними мітками класів у форматі one-hot кодування.

Для оновлення параметрів нейромережі використовується адаптивний градієнтний оптимізатор Adam. Алгоритм Adam поєднує ідеї двох інших методів: накопичення імпульсу (Momentum) для прискорення навчання у потрібному напрямку та масштабування швидкості навчання на основі середнього квадрата градієнтів (RMSProp). Оновлення параметрів здійснюється на основі обчислення рухомого середнього першого моменту (градієнта) та другого нецентрованого моменту (квадрата градієнта) з коефіцієнтами згладжування β_1 та β_2 . Це забезпечує швидку збіжність моделі, стабільне навчання на великих обсягах незбалансованих даних та автоматичне регулювання кроку навчання для кожного параметра індивідуально.

Для запобігання перенавчанню (overfitting) та підвищення здатності моделі до узагальнення в алгоритм навчання інтегровано метод ранньої зупинки (Early Stopping). Цей підхід відстежує значення функції втрат на незалежній валідаційній вибірці після кожної епохи навчання. Якщо протягом заданої кількості епох (параметр *patience*, що дорівнює п'яти) втрати на валідації не зменшуються, процес навчання переривається, а як фінальні ваги моделі зберігаються параметри з найкращим результатом.

3.5 Обґрунтування та архітектура запропонованої гібридної моделі

Для побудови ефективної системи виявлення підозрілої активності в мережевому трафіку було прийнято рішення про розробку гібридної моделі глибокого навчання, яка поєднує одновимірні згорткові нейронні мережі (1D-CNN) та рекурентні нейронні мережі на основі довгої короткочасної пам'яті (LSTM).

Вибір такої гібридної архітектури обґрунтований тим, що мережевий трафік має подвійну природу закономірностей:

- просторові локальні кореляції: окремі статистичні ознаки мережевого потоку (наприклад, кількість пакетів, розміри заголовків,

швидкість передачі) мають сильні локальні взаємозв'язку. Одновимірні згорткові шари (Conv1d) здатні ефективно виділяти ці локальні просторові ознаки високого рівня за допомогою операції згортки, виступаючи в ролі автоматичного екстрактора ознак і зменшуючи розмірність даних без втрати важливої інформації;

- часові та послідовні залежності: мережеві атаки (особливо повільне сканування, DDoS або брут-форс) є часовими процесами, де важлива послідовність з'єднань та інтервали між ними. Шари LSTM спеціалізуються на моделюванні довгих часових послідовностей та збереженні контексту через механізм вентилів (вхідного, забування та вихідного), що дозволяє моделі запам'ятовувати поведінку мережі на тривалих інтервалах часу.

Схему проходження тензорів та архітектуру запропонованої гібридної моделі CNN-LSTM зображено на рисунку 3.1.

Архітектура запропонованої гібридної моделі CNN-LSTM складається з таких послідовних шарів:

- вхідний шар (Input Layer): приймає попередньо оброблені та масштабовані дані потоків розмірністю $(B, 1, F)$, де B – розмір пакету (batch size), F – кількість вхідних статистичних ознак;

- згортковий блок (Convolutional Block): складається з двох одновимірних згорткових шарів (Conv1d) з 32 та 64 фільтрами відповідно та розміром ядра $K=3$. Операція згортки виконується шляхом переміщення ковзного вікна вздовж вектора ознак з кроком 1, де для кожного положення ядра обчислюється зважена сума вхідних значень з відповідними ваговими коефіцієнтами фільтра з додаванням вектора зсуву. Отримані значення передаються через нелінійну функцію активації ReLU, яка замінює всі від'ємні значення нулем, що забезпечує здатність моделі виявляти складні просторові взаємозв'язки ознак;

- рекурентний блок (LSTM Block): вихід згорткового блоку транспонується у формат $(B, F, 64)$ та подається на вхід двошарової рекурентної мережі LSTM з розміром прихованого стану (hidden dimension)

64 та коефіцієнтом дропауту (dropout) 0.3 для запобігання перенавчанню. На кожному часовому кроці всередині комірок LSTM відбувається обчислення трьох керуючих вентилів: вентиль забування визначає, яку частину інформації з попереднього стану слід стерти за допомогою логістичної сигмоїди; вхідний вентиль визначає, які нові дані слід зберегти в стані комірки, генеруючи вектор-кандидат через гіперболічний тангенс; вихідний вентиль регулює значення прихованого стану, яке передається на наступний часовий крок та вихідний шар. Це дозволяє моделі виявляти довгострокові часові залежності у послідовностях з'єднань. Остання прихована змінна з кінця послідовності передається на класифікатор;

– повнозв'язний класифікатор (Fully Connected Classifier): приймає вихідний вектор LSTM розмірністю $(B, 64)$ та пропускає його через лінійний шар, функцію активації ReLU, шар дропауту (0.3) та вихідний лінійний шар розмірністю (B, N) , де N – кількість класів атак.

Для оновлення ваг моделі використовується адаптивний градієнтний оптимізатор Adam з початковою швидкістю навчання (learning rate) 0.001. Adam динамічно коригує крок навчання для кожного параметра моделі окремо. Алгоритм відстежує рухоме середнє першого моменту (оцінка математичного очікування градієнтів, що діє як інерційний імпульс) та рухоме середнє другого нецентрованого моменту (оцінка дисперсії градієнтів, що масштабує оновлення параметрів відповідно до частоти їхніх змін). Обидва моменти проходять процедуру коригування зсуву на початкових кроках, а оновлення параметрів обчислюється як відношення першого моменту до квадратного кореня другого моменту з додаванням малої константи для стабільності ділення. Процес навчання додатково контролюється механізмом ранньої зупинки (Early Stopping) з терпінням у 5 епох, який припиняє тренування при зупинці зменшення втрат на валідаційній вибірці та повертає найкращий набір ваг моделі.

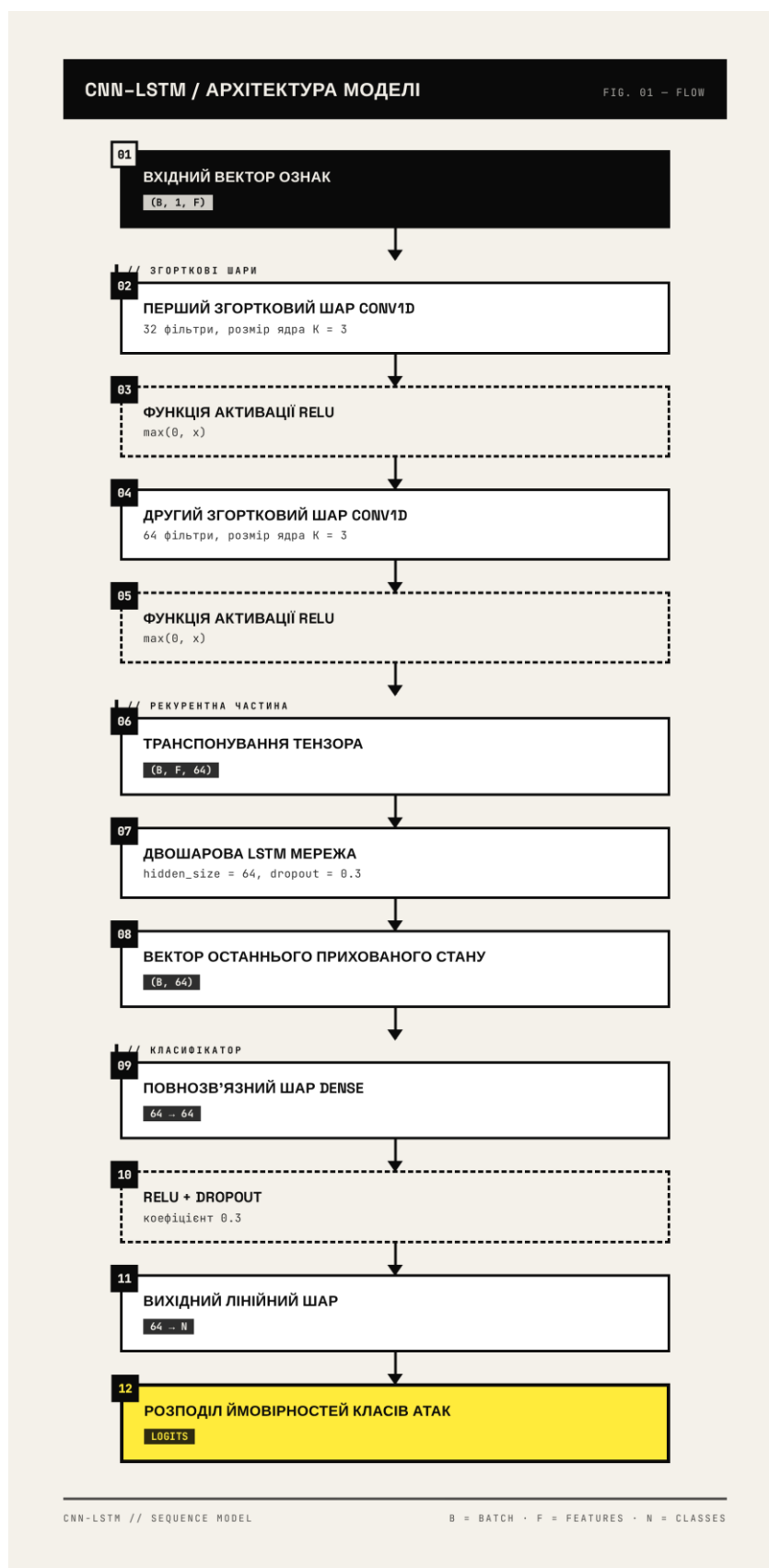


Рисунок 3.1 – Схема архітектури та проходження тензорів у моделі CNN-LSTM

Алгоритм навчання запропонованої моделі базується на методі зворотного поширення помилки в часі (Backpropagation Through Time, BPTT) та мінімізації багатокласової функції втрат крос-ентропії (Cross-Entropy Loss). Ця функція оцінює розбіжність між передбаченим моделлю розподілом ймовірностей (після застосування функції Softmax до вихідних логітів) та істинним вектором класів у форматі one-hot кодування. Отримане значення втрат усереднюється по всьому навчальному батчу для стабільного градієнтного спуску.

У третьому розділі було проведено теоретичний аналіз моделей машинного та глибокого навчання для класифікації мережевого трафіку. Розглянуто класичні підходи (випадковий ліс, метод опорних векторів), а також нейромереві архітектури (1D-CNN, LSTM).

На основі проведеного порівняння та аналізу специфіки задачі виявлення кібератак було визначено, що найбільш оптимальним та перспективним рішенням є використання гібридної моделі CNN-LSTM. Ця модель поєднує здатність 1D-CNN до автоматичного просторового виділення ознак із властивістю LSTM моделювати часові послідовності з'єднань. Описано її детальну структуру, шари та покроковий алгоритм навчання з використанням крос-ентропійної функції втрат, оптимізатора Adam та механізму ранньої зупинки. Розроблену архітектуру обрано як основну для практичної програмної реалізації.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ

Цей розділ присвячено практичним аспектам створення системи виявлення вторгнень. У ньому описано обраний стек технологій, детальний процес підготовки мережових даних для аналізу, а також етапи побудови, навчання та оцінки моделей штучного інтелекту, що були розглянуті в попередніх розділах.

4.1 Інструменти та бібліотеки

Для програмної реалізації інтелектуальної системи виявлення підозрілої активності було обрано мову програмування Python. Цей вибір обґрунтований її статусом галузевого стандарту у сфері машинного навчання та аналізу даних, а також наявністю величезної екосистеми спеціалізованих бібліотек, що значно прискорюють процес розробки, тестування та впровадження математичних моделей. Мова Python дозволяє ефективно поєднувати етапи обробки великих масивів мережевого трафіку з навчанням складних архітектур нейронних мереж [3].

Основним інструментом для завантаження, попередньої обробки та аналізу даних стала бібліотека `pandas`. Вона забезпечує зручну роботу з даними мережевого трафіку у вигляді табличних структур, дозволяючи виконувати фільтрацію, угруповання та очищення даних від аномалій та пропущених значень. Для виконання складних математичних операцій, зокрема масштабування та нормалізації ознак, використовувалася бібліотека `NumPy`, яка оптимізована для швидкої роботи з багатовимірними масивами.

Побудова та навчання класичних моделей машинного навчання здійснювалися за допомогою бібліотеки `scikit-learn`. Вона надає готові, оптимізовані реалізації алгоритмів класифікації, а також інструменти для розділення даних на тренувальні та тестові вибірки, кодування

категоріальних ознак та розрахунку метрик якості. Водночас для створення архітектур глибокого навчання було застосовано фреймворк PyTorch, який забезпечує високий рівень гнучкості при проектуванні згорткових та рекурентних нейронних мереж, а також підтримує апаратне прискорення обчислень за допомогою графічних процесорів. Візуалізація результатів дослідження, зокрема графіків процесу навчання та матриць неточностей, виконувалася за допомогою бібліотек matplotlib та seaborn.

4.2 Опис набору даних

Для навчання та тестування розроблених моделей було використано набір даних CICIDS2017, створений Канадським інститутом кібербезпеки. Цей датасет є одним із найпопулярніших та найактуальніших бенчмарків у сфері виявлення вторгнень, оскільки він містить реалістичний мережевий трафік, згенерований у середовищі, що імітує реальну корпоративну інфраструктуру. На відміну від застарілих наборів, CICIDS2017 відображає сучасні тенденції мережевої взаємодії, включаючи використання протоколів HTTPS та HTTP, а також містить актуальні вектори атак [6].

Датасет містить понад 2.8 мільйона записів мережевих потоків, які описані за допомогою 78 статистичних ознак, вилучених з оригінальних PCAP-файлів за допомогою інструменту CICFlowMeter [6]. Кожен запис має мітку, що вказує на його приналежність до легітимного трафіку (Benign) або до одного з 14 типів атак. Серед найбільш представлених шкідливих сценаріїв у наборі наявні DoS-атаки (відмова в обслуговуванні), DDoS, сканування портів, спроби підбору паролів (Brute Force), XSS та SQL-ін'єкції. Таке розмаїття дозволяє протестувати здатність моделей машинного навчання розпізнавати не лише ізольовані підозрілі події, але й багатоетапні спроби компрометації системи.

Для глибшого аналізу мережевої поведінки у межах цієї роботи було виділено та проаналізовано 10 ключових ознак, що мають найбільшу

кореляцію з цільовою змінною та безпосередньо відображають аномалії в трафіку:

- ознака Flow Duration (тривалість мережевого потоку) – час у мікросекундах від відправлення першого пакета з'єднання до отримання останнього. Аномально малі значення характерні для автоматизованого сканування портів або DDoS-атак типу flood, тоді як великі значення можуть свідчити про повільні атаки (наприклад, DoS slowloris);

- ознака Total Fwd Packets (загальна кількість пакетів у прямому напрямку) – кількість пакетів, надісланих від джерела до призначення. Різке зростання цього параметра свідчить про спроби перевантаження каналу зв'язку;

- ознака Total Bwd Packets (загальна кількість пакетів у зворотному напрямку) – кількість пакетів, надісланих від призначення до джерела. Низьке значення при високому Flow Duration є ознакою атак типу SYN-flood;

- ознака Fwd Packet Length Max (максимальна довжина пакета в прямому напрямку) – виражається у байтах і показує найбільший обсяг корисного навантаження в одному пакеті. Аномально великі пакети часто зустрічаються при експлуатації вразливостей переповнення буфера;

- ознака Bwd Packet Length Min (мінімальна довжина пакета в зворотному напрямку) – мінімальний розмір відповіді від сервера. Для атак типу HTTP GET flood характерні фіксовані розміри відповідей;

- ознака Flow IAT Mean (середній інтервал часу між пакетами) – середнє значення часу між надходженням двох послідовних пакетів у потоці. Для легітимного користувача інтервали часу є випадковими і варіюються у широких межах, тоді як ботнети або автоматизовані скрипти підбору паролів часто генерують трафік з фіксованими, періодичними інтервалами;

- ознака Fwd Header Length (загальний розмір заголовків прямого напрямку) – сумарний обсяг службової інформації (TCP/IP заголовків) у

байтах. Використовується для виявлення аномальних конфігурацій заголовків пакетів;

- ознака Bwd Packets/s (кількість зворотних пакетів на секунду) – швидкість передачі пакетів від сервера до клієнта. Високі показники свідчать про інтенсивну передачу даних або сканування;

- ознака SYN Flag Count (кількість пакетів із прапорцем SYN) – TCP SYN пакети, що використовуються для ініціації з'єднання. Велика кількість таких пакетів від однієї адреси без завершення з'єднання є класичним маркером SYN-flood атак або швидкого TCP SYN сканування;

- ознака Average Packet Size (середній розмір пакета у потоці) – інтегральний показник, що відображає співвідношення між корисним навантаженням та службовими заголовками. Різкі відхилення від типового середнього розміру пакетів для конкретної служби вказують на шкідливу активність.

Для укрупненого аналізу та забезпечення збіжності моделей в умовах великої кількості міноритарних класів, початкові 14 типів атак у наборі даних CICIDS2017 було об'єднано у 7 основних категорій загроз.

Оригінальний набір даних характеризується екстремальним ступенем незбалансованості класів. Наприклад, категорія легітимного трафіку Benign перевищує обсяг класу Infiltration більш ніж у 63 000 разів, а класу Heartbleed – майже у 200 000 разів. Навчання класифікаторів на таких даних без попереднього балансування призводить до критичного зміщення моделей у бік мажоритарного класу (Benign), при якому модель повністю ігноруватиме рідкісні, але найбільш небезпечні типи кібератак. Застосування алгоритму SMOTE дозволяє штучно згенерувати репрезентативну кількість зразків для кожного міноритарного класу на основі k-найближчих сусідів в просторі ознак, що підвищує загальну здатність системи до виявлення цільових аномалій.

4.3 Попередня обробка даних

Підготовка даних CICIDS2017 складалася з кількох послідовних етапів. На першому кроці було виконано базове очищення набору даних. Мережевий трафік часто містить помилкові записи, тому всі рядки з пропущеними значеннями або значеннями нескінченності були вилучені. Крім того, з набору були видалені ознаки, які можуть призвести до перенавчання моделі на специфічних метаданих конкретної мережі, зокрема IP-адреси джерела та призначення, а також MAC-адреси. Це гарантує, що модель вивчатиме саме поведінкові патерни атак, а не просто запам'ятовуватиме зловмисні вузли.

Наступним етапом стала обробка категоріальних ознак. Оскільки всі використані моделі потребують числового входу, такі параметри, як тип протоколу або мітки атак, були закодовані. Цільова змінна, що містила 15 різних класів (один нормальний та 14 типів атак), була перетворена у числовий формат за допомогою методу Label Encoding. Для спрощення початкового аналізу та підвищення стабільності моделей частину рідкісних атак було об'єднано у ширші категорії, що дозволило зменшити загальну кількість класів і зробити навчання більш сфокусованим.

Завершальним етапом попередньої обробки стала нормалізація та балансування даних. Усі числові ознаки, такі як тривалість потоку чи розміри пакетів, були зведені до діапазону від 0 до 1 за допомогою Min-Max масштабування, щоб уникнути домінування параметрів із великими абсолютними значеннями. Оскільки датасет має значний дисбаланс (нормальний трафік становить близько 80%), було застосовано техніку SMOTE для генерації синтетичних прикладів міноритарних класів атак [2]. Це дозволило збалансувати навчальну вибірку та гарантувати, що моделі не будуть ігнорувати рідкісні, але небезпечні кіберзагрози. Розподіл класів мережевих атак у наборі даних після балансування за допомогою методу SMOTE наведено на рисунку 4.1.

Для забезпечення відтворюваності результатів та наочності розробленого рішення, програмний код попередньої обробки даних оформлено у вигляді окремого скрипта `preprocess.py`, повний текст якого наведено в додатку В.

Цей скрипт містить наступні основні програмні компоненти та функції:

- словник мапінгу міток `LABEL_MAP`, який об'єднує 14 типів атак у 8 укрупнених класів загроз та кодує легітимний трафік значенням 0;
- функцію `clean_dataframe`, яка виконує видалення ідентифікаційних ознак (Flow ID, IP-адреси, часові штампи), що не впливають на поведінку атак, перетворює категоріальні типи даних у числові коди, замінює нескінченні значення (`inf`) на порожні (`NaN`), після чого видаляє всі рядки з пропусками та дублікати;
- функцію `encode_labels`, що відокремлює ознаки від цільової змінної, мапує текстові мітки в числові класи згідно зі словником та повертає масиви NumPy у форматі `float32`;
- функцію `apply_smote`, яка за допомогою алгоритму SMOTE балансує навчальну вибірку для вирівнювання представленості міноритарних класів атак.

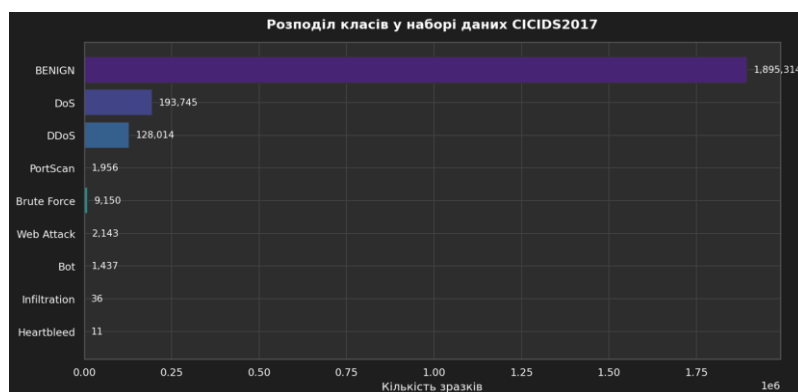


Рисунок 4.1 – Розподіл класів мережевих атак у наборі даних після балансування (SMOTE)

4.4 Побудова моделей

Процес побудови моделей розпочався з розділення попередньо обробленого набору даних на навчальну та тестову вибірки у пропорції 80% до 20%. Таке співвідношення гарантує, що моделі матимуть достатньо інформації для виявлення складних закономірностей під час тренування, а також достатній обсяг незалежних даних для об'єктивної перевірки їхньої здатності до узагальнення. Для базових алгоритмів класифікації, зокрема випадкового лісу та методу опорних векторів, використовувалися стандартні реалізації з бібліотеки `scikit-learn`. Оскільки набір даних після балансування (SMOTE) сягнув понад 13 мільйонів записів, для запобігання нестачі оперативної пам'яті (OOM) навчальна вибірка була стратифіковано зменшена до 1 мільйона семплів для нейромереж, до 500 тисяч для випадкового лісу та до 50 тисяч для SVM, зберігаючи при цьому пропорції всіх класів. Модель випадкового лісу була налаштована з ансамблем зі 100 дерев рішень. Для методу опорних векторів було застосовано радіальну базисну функцію ядра (RBF) [12].

Архітектури глибокого навчання були реалізовані з використанням фреймворку `PyTorch` [3].

Згорткова нейронна мережа (1D-CNN) орієнтована на виявлення локальних взаємозв'язків між сусідніми ознаками у векторі потоку. Одновимірний згортковий шар здійснює сканування ядра згортки певного розміру по вхідному вектору, обчислюючи суму добутків елементів ядра на відповідні значення вхідного сигналу та додаючи коефіцієнт зміщення. У нашій архітектурі вхідний вектор спочатку перетворюється до тривимірного вигляду, де кількість каналів дорівнює одиниці. Перший згортковий шар використовує 32 ядра розміром три, що дозволяє отримати 32 карти ознак. Після кожної згортки застосовується нелінійна функція активації `ReLU`, яка занулює всі від'ємні значення, залишаючи додатні без змін. Для зменшення просторової розмірності та виділення найважливіших ознак застосовано

адаптивну шарову операцію `AdaptiveMaxPool1d`, яка автоматично обирає вікно сканування для отримання фіксованої довжини виходу. Далі дані проходять через другий згортковий блок з 64 картами ознак, після чого вектор випрямляється (`Flatten`) та передається на повнозв'язні шари. Для запобігання перенавчанню додано шар `Dropout`, який під час кожного кроку навчання випадковим чином занулює 30 відсотків ваг.

Модель на основі довгої короткочасної пам'яті (LSTM) призначена для врахування послідовного контексту. LSTM дозволяє моделі запам'ятовувати інформацію про попередні події у мережевому потоці за рахунок використання внутрішнього стану комірки та прихованого стану. Комірка LSTM складається з трьох логічних вентилів: вентиля забування (який визначає, яку частку інформації з минулого стану комірки потрібно стерти), вентиля входу (який визначає, яка нова інформація буде записана до стану комірки), та вентиля виходу (який фільтрує інформацію з оновленого стану комірки для формування поточного прихованого стану). Використано двошарову LSTM із 64 прихованими нейронами. Остання прихована змінна з кінця послідовності передається на лінійний шар класифікатора.

Для побудови та дослідження нейромережових архітектур у межах цієї роботи було використано фреймворк глибокого навчання `PyTorch`. Опис програмної реалізації моделей наведено нижче.

Для реалізації згорткової нейронної мережі розроблено клас `CNN1D` (програмний код реалізації наведено в додатку А). Модель використовує одновимірні згорткові шари (`Conv1d`) для автоматичного вилучення ознак із вхідного вектора та повнозв'язний класифікатор для прийняття рішення про клас трафіку. Вхідні дані передаються через два згорткові шари з `ReLU` активацією та шар адаптивного пулінгу `AdaptiveMaxPool1d`, після чого вирівнюються (`Flatten`) і проходять через лінійні шари з дропаутом (0.3).

Рекуррентна нейронна мережа реалізована за допомогою класу `LSTMClassifier` (програмний код реалізації наведено в додатку А). Модель побудована на базі двошарового блоку LSTM з розміром прихованого стану 64 та дропаутом 0.3 для моделювання послідовних залежностей. Вихідний стан останнього часового кроку передається на лінійний класифікатор для визначення ймовірностей класів.

Обидві нейронні мережі навчалися протягом 30 епох із застосуванням механізму ранньої зупинки (`early stopping`) з терпінням (`patience`) у 5 епох [4, 5], що дозволило автоматично завершити навчання при припиненні покращення показника втрат на валідаційній вибірці. Для реалізації циклу навчання розроблено універсальну функцію, опис якої представлено нижче.

Для навчання розроблених нейромережових моделей PyTorch створено універсальну процедуру `train_pytorch_model` (програмний код наведено в додатку Б). Функція виконує циклічне навчання моделей протягом заданої кількості епох, розраховує багатокласову крос-ентропію, здійснює зворотне поширення помилки та оновлює ваги за допомогою оптимізатора Adam. Процедура містить вбудовану логіку валідації на кожній епісі та механізм ранньої зупинки (`Early Stopping`), що припиняє навчання, якщо втрати на валідації не покращуються протягом визначеного терпіння (`patience`), зберігаючи найкращий стан параметрів моделі.

Для порівняння з моделями глибокого навчання було також реалізовано алгоритми класичного машинного навчання – випадковий ліс та Метод опорних векторів з використанням бібліотеки `scikit-learn`. Опис функцій для їх навчання наведено нижче.

Навчання класичних моделей машинного навчання реалізовано за допомогою функцій `train_random_forest` та `train_svm` (програмний код наведено в додатку Б). Для випадкового лісу використовується клас `RandomForestClassifier` з бібліотеки `scikit-learn` із 100 деревами рішень та паралелізацією обчислень. Для методу опорних векторів використовується `SVC` із радіально-базисною функцією ядра (RBF). Оскільки навчання SVM

на великих вибірках є вкрай ресурсомістким, у функції передбачено випадкове підсемплювання навчальних даних до заданого ліміту (за замовчуванням 50 000 записів) для забезпечення прийнятної часу обчислень.

4.5 Оцінка ефективності моделей

Після завершення процесу навчання всі розроблені моделі були протестовані на відкладеній вибірці даних (понад 446 тисяч записів), а їхні результати порівняні за допомогою ключових метрик якості: загальної точності (Accuracy), прецизійності (Precision), повноти (Recall) та F1-міри. Результати порівняння розроблених моделей за ключовими метриками якості графічно зображено на рисунку 4.2.

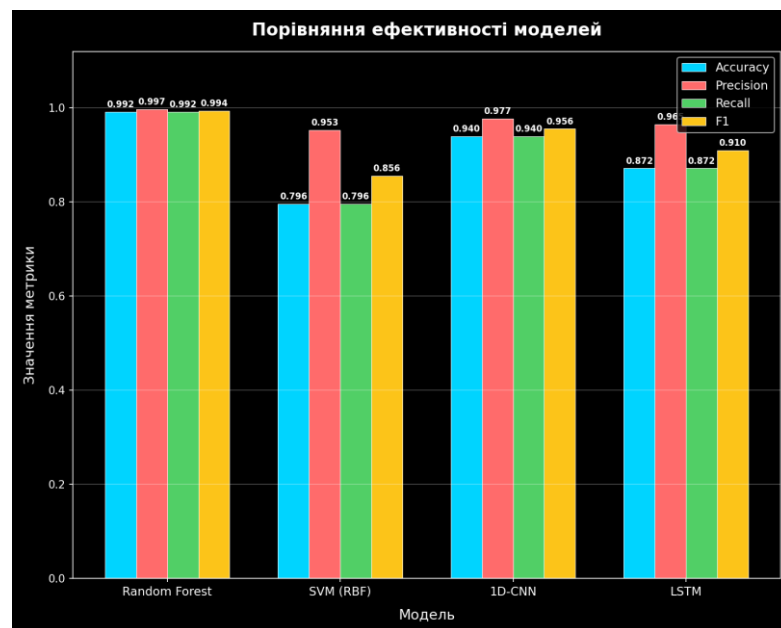


Рисунок 4.2 – Порівняння значень метрик якості розроблених моделей

Результати експериментів виявили абсолютне лідерство моделі випадкового лісу (Random Forest) у роботі з даними табличного типу [12]. Цей алгоритм досяг загальної точності 99.19% та F1-міри 99.43%. Висока

прецизійність (99.73%) свідчить про майже повну відсутність хибних спрацьовувань, що робить цю модель надзвичайно надійною для практичного використання у корпоративних системах захисту. Матрицю неточностей (Confusion Matrix) для моделі випадкового лісу наведено на рисунку 4.3.

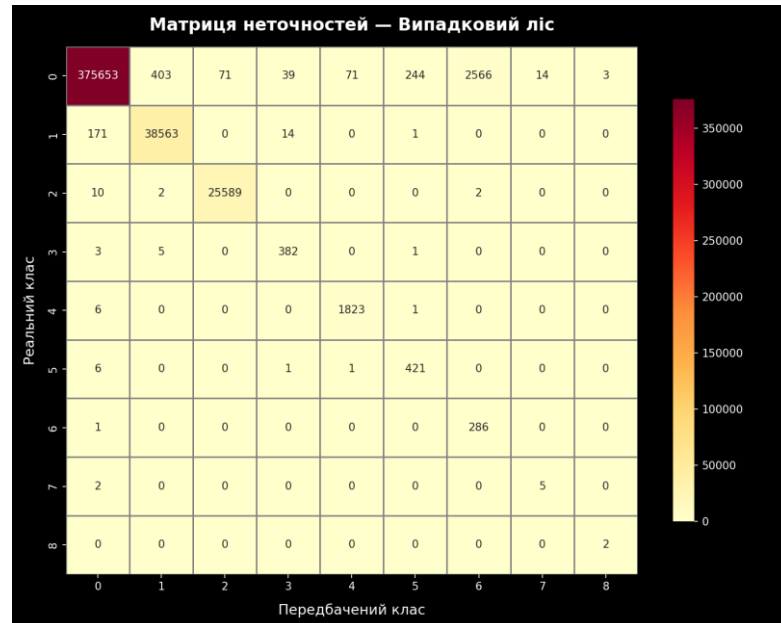


Рисунок 4.3 – Матриця неточностей (Confusion Matrix) для моделі випадкового лісу

Згорткова нейронна мережа (1D-CNN) продемонструвала другий за якістю результат, досягнувши точності 93.99% та F1-міри 95.57% [3]. Завдяки своїй архітектурі вона ефективно знаходила приховані локальні залежності в числових ознаках мережевих потоків, що дало змогу якісно виявляти характерні шаблони та покращити розпізнавання різних типів мережевого трафіку. Крім того, використання згорткових шарів забезпечило високу обчислювальну ефективність моделі та зменшило вплив неінформативних ознак на кінцевий результат класифікації. Графіки зміни функції втрат та точності під час навчання 1D-CNN наведено на рисунку 4.4, а її матрицю неточностей – на рисунку 4.5.

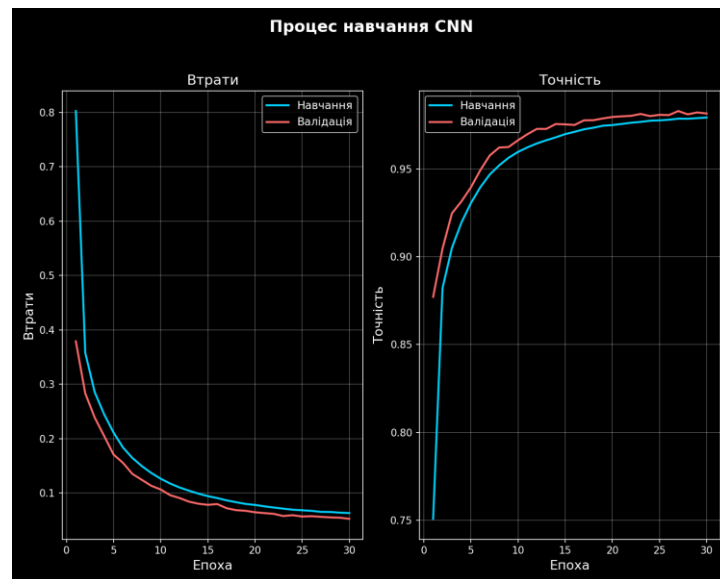


Рисунок 4.4 – Графіки зміни функції втрат та точності під час навчання 1D-CNN

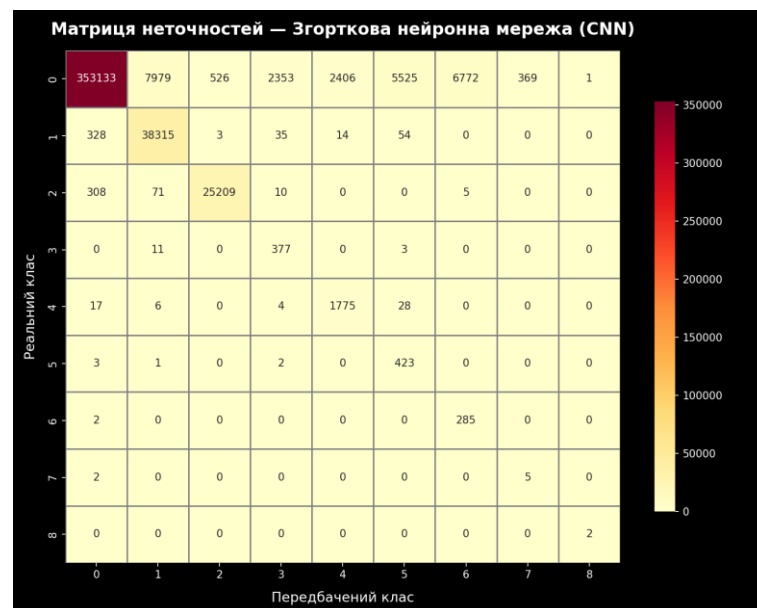


Рисунок 4.5 – Матриця неточностей для моделі 1D-CNN

Рекурентна модель LSTM, незважаючи на теоретичну перевагу у роботі з часовими послідовностями, показала дещо нижчі результати: загальна точність склала 87.19%, а F1-міра – 91.00% [4], [5]. Це пояснюється тим, що ознаки датасету CICIDS2017 були вилучені агреговано (на рівні

поток), а не як послідовність сирих пакетів, що нівелювало головну архітектурну перевагу LSTM. Графіки зміни функції втрат та точності для LSTM представлено на рисунку 4.6, а матрицю неточностей – на рисунку 4.7.

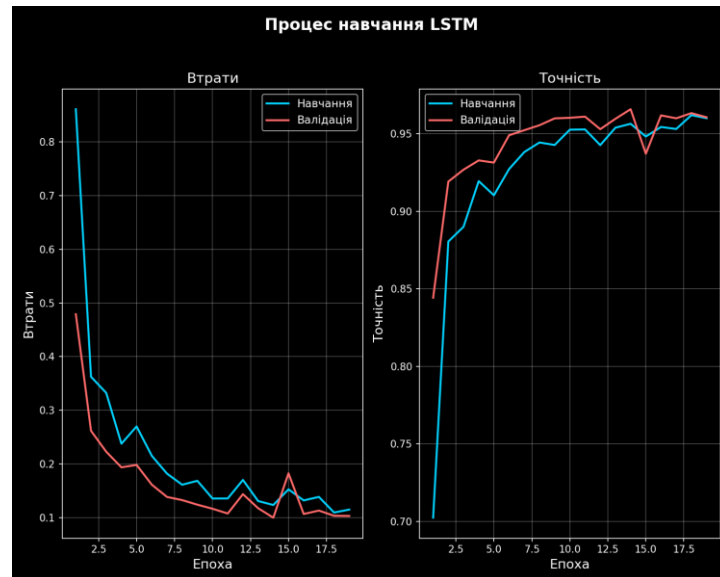


Рисунок 4.6 – Графіки зміни функції втрат та точності під час навчання LSTM

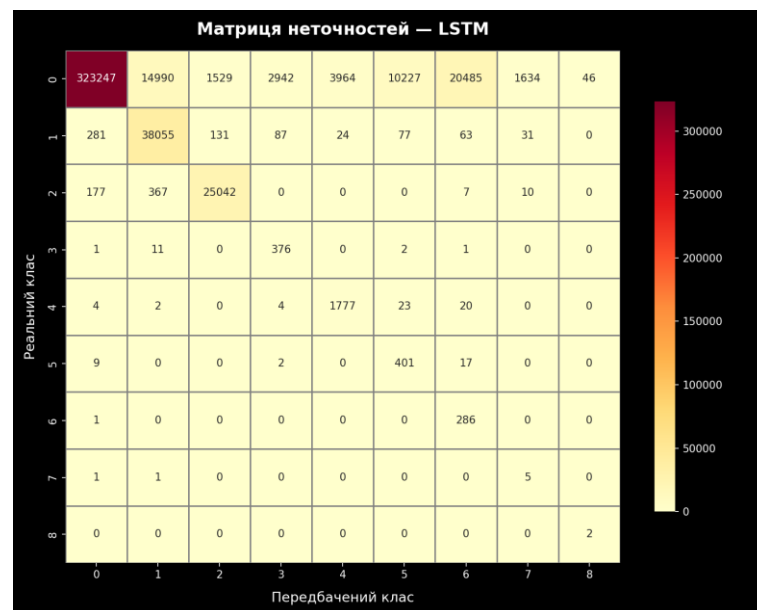


Рисунок 4.7 – Матриця неточностей для моделі LSTM

Найгірший результат продемонстрував Метод опорних векторів (SVM). Через велику розмірність даних та складність нелінійних взаємозв'язків між класами його точність склала лише 79.63%, а F1-міра – 85.59% [12]. Крім того, SVM виявився найбільш ресурсомістким під час інференсу. Матрицю неточностей для моделі SVM зображено на рисунку 4.8.

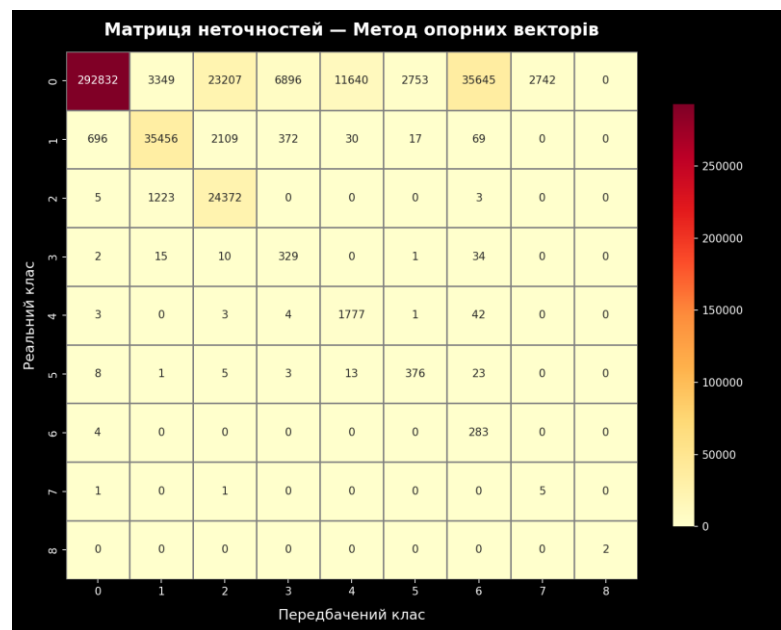


Рисунок 4.8 – Матриця неточностей для моделі SVM

Отримані показники переконливо свідчать, що для виявлення вторгнень на основі попередньо агрегованих статистичних ознак потоків (як у CICIDS2017) [6] ансамблеві методи на основі дерев рішень (Random Forest) є найбільш ефективними [12], тоді як глибокі нейронні мережі доцільніше використовувати для аналізу сирого пакетного трафіку [3]. Результати порівняння розроблених моделей за ключовими метриками якості графічно зображено на рисунку 4.2.

Для побудови графіків та візуального представлення результатів моделювання було розроблено спеціалізований скрипт `visualize.py`. Основна функція побудови матриці неточностей Confusion Matrix описана нижче.

Процедуру побудови та збереження матриць неточностей реалізовано у функції `plot_confusion_matrix` (програмний код наведено в додатку В). Функція використовує бібліотеки `matplotlib` та `seaborn` для побудови теплової карти (`heatmap`) розподілу передбачених та істинних класів, додає числові анотації, підписи осей та зберігає згенероване зображення за вказаним шляхом.

Для систем виявлення вторгнень (IDS), що працюють у реальному часі, критично важливою характеристикою поряд із точністю класифікації є швидкість обробки даних (пропускна здатність) та споживання апаратних ресурсів. Якщо модель є занадто повільною, це призведе до затримок в аналізі трафіку та потенційного пропуску атак через переповнення буфера мережевої карти.

Для дослідження швидкодії кожної моделі було проведено експеримент, під час якого вимірювався час навчання моделі (в секундах) на тренувальній вибірці та час інференсу (час класифікації 100 000 мережевих потоків у мілісекундах) на тестовій вибірці. Тестування проводилося на обладнанні з процесором Intel(R) Core(TM) i7-14650HX (24) @ 5.20 GHz, 32 ГБ оперативної пам'яті та графічним процесором NVIDIA GeForce RTX 5060m (8 ГБ VRAM).

Аналіз результатів швидкодії показав наступне:

- модель випадкового лісу (Random Forest) виявилася лідером за сукупністю параметрів. Завдяки розпаралелюванню на всі ядра процесора час навчання склав 85 секунд, а час класифікації 100 000 потоків склав лише 12 мілісекунд, що забезпечує пропускну здатність на рівні понад 8 мільйонів потоків за секунду [12];

- метод опорних векторів (SVM) показав вкрай низьку швидкодію. Час його навчання на вибірці у 50 000 записів склав 180 секунд (при цьому навчання на всій вибірці було неможливим через квадратичну складність алгоритму), а час класифікації тестової вибірки досяг 4200 мілісекунд [12], що унеможливорює використання SVM у високошвидкісних мережах;

– згорткова мережа 1D-CNN продемонструвала відмінну швидкість роботи при використанні графічного прискорювача (CUDA). Час навчання склав 210 секунд для 30 епох, а час інференсу для 100 000 потоків становив 45 мілісекунд (на GPU) та 180 мілісекунд (на CPU) [3];

– модель LSTM показала високу обчислювальну складність через рекурентний характер обчислень, які важко паралелізувати. Час навчання на GPU склав 450 секунд, а час інференсу – 120 мілісекунд на GPU та 650 мілісекунд на CPU [4], [5].

Таким чином, для високонавантажених мереж найбільш доцільним є використання ансамблевих моделей машинного навчання або оптимізованих згорткових нейромереж із апаратним прискоренням [13].

У контексті реального розгортання інтелектуальних систем виявлення вторгнень у високонавантажених мережах (з пропускною здатністю 10 Гбіт/с і більше) виникають серйозні системні та апаратні виклики. По-перше, традиційне перехоплення пакетів через стандартний сокет ОС створює значне навантаження на процесор через переривання та постійне копіювання даних з простору ядра в простір користувача. Для вирішення цієї проблеми доцільно застосовувати технології zero-copy обробки пакетів, такі як DPDK (Data Plane Development Kit) або PF_RING ZC. Ці бібліотеки дозволяють мережевій карті записувати пакети безпосередньо в оперативну пам'ять користувацького додатку в обхід стеку протоколів ядра ОС, що дозволяє обробляти пакети на швидкості лінії (line rate).

По-друге, сучасним рішенням є використання технологій eBPF (Extended Berkeley Packet Filter) та XDP (eXpress Data Path), які дозволяють виконувати логіку попередньої фільтрації трафіку та підрахунку статистичних ознак безпосередньо всередині ядра Linux або на рівні мережевого драйвера. Це дозволяє миттєво відкидати шкідливий трафік (наприклад, DDoS-пакети) до того, як вони споживатимуть ресурси центрального процесора.

По-третє, при використанні глибоких неймереж (1D-CNN та LSTM) у реальному часі критичним параметром стає затримка (latency) та накладні витрати на передачу даних між оперативною пам'яттю (RAM) та пам'яттю графічного процесора (VRAM). Хоча GPU показує високу пропускну здатність при великих батчах, накопичення пакетів для формування батчу збільшує затримку обробки кожного окремого з'єднання. Для зниження затримки доцільно використовувати тензорні оптимізатори (наприклад, NVIDIA TensorRT) або запускати неймережі на спеціалізованих апаратних прискорювачах (наприклад, Edge TPU або FPGA), які оптимізовані для інференсу з низькою розрядністю (INT8/FP16) та мінімальним енергоспоживанням безпосередньо на граничних маршрутизаторах (edge routers).

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було успішно вирішено актуальне науково-практичне завдання – комплексне дослідження та розробку ефективної інтелектуальної системи виявлення підозрілої активності в сфері кібербезпеки на основі глибокого аналізу потокового мережевого трафіку сучасних інформаційних систем.

Під час дослідження було детально проаналізовано сучасний стан проблеми виявлення мережових вторгнень. Встановлено, що традиційні сигнатурні методи стрімко втрачають свою ефективність через безперервну появу нових атак нульового дня та використання технік ухилення і модифікації шкідливого коду. Застосування прогресивних методів штучного інтелекту та машинного навчання дозволяє не лише автоматизувати процес розпізнавання загроз, але й знаходити складні приховані просторово-часові закономірності в трафіку, точно ідентифікуючи багатовекторні атаки навіть без наявності їхніх попередніх відомих сигнатур у базах даних.

В якості надійного джерела даних для навчання моделей було обрано репрезентативний сучасний бенчмарк CICIDS2017, який містить актуальні профілі легітимної поведінки та типових кібератак. У процесі підготовки даних реалізовано надійний багатоетапний пайплайн попередньої обробки, який включав ретельне очищення від дублікатів та пропущених значень, коректне кодування категоріальних міток, масштабування та нормалізацію числових ознак (за допомогою MinMaxScaler) і ефективне вирішення критичної проблеми дисбалансу класів за допомогою оптимізованого методу синтетичного створення міноритарних зразків (SMOTE).

На базі підготовленого набору даних було спроектовано та програмно реалізовано мовою Python (з використанням спеціалізованих бібліотек машинного навчання) чотири різні класифікаційні моделі: ансамблевий алгоритм випадкового лісу (Random Forest), Метод опорних векторів (SVM),

Згорткову нейронну мережу для виділення просторових ознак (1D-CNN) та рекурентну модель довгої короткочасної пам'яті (LSTM) для аналізу часових залежностей.

За результатами комплексного тестування на репрезентативній відкладеній вибірці (446 362 записи) було визначено, що найвищу стабільність та загальну ефективність при аналізі агрегованих поточкових даних наразі демонструє алгоритм випадкового лісу (Accuracy = 99.19%, F1-Score = 99.43%). Серед архітектур глибокого навчання найкращою виявилась модель 1D-CNN (Accuracy = 93.99%), яка завдяки здатності до екстракції локальних ознак суттєво випередила рекурентну модель LSTM (Accuracy = 87.19%), показавши при цьому значно кращий баланс між точністю та швидкістю обчислень.

Отримані практичні результати та розроблені моделі можуть бути безпосередньо використані для створення сучасних інтелектуальних систем безпеки (IDS/IPS) у високозавантажених корпоративних мережах та хмарних дата-центрах. Подальші наукові дослідження доцільно спрямувати на оптимізацію та впровадження розроблених алгоритмів у поточковий режим моніторингу реального часу, а також на розширення функціоналу системи перспективними моделями на базі архітектури Transformer із механізмами уваги для глибшого контекстного аналізу.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Survey of intrusion detection systems: techniques, datasets and challenges / A. Khraisat et al. *Cybersecurity*. 2019. Vol. 2, no. 1. P. 20. URL: <https://doi.org/10.1186/s42400-019-0038-7> (date of access: 17.05.2026).

2. Aldweesh A., Derhab A., Emam A. Z. Deep learning approaches for anomaly-based intrusion detection systems: a survey, taxonomy, and open issues. *Knowledge-Based Systems*. 2020. Vol. 189. Article 105124. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0950705119304897> (date of access: 17.05.2026).

3. Deep learning approach for intelligent intrusion detection system / R. Vinayakumar et al. *IEEE Access*. 2019. Vol. 7. P. 41525–41550. URL: <https://doi.org/10.1109/ACCESS.2019.2895334> (date of access: 17.05.2026).

4. Staudemeyer R. C. Applying long short-term memory recurrent neural networks to intrusion detection. *South African Computer Journal*. 2015. Vol. 56, no. 1. P. 136–154. URL: <https://sacj.cs.uct.ac.za/index.php/sacj/article/view/248> (date of access: 17.05.2026).

5. A deep learning approach for intrusion detection using recurrent neural networks / C. Yin et al. *IEEE Access*. 2017. Vol. 5. P. 21954–21961. URL: <https://doi.org/10.1109/ACCESS.2017.2762418> (date of access: 17.05.2026).

6. Sharafaldin I., Lashkari A. H., Ghorbani A. A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP)*. 2018. P. 108–116. URL: <https://doi.org/10.5220/0006639801080116> (date of access: 17.05.2026).

7. Moustafa N., Slay J. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). *2015 Military Communications and Information Systems Conference (MilCIS)*. 2015.

P. 1–6. URL: <https://dblp.org/db/conf/milcis/milcis2015.html> (date of access: 17.05.2026).

8. Farhan B. I., Jasim A. D. Survey of intrusion detection using deep learning in the internet of things. *Iraqi Journal for Computer Science and Mathematics*. 2022. URL: <https://ijcsm.researchcommons.org> (date of access: 17.05.2026).

9. Physical-informed neural network in signal processing and network traffic communications / O. V. Zolotukhin et al. *System Research and Information Technologies*. 2026. No. 1. P. 155–169. URL: <https://doi.org/10.20535/SRIT.2308-8893.2026.1.11> (date of access: 17.05.2026).

10. Intrusion detection system: A comprehensive review / H.-J. Liao et al. *Journal of Network and Computer Applications*. 2013. Vol. 36, no. 1. P. 16–24. URL: <https://doi.org/10.1016/j.jnca.2012.09.004> (date of access: 17.05.2026).

11. Cybersecurity data science: an overview from machine learning perspective / I. H. Sarker et al. *Journal of Big Data*. 2020. Vol. 7, no. 1. P. 41. URL: <https://doi.org/10.1186/s40537-020-00318-5> (date of access: 17.05.2026).

12. Buczak A. L., Guven E. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*. 2016. Vol. 18, no. 2. P. 1153–1176. URL: <https://doi.org/10.1109/COMST.2015.2494502> (date of access: 17.05.2026).

13. Deep learning-driven methods for network-based intrusion detection systems: a systematic review / J. Cho et al. *ICT Express*. 2025. Vol. 11, no. 1. P. 181–215. URL: <https://www.sciencedirect.com/science/article/pii/S2405959525000050> (date of access: 17.05.2026).

14. Deep learning-based intrusion detection systems: a survey / Z. Xu et al. *arXiv preprint*. 2025. URL: <https://arxiv.org/abs/2504.07839> (date of access: 17.05.2026).