

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

СИСТЕМА ВИЯВЛЕННЯ АНОМАЛІЙ НА ОСНОВІ НЕЙРОННИХ МЕРЕЖ

**Текстова частина до дипломної роботи
за спеціальністю 122 «Комп'ютерні науки»**

Керівник дипломної роботи
к.т.н., доц. Савченко Т.В.

(підпис)

“ ____ ” _____ 2025 р.

Виконала студентка
Томенко Н.Д

“ ____ ” _____ 2025 р.

Київ 2025

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ

Зав. кафедри інформатики,

к.т.н., доцент

_____ С.С. Гороховський

(підпис)

» _____ 2025 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на дипломну роботу

студентці 4-го курсу, факультету інформатики

Томенко Наталі Дмитрівні

ТЕМА Система виявлення аномалій на основі нейронних мереж

Зміст ТЧ до бакалаврської роботи:

Зміст

Анотація

Вступ

1 Аналіз досліджень виявлення аномалій за допомогою нейронних мереж

2 Нейромережеві підходи до розпізнавання аномалій

3 Побудова системи виявлення аномалій

Висновки

Список літератури

Додатки

Дата видачі „ 14 ” жовтня 2024 р. Керівник _____

(підпис)

Завдання отримала _____

(підпис)

Тема: Система виявлення аномалій на основі нейронних мереж

Календарний план виконання роботи:

№ п/п	Назва етапу дипломного проекту (роботи)	Термін виконання етапу	Примітка
1.	Отримання завдання на дипломну роботу.	14.10.2024	Виконано
2.	Пошук та огляд літератури за темою роботи	30.12.2024	Виконано
3.	Написання першого розділу текстової частини роботи щодо аналізу предметної області та літературних джерел	30.01.2025	Виконано
4.	Написання другого розділу текстової частини роботи щодо дослідження використання моделей штучних нейронних мереж у системах виявлення аномалій	15.03.2025	Виконано
5.	Виконання практичної частини роботи з розробки програмної реалізації системи виявлення аномалій	30.04.2025	Виконано
6.	Аналіз отриманих результатів та написання третього розділу текстової частини роботи	30.04.2025	Виконано
7.	Оформлення роботи	15.05.2025	Виконано
8.	Створення слайдів до дипломної роботи	25.05.2025	Виконано
9.	Попередній захист роботи	28.05.2025	Виконано

Студентка Томенко Наталя Дмитрівна

Керівник Савченко Тетяна Віталіївна

“ _____ ”

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1. АНАЛІЗ ДОСЛІДЖЕНЬ ВИЯВЛЕННЯ АНОМАЛІЙ ЗА ДОПОМОГОЮ НЕЙРОННИХ МЕРЕЖ	10
1.1 Застосування нейронних мереж для виявлення аномалій	10
1.1.1 Еволюція підходів до виявлення аномалій	10
1.1.2 Проблема виявлення аномалій	12
1.2 Теоретичні засади та наукові підходи	12
1.2.1 Фундаментальні дослідження виявлення аномалій	13
1.2.2 Ключові проблеми та потенційні шляхи розвитку	15
1.3 Постановка завдання дослідження	15
РОЗДІЛ 2. НЕЙРОМЕРЕЖЕВІ ПІДХОДИ ДО РОЗПІЗНАВАННЯ АНОМАЛІЙ	17
2.1 Аналіз архітектур нейронних мереж для задачі виявлення аномалій	17
2.1.1 CNN	18
2.1.2 RNN	19
2.1.3 Autoencoder	20
2.2 Ідентифікація аномалій мережевого трафіку засобами нейронних мереж	22
2.2.1 Базові концепції виявлення аномалій	22
2.2.2 Аномалії мережевого трафіку	25
2.2.3 Системи виявлення вторгнень	26
2.2.4 Проблемні аспекти виявлення мережевих аномалій	27
РОЗДІЛ 3. ПОБУДОВА СИСТЕМИ ВИЯВЛЕННЯ АНОМАЛІЙ	30
3.1 Експериментальне дослідження архітектур нейронних мереж	30
3.1.1 Використані технології	30
3.1.2 Вибір та підготовка набору даних	30
3.1.3 Методи оцінки моделей виявлення аномалій	32
3.1.4 Експериментальне дослідження CNN архітектур	34
3.1.5 Експериментальне дослідження LSTM архітектур	42
3.1.6 Експериментальне дослідження архітектур автоенкодерів	50
3.1.7 Аналіз та порівняння ефективності різних архітектур нейронних мереж на наборі NSL-KDD	57
3.2 Дослідження узагальнення та трансферності моделей нейронних мереж	59
3.2.1 Вибір та підготовка набору даних	60
3.2.2 Експеримент з прямим перенесенням моделей без донавчання	62
3.2.3 Експеримент з донавчанням моделей на датасеті UNSW-NB15	68
3.2.4 Оцінка обчислювальної ефективності моделей	73
3.2.5. Аналіз дослідження трансферності нейромережевих моделей	76

3.3 Рекомендації щодо вибору архітектури та побудови системи виявлення аномалій	78
3.3.1 Критерії вибору нейромережевої архітектури	78
3.3.2 Архітектурні особливості моделей та їх вплив на ефективність	80
3.3.3 Підходи до трансферного навчання моделей	81
3.3.4 Рекомендації щодо побудови повноцінної системи виявлення мережевих аномалій	82
ВИСНОВКИ	84
ВИКОРИСТАНІ ДЖЕРЕЛА	86
ДОДАТОК	92

АНОТАЦІЯ

Дипломна робота присвячена дослідженню ефективності архітектур нейронних мереж для виявлення мережевих аномалій та розробці рекомендацій щодо їх практичного застосування. Проведено аналіз літератури з виявлення аномалій, систематизовано типи мережевих аномалій та досліджено архітектури CNN, LSTM, GRU та автоенкодерів. Експериментально протестовано дев'ять моделей на наборі NSL-KDD. Найвищу точність показав GRU Autoencoder (F1 Score 98,5%), серед класифікаційних моделей – Deep CNN (90,9%) та Bidirectional LSTM (88,4%). Досліджено трансферність моделей на UNSW-NB15. Bidirectional LSTM демонструє найкращу адаптивність (F1 Score 95,99% при донавчанні на 1% даних), CNN потребують 20% даних, автоенкодери показують обмежену трансферність. Проаналізовано обчислювальну ефективність: CNN оптимальні для пакетної обробки, рекурентні мережі забезпечують баланс точності та швидкості, автоенкодери найбільш компактні. Розроблено рекомендації щодо вибору архітектури залежно від типу аномалій, стабільності середовища та ресурсних обмежень.

Результати можуть використовуватись для розробки систем виявлення мережевих аномалій та подальших досліджень у галузі.

Ключові слова: виявлення аномалій, нейронні мережі, мережеві аномалії, мережевий трафік, CNN, LSTM, автоенкодери, Autoencoders.

ВСТУП

Виявлення аномалій, також відоме як виявлення викидів, є важливим напрямом аналізу даних, який зосереджується на ідентифікації екземплярів або закономірностей, що суттєво відхиляються від очікуваної поведінки. Такі відхилення можуть вказувати на критичні проблеми або аномальні ситуації, що потребують особливої уваги. Це робить виявлення аномалій важливим компонентом у багатьох галузях. Зокрема, у фінансовому секторі системи виявлення аномалій дозволяють ідентифікувати шахрайські транзакції шляхом розпізнавання патернів, які відхиляються від встановлених норм [1]. У медицині моніторинг життєвих показників пацієнтів може виявляти аномалії, що сигналізують про погіршення стану здоров'я [2]. У кібербезпеці аналіз мережевого трафіку на предмет аномалій є важливим інструментом для попередження потенційних атак [3].

Виявлення аномалій активно досліджується протягом понад пів століття, починаючи з 1960-х років [4]. Традиційно цей процес ґрунтувався на статистичних методах, методах кластеризації та системах на основі правил, які були ефективними для низьковимірних та структурованих наборів даних [5]. Однак традиційні підходи виявляються недостатньо адаптивними до зростаючої складності та обсягів сучасних даних, що створює труднощі у масштабуванні та підвищенні продуктивності систем. Як перспективне рішення для виявлення аномалій було запропоновано використання нейронних мереж. Завдяки здатності моделювати та вивчати приховані складні закономірності в даних, нейронні мережі часто демонструють кращі результати порівняно з традиційними методами. Зокрема, такі архітектури, як згорткові нейронні мережі (CNN) та рекурентні нейронні мережі (RNN), активно застосовуються для виявлення аномалій у різних типах вхідних даних, таких як зображення, аудіо та відео [6]. Останні дослідження [7] також свідчать про трансформаційний вплив глибинного навчання на завдання виявлення аномалій, зокрема завдяки використанню таких технік, як автоенкодер

та генеративні змагальні мережі (GAN). Ці методи суттєво покращили здатність моделювати складні закономірності у даних та ідентифікувати аномалії. Водночас ефективність моделей нейронних мереж значною мірою залежить від типу аномалії, джерела даних та контексту застосування, що формує перспективний напрям для подальших досліджень.

Метою даної роботи є дослідження систем виявлення аномалій, застосування в них нейромережових моделей, проведення глибинного аналізу методів виявлення аномалій на основі нейронних мереж та оцінка їх ефективності в різних контекстах з метою розробки практичних рекомендацій щодо вибору оптимальної архітектури.

Об'єктом дослідження є процеси виявлення аномалій у мережевому трафіку на основі нейронних мереж.

Предметом дослідження є архітектури нейронних мереж для виявлення мережових аномалій, їх порівняльна ефективність, трансферність між різними доменами даних та обчислювальна складність.

Завданням дипломної роботи є:

1. Аналіз наукової літератури та існуючих досліджень на тему використання нейронних мереж у виявленні аномалій.
2. Дослідження теоретичних основ виявлення аномалій та застосування систем виявлення аномалій у різних контекстах.
3. Дослідження архітектур нейронних мереж, що використовуються для виявлення аномалій, а також підходів та особливостей їх застосування.
4. Експериментальне порівняння ефективності різних архітектур нейронних мереж (CNN, LSTM, GRU, автоенкодер) для виявлення мережових аномалій.
5. Дослідження трансферності та адаптивності моделей нейронних мереж при зміні мережевого середовища та оцінка їх обчислювальної ефективності.

6. Розробка практичних рекомендацій щодо вибору архітектури нейронної мережі та побудови системи виявлення аномалій залежно від конкретних вимог та обмежень.

Наукова новизна отриманих результатів полягає у:

1. Комплексному порівняльному аналізу різних архітектур нейронних мереж для виявлення мережевих аномалій в єдиних експериментальних умовах
2. Систематичному дослідженні трансферності нейромережевих моделей виявлення аномалій між різними доменами мережевих даних
3. Розробці науково обґрунтованої методології вибору архітектури нейронної мережі для систем виявлення аномалій на основі багатокритеріального аналізу, що враховує тип аномалій, стабільність середовища, доступність даних та обчислювальні обмеження.

РОЗДІЛ 1. АНАЛІЗ ДОСЛІДЖЕНЬ ВИЯВЛЕННЯ АНОМАЛІЙ ЗА ДОПОМОГОЮ НЕЙРОННИХ МЕРЕЖ

1.1 Застосування нейронних мереж для виявлення аномалій

1.1.1 Еволюція підходів до виявлення аномалій

Виявлення аномалій виникло в галузі класичної статистики, де основна увага приділялася виявленню викидів або спостережень, які суттєво відхилялися від більшої частини набору даних. [4] Прості методи включали обчислення таких показників, як середнє значення та стандартне відхилення, для визначення точок даних, що виходять за межі певного діапазону на основі припущених розподілів даних. Проте, хоча традиційні методи виявлення аномалій забезпечили фундаментальну основу для виявлення викидів у даних, з часом вони почали все помітніше демонструвати свої обмеження. Це стимулювало розвиток нових підходів, які могли б краще справлятися зі зростаючою складністю даних, їх високою розмірністю, наявністю нелінійних залежностей та різними типами аномалій.

На противагу традиційним підходам з'явилися методи, що базуються на відстані та щільності (Distance-Based та Density-Based methods). Ці підходи ідентифікують аномалії на основі їхньої віддаленості від інших точок даних або низької локальної щільності в певному просторі. Методи, такі як Local Outlier Factor (LOF) або k-Nearest Neighbors (k-NN) на основі відстані, дозволили виявляти аномалії без жорстких припущень про глобальний розподіл даних, концентруючись на локальних відхиленнях [5].

Згодом, з подальшим розвитком комп'ютерних технологій та появою нових алгоритмів машинного навчання, стався ще один крок вперед для сфери виявлення аномалій. Методи машинного навчання дозволили будувати складніші моделі нормальної поведінки даних, навчаючись безпосередньо з даних та адаптуючись до їхньої структури. Були розроблені такі підходи, як методи опорних векторів

(зокрема, One-Class SVM), ізоляційні ліси (Isolation Forest), а пізніше і методи на основі нейронних мереж та глибинного навчання, що продемонстрували величезні можливості у вивченні виразних представлень складних даних, таких як багатовимірні дані, часові дані, просторові дані та графічні дані, розширюючи межі різних навчальних завдань [8].

Впровадження технологій штучного інтелекту (ШІ) та машинного навчання (МН) у виявлення аномалій революціонізує цю галузь, пропонуючи потужний набір інструментів та методологій, які значно перевершують можливості традиційних методів. Ці передові алгоритми розроблені для обробки багатовимірних та великомасштабних даних, що робить їх добре придатними для сучасних застосувань, які часто пов'язані з великими даними та потоковою аналітикою. Одним із найважливіших [9] досягнень є впровадження методів напівконтрольованого та неконтрольованого навчання. Ці моделі не потребують повністю маркованого набору даних для навчання, що є особливо вигідним у сценаріях, де маркування є дорогим або непрактичним. Це відкриває нові можливості для виявлення аномалій у таких галузях, як кібербезпека, де атаки постійно розвиваються, а ручне маркування швидко застаріває.

Крім того, моделі штучного інтелекту та машинного навчання все більше стають здатними до навчання в режимі реального часу, що є критично важливою вимогою в динамічних середовищах. Наприклад, алгоритми навчання з підкріпленням можуть взаємодіяти зі своїм середовищем, адаптуючи стратегії виявлення аномалій у міру отримання більшої кількості інформації [10]. Це безцінно в таких застосуваннях, як автономне водіння та мережева безпека в режимі реального часу, де вартість невиявлення аномалії може бути катастрофічною.

1.1.2 Проблема виявлення аномалій

Вирішення проблеми виявлення аномалій на абстрактному рівні має низку складнощів [5]:

- Визначення області, яка охоплює кожен можливу нормальну поведінку, є дуже складним завданням.
- У багатьох галузях нормальна поведінка постійно змінюється
- Точне поняття аномалії відрізняється для різних галузей застосування, наприклад у тому, яке відхилення від норми є допустимим.
- Доступність маркованих даних для навчання та перевірки моделей часто проблемна.
- Зашумованість даних.

Іншими словами, вирішення проблеми виявлення аномалій у її найбільш загальному вигляді є надзвичайно складною задачею, через що всі існуючі дослідження фокусуються за застосуванні технік виявлення аномалій лише в межах конкретного домену.

Для даної роботи предметна область була звужена до дослідження мережевих аномалій. Саме такий вибір обумовлюється кількома причинами. Перша – наявність досить великої, порівняно з іншими суб-доменами, кількості літературних джерел для дослідження [1; 3; 11-15]. Друга – наявність в публічному доступі достатньої кількості наборів даних для навчання моделей - KDD Cup 99, DARPA 1998, UNSW-NB15, CICIDS2017, та інші.

1.2 Теоретичні засади та наукові підходи

Академічні дослідження із застосування нейронних мереж для задач виявлення аномалій значно зросли за кількістю та популярністю протягом останнього десятиліття, що підкреслює актуальність та перспективність цієї галузі. За даними систематичного аналізу Filho та співавторів [16], у період 2017-2021 років було опубліковано понад 1400 наукових праць на тему

застосування нейронних мереж для задачі виявлення аномалій. Однак глибина, широта та деталізація цих досліджень значно варіюються: більшість із них пропонують конкретні підходи до розв'язання окремих завдань.

Досліджень, які б одночасно і мали на меті структурувати інформацію щодо виявлення аномалій, і робили б це з акцентом на застосування нейронних мереж не так багато [8-9, 16-22].

1.2.1 Фундаментальні дослідження виявлення аномалій

Базовою працею у сфері виявлення аномалій вважається огляд Chandola et al. [5], який пропонує структуровану таксономію та визначає три основні типи аномалій: точкові, контекстуальні та колективні. Ця класифікація стала фундаментом для подальших досліджень, хоча робота не зосереджена саме на нейромережевих методах, оскільки була опублікована до періоду активного розвитку глибинного навчання.

У контексті мережевих аномалій Ahmad et al. [20] та Ahmed et al. [18] пропонують узагальнені фреймворки для виявлення аномалій у мережевому трафіку. Ahmed et al. [18] розглядають застосування виявлення аномалій не лише в контексті безпеки, але й для моніторингу продуктивності та управління мережами, демонструючи широкий спектр практичних застосувань. Ґрунтовні огляди мережевих систем виявлення вторгнень на основі аномалій представлені у роботах Bhuayan et al. (2014) [2] та García-Teodoro et al. [3], які систематизують архітектури та компоненти таких систем.

Аналіз наукової літератури щодо застосування нейронних мереж для виявлення аномалій дозволяє виокремити декілька ключових напрямків досліджень, які мають різний ступінь релевантності та достатності для цільового фокусу даної роботи – системи виявлення мережевих аномалій.

Перший напрямок представлений фундаментальними працями, що пропонують таксономії та систематизацію методів. Дослідження Pang et al. [8]

формує концептуальний базис, класифікуючи методи глибинного навчання для виявлення аномалій, однак не зосереджується спеціально на мережевих аномаліях. Аналогічно, огляд Filho et al. [16] надає цінні метрики щодо розподілу публікацій за тематиками, але має слабку деталізацію стосовно технічних особливостей імплементації.

Другий напрямок фокусується безпосередньо на нейромережевих архітектурах для виявлення мережевих аномалій. Праці Kwon et al. [22] та Ahmad et al. [20] пропонують порівняльний аналіз різних архітектур, що найбільш релевантно тематиці даного дослідження. Проте ці роботи недостатньо розкривають питання адаптації моделей до мережевого середовища.

Третій напрямок досліджень орієнтований на конкретні типи нейронних мереж. Наприклад, Naseer et al. [17] детально аналізують автоенкодерів, а Liu et al. [21] – гібридні CNN-RNN архітектури. Такі дослідження надають критично важливу інформацію щодо технічних аспектів імплементації, проте жодне з них не пропонує комплексного рішення для балансування між обчислювальною ефективністю та точністю виявлення.

Інші дослідження [14-15, 23-26] фокусуються на вузких аспектах застосування нейронних мереж для виявлення специфічних типів мережевих аномалій, що доповнює загальну картину, але не вирішує ключових обмежень в існуючій літературі: (1) недостатнього вивчення методів адаптації до динамічних змін; (2) відсутності оптимального балансу між обчислювальною ефективністю та точністю; (3) обмеженої інтерпретованості результатів; (4) неефективності при виявленні складних колективних аномалій.

Таким чином, наявна література формує теоретичний фундамент, проте залишає простір для внеску в практичну імплементацію адаптивних, обчислювально ефективних систем виявлення мережевих аномалій на основі нейронних мереж.

1.2.2 Основні виклики та перспективи розвитку

Аналіз літератури виявив кілька суттєвих прогалин у дослідженнях та викликів, які залишаються актуальними:

1. *Проблема узагальнення моделей* – більшість досліджень зосереджується на конкретних наборах даних та типах аномалій, але недостатньо уваги приділяється можливості переносу моделей між різними середовищами.
2. *Динамічна адаптація до змін мережевого середовища* – концептуальний дрейф залишається суттєвим викликом для систем виявлення аномалій.
3. *Інтерпретованість результатів* – інтерпретованість є критичним викликом для нейромережових систем виявлення аномалій, особливо у контексті безпеки.
4. *Виявлення нульового дня та невідомих атак* – відзначається, що більшість моделей демонструє значно нижчу ефективність при виявленні атак, які не були представлені в навчальних даних.
5. *Баланс між точністю та обчислювальною ефективністю* – складність застосування нейромережових моделей у режимі реального часу досі лишається проблемною.

Подолання цих викликів суттєво сприятиме зростанню ефективності та прикладної цінності моделей виявлення аномалій, побудованих із використанням нейронних мереж.

1.3 Постановка завдання дослідження

Аналіз наукової літератури виявив відсутність систематичного порівняння різних архітектур нейронних мереж для виявлення мережових аномалій в єдиних експериментальних умовах. Більшість досліджень фокусуються на окремих архітектурах або специфічних типах атак, що ускладнює об'єктивний вибір оптимального рішення для практичного застосування. Це визначає необхідність

проведення комплексного порівняльного дослідження основних типів нейромережових архітектур.

Мета дослідження: експериментальне порівняння ефективності різних архітектур нейронних мереж для виявлення мережових аномалій за різними характеристиками з метою систематизації знань та відповідей на відкриті питання.

Завдання дослідження:

- Аналіз архітектур нейронних мереж для виявлення аномалій на основі наявних літературних джерел.
- Експериментальне порівняння архітектур нейронних мереж в стабільних умовах.
- Дослідження узагальнення, трансферності та оптимізації балансу між точністю та обчислювальною складністю для різних архітектур нейронних мереж.
- Формування практичних рекомендацій щодо застосування нейронних мереж у системах виявлення аномалій.

РОЗДІЛ 2. НЕЙРОМЕРЕЖЕВІ ПІДХОДИ ДО РОЗПІЗНАВАННЯ АНОМАЛІЙ

2.1 Аналіз архітектур нейронних мереж для задачі виявлення аномалій

У контексті виявлення аномалій нейронні мережі (НМ) являють собою обчислювальні моделі, які демонструють особливу ефективність при роботі з високорозмірними даними та складними нелінійними залежностями. На відміну від традиційних статистичних методів, НМ здатні автоматично вивчати приховані закономірності в даних, що робить їх особливо придатними для ідентифікації нетипових патернів у мережевому трафіку. Розвиток від простих одношарових до глибоких багатошарових архітектур дозволив подолати обмеження лінійної роздільності та створити моделі, ефективні для виявлення складних аномалій [10; 27].

Сучасні архітектури нейронних мереж будуються з різних типів шарів, кожен з яких виконує специфічні функції. Згорткові шари спеціалізуються на виявленні локальних патернів, рекурентні – на аналізі послідовностей, а повнозв'язні – на інтеграції ознак [8]. Ця модульність дозволяє створювати спеціалізовані архітектури для різних типів задач виявлення аномалій, комбінуючи переваги різних типів шарів відповідно до характеру аналізованих даних.

Аналіз літератури показує, що для задачі виявлення мережевих аномалій найчастіше застосовуються три основні класи архітектур: згорткові нейронні мережі (CNN), рекурентні мережі з їх варіаціями (RNN, LSTM, GRU) та різні типи автоенкодерів [8, 16]. Кожна з цих архітектур має свої переваги залежно від типу аналізованих даних та характеру аномалій, що підлягають виявленню. Розглянемо детальніше принципи роботи цих архітектур та їх застосування для аналізу мережевого трафіку.

2.1.1 CNN

Згорткові нейронні мережі (CNN) у контексті виявлення аномалій використовуються для вилучення локальних патернів із структурованих даних. Їх перевага над повнозв'язними мережами полягає в ефективному використанні просторової структури даних через механізми локальних зв'язків та спільного використання параметрів [28]. Основними компонентами CNN є згорткові шари, які застосовують набір фільтрів для виявлення специфічних патернів у даних, та підвибіркові шари (pooling), що зменшують розмірність представлення, зберігаючи найбільш важливі ознаки [29]. Така архітектура дозволяє ієрархічно вивчати ознаки різного рівня абстракції – від простих байтових сигнатур до складних структурних аномалій у мережевих пакетах.

Для виявлення мережевих аномалій найчастіше використовуються два типи CNN архітектур. Одновимірні CNN (1D-CNN) ефективно працюють з послідовностями пакетів або часовими рядами мережевого трафіку. Дослідження Wang та співавторів [30] показало, що 1D-CNN досягають високої точності (94%) при класифікації зашифрованого трафіку, демонструючи здатність виявляти приховані патерни навіть у заплутаних даних. Двовимірні CNN (2D-CNN) застосовуються, коли мережеві дані представлені у вигляді матриць, наприклад, при аналізі структури заголовків пакетів. Wang та співавтори [25] продемонстрували, що їх HAST-IDS система на основі 2D-CNN досягає високих показників виявлення атак завдяки здатності аналізувати просторово-часові залежності.

Ключовою перевагою CNN для виявлення аномалій є їх здатність автоматично навчатися виявляти релевантні ознаки без необхідності ручного проектування характеристик (feature engineering). Це особливо важливо для виявлення нових типів атак, сигнатури яких заздалегідь невідомі. Архітектури CNN для мережевих аномалій зазвичай включають кілька згорткових шарів із

функціями активації ReLU, шари підвибірки для зменшення розмірності, та повнозв'язні шари для фінальної класифікації [29].

2.1.2 RNN

Рекурентні нейронні мережі (RNN) у контексті виявлення мережевих аномалій спеціально призначені для аналізу послідовностей мережевих подій, де темпоральні залежності відіграють ключову роль. Здатність RNN зберігати інформацію про попередні стани робить їх особливо ефективними для виявлення аномалій, що проявляються як відхилення від нормальних патернів поведінки в часі [27]. У мережевому трафіку такі аномалії можуть включати поступові зміни в інтенсивності потоків, аномальні послідовності пакетів або багатоетапні атаки, що розгортаються протягом тривалого періоду. Механізм рекурентних зв'язків дозволяє RNN накопичувати контекстну інформацію про мережеву активність, що критично важливо для розпізнавання складних атак. Проте, базові RNN стикаються з проблемою зникаючих градієнтів при аналізі довгих послідовностей, для вирішення якої у виявленні мережевих аномалій використовуються спеціалізовані архітектури.

LSTM (Long Short-Term Memory) мережі стали стандартом для аналізу довгострокових патернів у мережевому трафіку. Механізм гейтів LSTM – вхідний, забування та вихідний – дозволяє селективно зберігати релевантну інформацію про нормальну поведінку мережі та виявляти відхилення [31]. Дослідження показують, що LSTM ефективно виявляють повільні сканування портів та розподілені атаки, які можуть тривати години або дні [23]. Наприклад, Kim та співавтори [32] продемонстрували, що LSTM-RNN досягають високої точності при аналізі журналів безпеки, виявляючи складні послідовності подій, характерні для просунутих персистентних загроз (APT).

GRU (Gated Recurrent Unit) представляють спрощену альтернативу LSTM з меншою обчислювальною складністю. Використання лише двох гейтів –

оновлення та скидання - робить GRU на 30% швидшими за LSTM при збереженні порівнянної точності виявлення аномалій [33]. Це робить GRU привабливим вибором для систем реального часу, де швидкість обробки критична. Лі та співавтори [34] показали, що GRU особливо ефективні для виявлення аномалій в IoT мережах, де обмежені обчислювальні ресурси вимагають балансу між точністю та ефективністю.

Порівняльні дослідження архітектур RNN для виявлення мережових аномалій виявляють, що вибір між LSTM та GRU залежить від конкретного застосування. LSTM демонструють кращі результати при аналізі складних багатоступінчатих атак з довгими часовими залежностями, тоді як GRU оптимальні для швидкого виявлення простіших аномалій у потоках реального часу [20]. Обидві архітектури значно перевершують базові RNN у здатності моделювати темпоральні патерни мережового трафіку та виявляти відхилення від нормальної поведінки.

2.1.3 Autoencoder

Автоенкодер (АЕ) у контексті виявлення мережових аномалій використовуються за принципом навчання на нормальних даних та ідентифікації відхилень через помилку реконструкції. Основна ідея полягає в тому, що автоенкодер, навчений відтворювати нормальний мережовий трафік, буде демонструвати високу помилку реконструкції при зустрічі з аномальними патернами [17]. Ця властивість робить автоенкодер особливо ефективними для неконтрольованого виявлення аномалій, коли мітки атак недоступні або нові типи атак ще невідомі. Архітектура автоенкодера складається з енкодера, який стискає вхідні дані до компактного латентного представлення, та декодера, що відновлює оригінальні дані з цього представлення [29]. Для виявлення аномалій використовується недокомплектований автоенкодер, де розмірність латентного простору менша за розмірність вхідних даних. Це змушує мережу вивчати

найбільш значущі ознаки нормального трафіку, ігноруючи шум та випадкові варіації. Аномалії, які не відповідають вивченим патернам, реконструюються з великою похибкою, що слугує індикатором їх наявності.

Різні варіанти автоенкодерів адаптовані для специфічних задач виявлення мережових аномалій. Шумопрігнічуючі автоенкодери (DAE) навчаються на спеціально зашумлених даних, що підвищує їх стійкість до варіацій у нормальному трафіку та покращує здатність розрізняти справжні аномалії від природних коливань [35]. Це особливо важливо в мережевому середовищі, де трафік характеризується значною варіативністю.

Конволюційні автоенкодери (ConvAE) ефективно працюють з просторово структурованими представленнями мережових даних, такими як матриці характеристик потоків або візуалізації пакетів [36]. Використання згорткових шарів дозволяє ConvAE виявляти локальні патерни в структурі мережових даних, що корисно для ідентифікації специфічних сигнатур атак. Yu та співавтори [24] продемонстрували, що стековані ConvAE з dilated convolutions досягають високої точності виявлення вторгнень завдяки здатності аналізувати ієрархічні ознаки мережевого трафіку.

Варіаційні автоенкодери (VAE) моделюють розподіл ймовірності латентних представлень, що дозволяє їм краще узагальнювати дані та виявляти тонкі аномалії [38]. Yang та співавтори [39] показали, що покращені умовні VAE підвищують ефективність класифікації вторгнень завдяки кращому моделюванню варіативності нормального трафіку. VAE особливо корисні для виявлення аномалій у середовищах з високою варіативністю легітимного трафіку, де традиційні автоенкодери можуть генерувати надмірну кількість хибних спрацювань.

Дослідження Naseer та співавторів [17] на датасеті NSL-KDD показало, що глибинні автоенкодери досягають AUC 0.94, перевершуючи інші архітектури при виявленні невідомих атак. Ключовою перевагою автоенкодерів є їх здатність

працювати без міток аномалій, що робить їх практичним рішенням для реальних мережових середовищ, де нові типи атак постійно еволюціонують.

2.2 Ідентифікація аномалій мережевого трафіку засобами нейронних мереж

Виявлення мережевих аномалій становить критично важливу галузь у межах систем моніторингу мереж та забезпечення безпеки, зосереджуючись на ідентифікації шаблонів, що значно відхиляються від встановленої нормальної поведінки мережевого трафіку [5]. Аномалії мережі можуть виникнути через такі причини, як перевантаження в мережі, несправність пристроїв, неправильна конфігурація мережі, зловмисна діяльність або мережеві вторгнення, які перехоплюють та інтерпретують звичайні мережеві служби [2]. В цьому розділі досліджуються концептуальні основи виявлення мережевих аномалій, їх застосування, а також принципи роботи.

2.2.1 Базові концепції виявлення аномалій

Виявлення мережевих аномалій функціонує на фундаментальному припущенні, що аномальна мережева активність проявляється у статистичних, поведінкових або структурних відхиленнях від встановлених базових показників нормальності [18]. Ці відхилення можуть свідчити про порушення безпеки, деградацію продуктивності, збої інфраструктури або інші значні події, що потребують втручання.

Пошук оптимальної архітектури систем виявлення аномалій триває, і в літературі було кілька спроб її узагальнення з різним рівнем деталізації, наприклад [2] та [3], представлені на Рис. 1 і Рис. 2 відповідно, а також загальний фреймворк виявлення мережевих аномалій [18], представлений на Рис. 3.

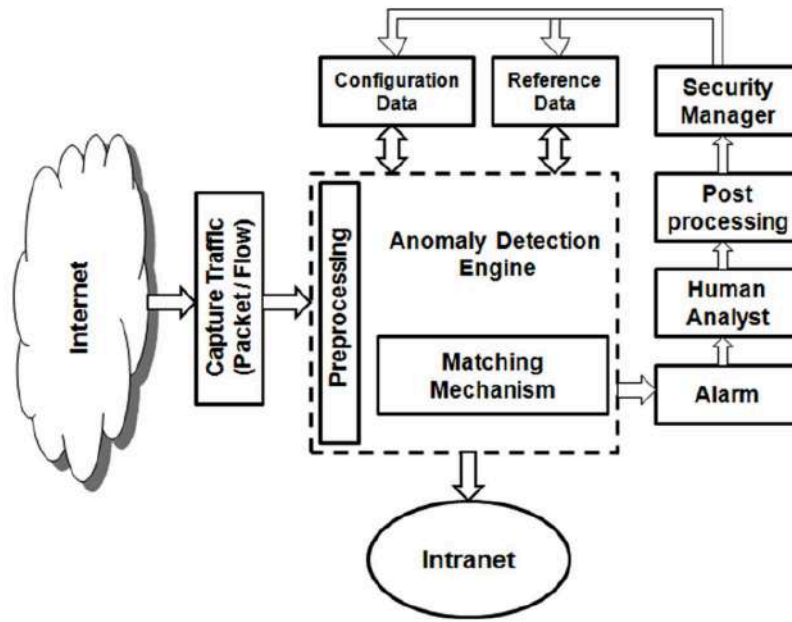


Рис. 1 – Загальна архітектура A-NIDS [2]

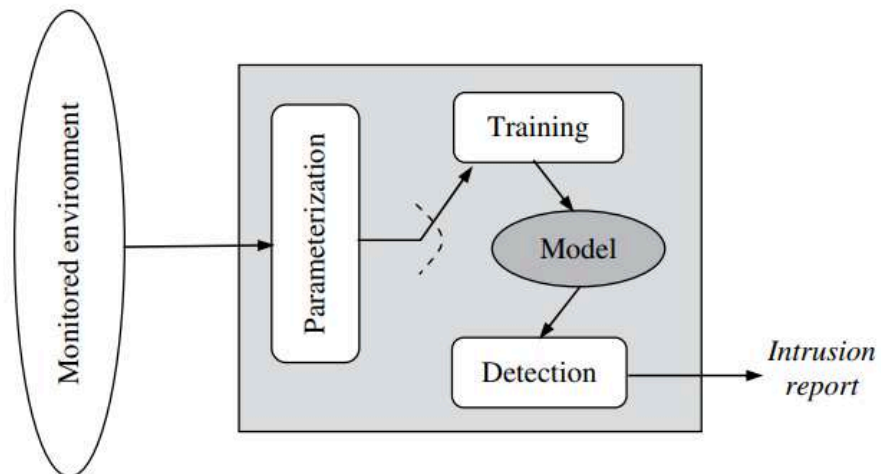


Рис. 2 – Загальна архітектура A-NIDS [3]

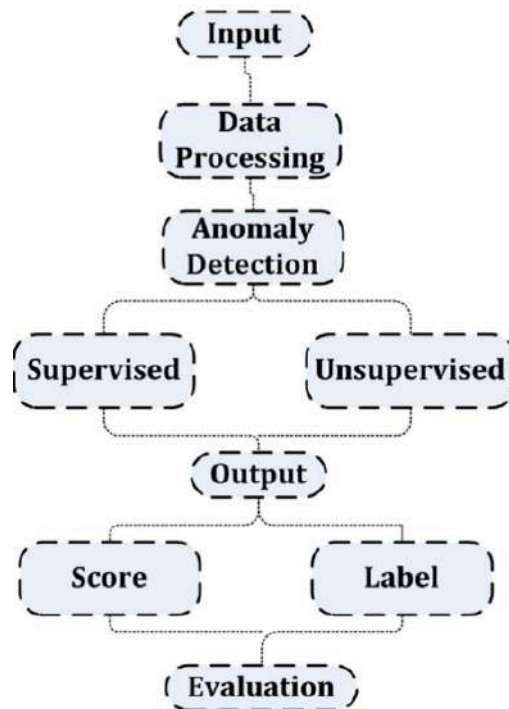


Рис. 3 – Загальний фреймворк для виявлення мережесих аномалій [18]

Проте, незалежно від конкретної системи виявлення аномалій, вони зазвичай працюють за схожим принципом, що охоплює кілька послідовних фаз:

1. Збір даних.
2. Попередня обробка даних.
3. Вилучення ознак.
4. Моделювання нормальної поведінки.
5. Виявлення аномалій.
6. Оцінка результатів.
7. Реагування.

Ефективність фреймворків виявлення аномалій сильно залежить від їхньої здатності моделювати нормальну мережеву поведінку, зберігаючи чутливість до незначних відхилень, що вказують на справжні аномалії [2].

2.2.2 Аномалії мережевого трафіку

У контексті розробки систем виявлення аномалій на основі нейронних мереж критично важливо розуміти природу та характеристики мережевих аномалій, оскільки різні типи аномалій вимагають різних архітектурних рішень [5]. Класифікація аномалій визначає вибір конкретної нейромережевої архітектури та методу навчання для її ефективного виявлення.

Типи аномалій за їх проявом у даних.

Точкові аномалії представляють собою окремі відхилення від нормального розподілу даних і найкраще виявляються автоенкодерами або класичними CNN, які аналізують локальні патерни [17, 25]. Ці аномалії проявляються як статистично значущі девіації в характеристиках окремих пакетів або з'єднань, наприклад, аномально великі пакети або підозрілі значення в заголовках.

Контекстуальні аномалії вимагають аналізу як поведінкових, так і контекстуальних атрибутів, що робить LSTM та GRU оптимальним вибором для їх виявлення [20, 40]. Chandola та співавтори [5] підкреслюють важливість розділення атрибутів на контекстуальні (час) та поведінкові (обсяг трафіку, частота запитів). Для прикладу, пікове навантаження о 3:00 ночі може бути нормальним для резервного копіювання, але аномальним для веб-сервера [41].

Колективні аномалії представляють найбільший виклик для виявлення, оскільки окремі компоненти можуть виглядати нормальними [5]. Гібридні архітектури CNN-LSTM показують найкращі результати при виявленні таких складних патернів, як DDoS-атаки або APT [21]. Casas та співавтори [42] демонструють, що розподілені атаки часто маскуються під легітимний трафік, що вимагає аналізу довгострокових темпоральних залежностей.

Класифікація за походженням та її вплив на вибір моделі.

Розуміння походження аномалій критично важливе для налаштування порогів виявлення та зменшення хибних спрацювань. Мережеві атаки, як найбільш

критичний тип аномалій, вимагають моделей з високою чутливістю. Дослідження показують, що ансамблі різних архітектур досягають найкращих результатів у виявленні складних атак [45].

Операційні події та сплески трафіку (flash crowds) представляють особливий виклик, оскільки є легітимними, але можуть бути помилково класифіковані як атаки. Junior та співавтори [41] підкреслюють необхідність адаптивних порогів для розрізнення між легітимними сплесками та DDoS-атаками. Автоенкодері з динамічними порогами показують хороші результати у цьому контексті [17].

Аномалії вимірювань, хоча й не є безпековими загрозами, можуть суттєво впливати на точність моделей виявлення. Попередня фільтрація таких аномалій або використання robust автоенкодерів допомагає підвищити якість виявлення справжніх загроз [35].

2.2.3 Системи виявлення вторгнень

Системи виявлення вторгнень (IDS) є ключовим застосуванням технологій виявлення аномалій у мережевій безпеці [14]. Їхній різновид, мережеві системи виявлення вторгнень на основі аномалій (A-NIDS), використовують нейронні мережі для ідентифікації відхилень від нормальної поведінки мережі. Фундаментальна різниця між сигнатурними та аномальними IDS визначає переваги застосування нейронних мереж. Сигнатурні системи обмежені виявленням лише відомих атак, тоді як аномальні IDS здатні ідентифікувати zero-day атаки та нові варіації існуючих загроз. García-Teodoro та співавтори [3] підкреслюють, що саме ця здатність виявляти невідомі атаки робить аномальні IDS основним напрямком сучасних досліджень.

Нейронні мережі особливо ефективні в контексті A-NIDS через їх здатність:

- Автоматично вивчати складні патерни нормальної поведінки без явного програмування правил.

- Адаптуватися до змін у мережевому середовищі через механізми онлайн-навчання.
- Працювати з високорозмірними даними мережевого трафіку.
- Виявляти subtle аномалії, які можуть бути пропущені традиційними статистичними методами.

Проте, як зазначають Вhуан та співавтори [2], основним викликом для A-NIDS залишається баланс між чутливістю виявлення (True Positive Rate) та кількістю хибних спрацювань (False Positive Rate). Нейронні мережі демонструють покращення в цьому аспекті, але питання оптимізації цього балансу залишається актуальним для практичного застосування.

Інтеграція різних архітектур нейронних мереж в A-NIDS дозволяє адресувати специфічні типи загроз:

- Автоенкодера ефективні для виявлення аномалій при відсутності міток атак.
- CNN оптимальні для аналізу структури пакетів та виявлення сигнатур на низькому рівні.
- RNN/LSTM підходять для виявлення темпоральних аномалій та багатоетапних атак.

2.2.4 Проблемні аспекти виявлення мережевих аномалій

Застосування нейронних мереж для виявлення мережевих аномалій, попри їх переваги, стикається з рядом фундаментальних викликів, які впливають на ефективність практичної реалізації таких систем.

Проблема визначення нормальності. Junior та співавтори [41] ідентифікують створення точного уявлення про нормальність як основну невирішену проблему в галузі. Для нейронних мереж це особливо критично, оскільки вони навчаються на основі припущення про репрезентативність навчальних даних. Динамічна природа мережевого трафіку ускладнює формування стабільної моделі "норми", що може призводити до деградації точності моделі з часом.

"Прокляття розмірності" та обчислювальна складність. Chandola та співавтори [5] підкреслюють, що високорозмірність мережевих даних створює фундаментальні проблеми для алгоритмів виявлення аномалій. Для нейронних мереж це означає експоненційне зростання кількості параметрів та вимог до обчислювальних ресурсів. Глибокі архітектури, хоча й демонструють кращу точність, можуть бути непрактичними для розгортання в системах реального часу через їх обчислювальну складність.

Проблема незбалансованості класів. Аномалії за визначенням є рідкісними подіями, що створює суттєвий дисбаланс у навчальних даних. Це особливо проблематично для supervised навчання нейронних мереж, де моделі схильні до зміщення в бік мажоритарного класу (нормальний трафік). Хоча автоенкодери частково вирішують цю проблему через неконтрольоване навчання, вони все ще потребують репрезентативних даних для калібрування порогів виявлення.

Варіативність трафіку та хибні спрацювання. Дослідження [43-44] показують, що до 30% сповіщень систем виявлення аномалій спричинені природною варіативністю мережевого середовища. Для нейронних мереж це створює дилему між чутливістю виявлення та кількістю хибних спрацювань. Занадто чутливі моделі генерують надмірну кількість false positives, тоді як менш чутливі пропускають реальні загрози.

Обмеженість навчальних даних. Soltani та співавтори [48] відзначають критичну нестачу стандартизованих наборів даних для навчання та валідації моделей. Існуючі публічні датасети часто не відображають сучасні мережеві середовища та типи атак. Це обмежує можливості порівняння різних архітектур та ускладнює оцінку їх реальної ефективності.

Інтерпретованість результатів. "Чорний ящик" природи глибоких нейронних мереж створює серйозні проблеми для практичного застосування, особливо в критичній інфраструктурі. Адміністратори безпеки потребують не

лише виявлення аномалій, але й розуміння причин спрацювання системи для прийняття адекватних рішень.

Адаптація до шифрованого трафіку. Зростання частки шифрованого трафіку обмежує можливості аналізу вмісту пакетів, змушуючи моделі покладатися на метадані та статистичні характеристики. Це вимагає розробки спеціалізованих архітектур, здатних ефективно працювати з обмеженою інформацією.

РОЗДІЛ 3. ПОБУДОВА СИСТЕМИ ВИЯВЛЕННЯ АНОМАЛІЙ

3.1 Експериментальне дослідження архітектур нейронних мереж

Першим кроком до побудови системи виявлення аномалій є створення та навчання нейромережевої моделі. В цьому дослідженні підхід до вибору моделі полягає у тому аби експериментальним шляхом визначити найкращу архітектуру із запропонованих, і саме її використовувати для побудови системи з метою досягнення найкращих результатів.

3.1.1 Використані технології

Експериментальне дослідження проводилось на мові програмування Python версії 3.11 із використанням наступних бібліотек:

- TensorFlow, Keras – для нейронних мереж;
- Pandas – робота з даними;
- NumPy – числові обчислення;
- Scikit-learn – метрики та допоміжні функції;
- Matplotlib/Seaborn – візуалізація.

В якості середовища розробки був використаний JupyterLab.

3.1.2 Вибір та підготовка набору даних

В якості набору даних для тренування моделей був вибраний датасет NSL-KDD [65], що залишається найпопулярнішим еталонним набором даних для виявлення мережевих вторгнень. Як покращена версія KDD Cup 1999 [66], NSL-KDD усуває проблеми дублювання записів та незбалансованості класів. Набір має наступну структуру:

- 125,973 тренувальних записів;
- 22,544 тестувальних записів;
- Не містить дублікатів;

- 41 ознака – 9 базових, таких як тривалість з'єднання, протокол, тощо; 13 контекстних – таких як кількість помилок входу, кількість команд у потоці, тощо; 19 трафік-орієнтованих – наприклад, кількість з'єднань до тієї ж IP-адреси протягом останніх двох секунд;
- Найвні 22 типи атак: neptune, warezclient, ipsweep, portsweep, teardrop, nmap, satan, smurf, pod, back, guess_passwd, ftp_write, multihop, rootkit, buffer_overflow, imap, warezmaster, phf, land, loadmodule, spy, perl.

Під час обробки даних були проведені наступні маніпуляції:

- всі атаки погруповані в 4 класи до яких вони належать: Denial of Service (dos), Probing (probe), Remote to Local (r2l), User to Root (u2r). Результат розподілу атак в обох вибірках представлений на Рис. 4.

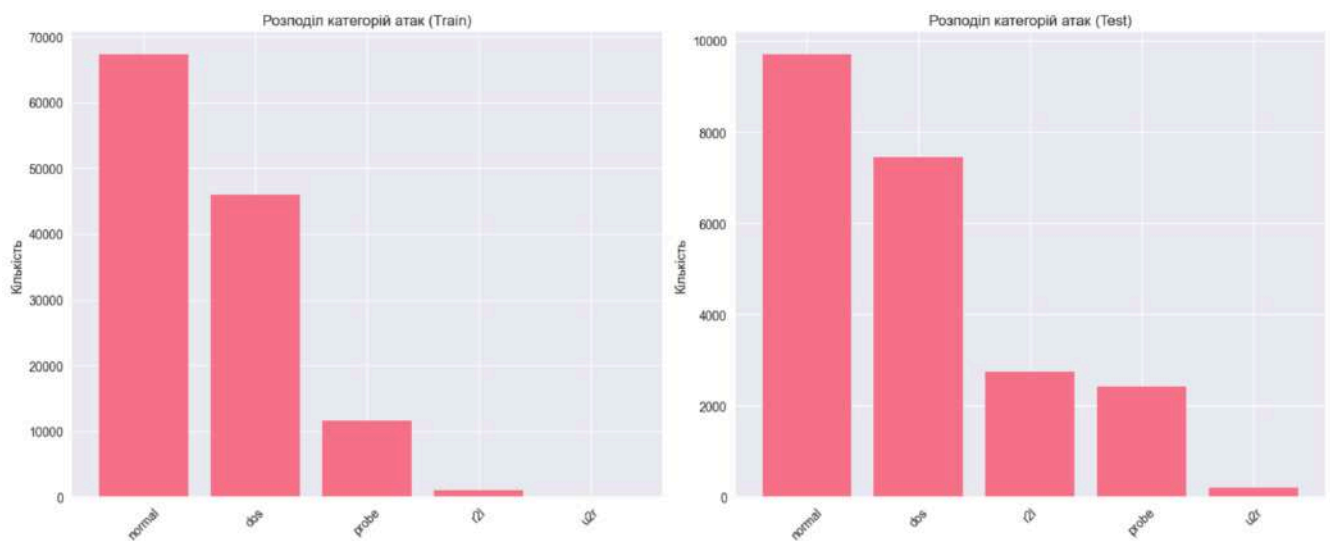


Рис. 4 – Розподіл класів атак у тренувальній та тестовій вибірках NSL-KDD

- Додане бінарне маркування (is_anomaly) для розділення даних на нормальний трафік (0) та аномалії (1).
- Застосовано one-hot кодування до категоріальних ознак, що перетворило їх на бінарні вектори і збільшило загальну кількість ознак з 41 до 122.
- Застосовано StandardScaler до всіх числових ознак, що трансформувало їх до розподілу з середнім 0 та стандартним відхиленням 1.

- Параметри масштабування (середнє та стандартне відхилення) збережені для подальшого застосування до тестових даних.
- Для *CNN* дані були перетворені до формату (samples, features, 1), що відповідає вимогам CNN.
- Для *LSTM* дані були трансформовані в часові послідовності фіксованої довжини (10 часових кроків) для аналізу часової динаміки.
- Для *автоенкодерів* були відібрані лише нормальні (не аномальні) дані для навчання, що відповідає принципу навчання на нормальних патернах.
- Розраховані ваги класів для боротьби з дисбалансом у навчальних даних.

Переваги NSL-KDD для навчання моделей у його структурованості та майже повній готовності для застосування на нейромережових моделях: попередньо виділені ознаки та розділення на тестову й навчальну вибірку. Незважаючи на обмеження NSL-KDD, пов'язані з віком даних на яких він ґрунтується (застарілі типи атак, відсутність сучасних протоколів), його використання залишається стандартом для початкової оцінки моделей виявлення аномалій. Однак, для практичного застосування систем виявлення аномалій, побудованих на NSL-KDD необхідна додаткова валідація на сучасніших датасетах, таких як CICIDS2017 або UNSW-NB15.

3.1.3 Методи оцінки моделей виявлення аномалій

Коли нормальний трафік складає 95-99% даних – традиційна метрика асигурації стає неінформативною [18], тому поширене застосування спеціалізованих метрик для оцінки ефективності моделей. Основні поняття, якими оперують при підрахунку метрик – пропущені атаки (False Negatives, FN) та хибні спрацювання (False Positives, FP).

Повнота (Recall) (1) визначає частку виявлених аномалій серед усіх реальних аномалій. Низьке значення означає, що модель пропускає значну кількість атак.

$$Recall = \frac{TP}{TP+FN} \quad (1)$$

Точність (*precision*) (2) визначає частку реальних аномалій серед усіх зразків, класифікованих як аномалії.

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

F1-оцінка (F-measure, F1-score) (3) є гармонійним середнім між Precision та Recall і забезпечує баланс між цими метриками. Значення F1 Score більше 0.9 прийнято вважати [69] відмінним, а значення 0.8 - 0.9 - хорошим. Проте, так як мережевий трафік є досить high-risk середовищем для виявлення аномалій, чим більше – тим краще.

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

Receiver Operating Characteristic (ROC) крива відображає залежність між True Positive Rate (Повнота, Recall) та False Positive Rate ($FPR = FP / (FP + TN)$) при різних порогових значеннях для класифікації. Дана крива дозволяє візуально оцінити компроміс між здатністю моделі виявляти аномалії та схильністю до помилкових спрацьовувань [19].

AUC-ROC є числовою характеристикою ROC кривої і приймає значення від 0 до 1, де 1 відповідає ідеальній моделі, а 0.5 — випадковому класифікатору [20]. Ця метрика широко використовується для порівняння різних моделей, оскільки вона нечутлива до конкретного порогового значення і дає загальну оцінку здатності моделі розрізняти класи. Дослідження [22] показує, що для моделей виявлення мережевих аномалій на основі нейронних мереж значення AUC-ROC вище 0.95 вважається відмінним результатом, тоді як значення нижче 0.85 зазвичай вказує на необхідність удосконалення моделі.

3.1.4 Експериментальне дослідження CNN архітектур

Базова архітектура CNN

Для встановлення базового рівня продуктивності було розроблено відносно просту CNN архітектуру з послідовним збільшенням кількості фільтрів (32→64→128) (Рис. 5).

Layer (type)	Output Shape	Param #
conv1d_13 (Conv1D)	(None, 120, 32)	128
max_pooling1d_6 (MaxPooling1D)	(None, 60, 32)	0
conv1d_14 (Conv1D)	(None, 58, 64)	6,208
max_pooling1d_7 (MaxPooling1D)	(None, 29, 64)	0
conv1d_15 (Conv1D)	(None, 27, 128)	24,704
global_average_pooling1d_3 (GlobalAveragePooling1D)	(None, 128)	0
dense_6 (Dense)	(None, 64)	8,256
dropout_3 (Dropout)	(None, 64)	0
dense_7 (Dense)	(None, 1)	65

Рис. 5 – Структура базової CNN

Для перевірки гіпотези щодо потреби у високій чутливості моделей для мережевого середовища, була імплементована функція пошуку оптимального порогу класифікації замість використання класичного на рівні 0.5. Оптимальність тут визначається на основі значення F1 Score. Результати тестування візуалізовані на Рис. 6. Оптимальним виявився поріг 0.004, за якого вдалося досягнути Accuracy 89%, precision 89%, recall 92%, F1 Score 90% та ROC AUC 89% (Рис. 7).

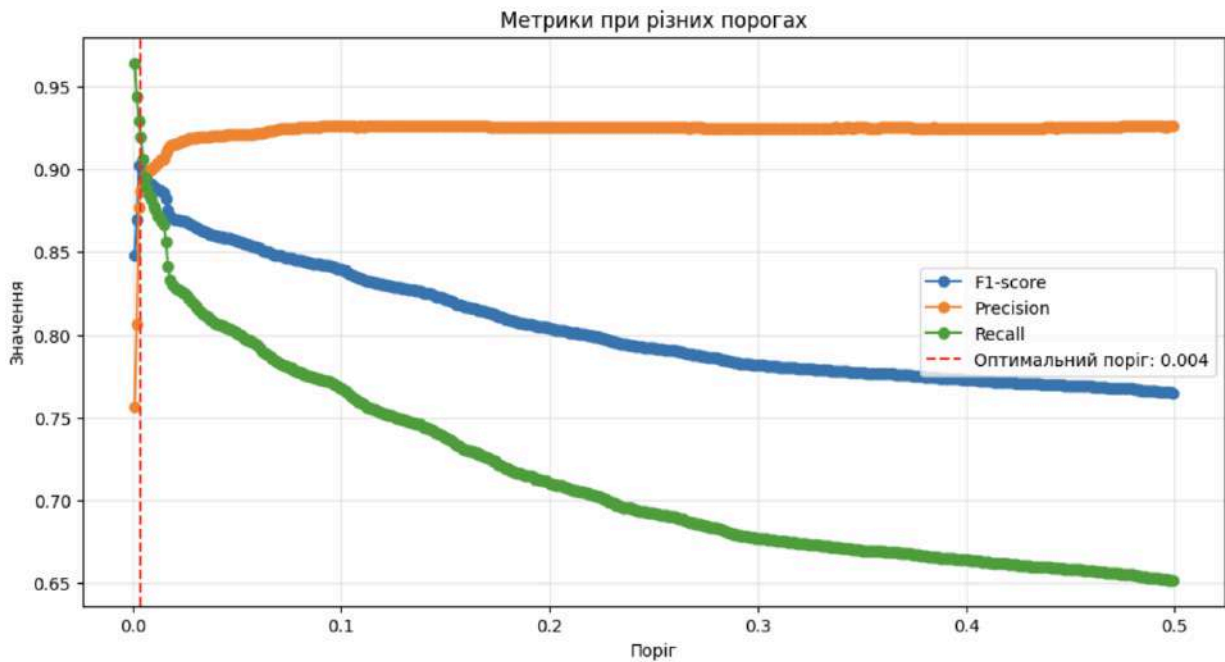


Рис. 6 – Результати базової CNN моделі для різних порогів

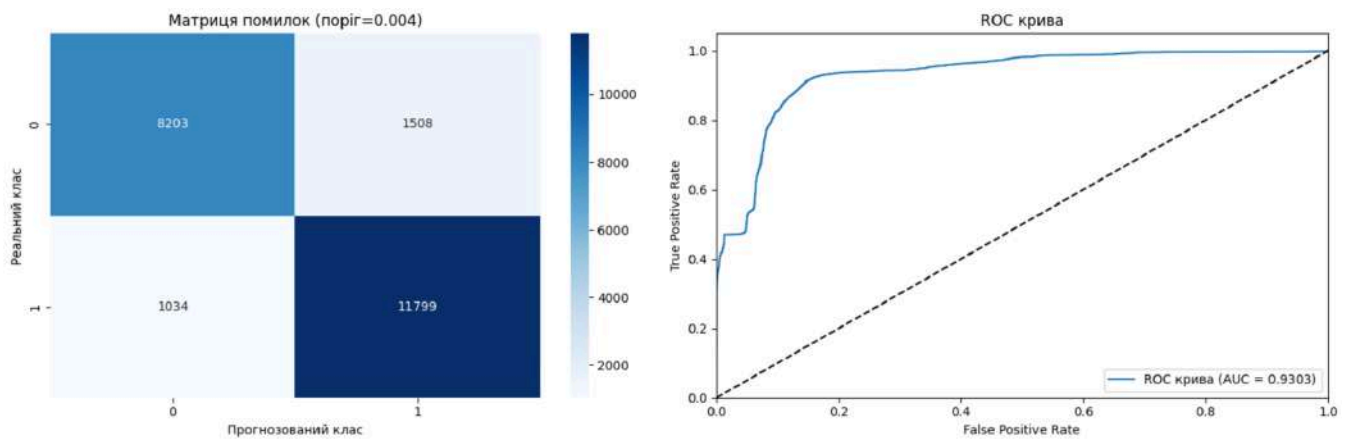


Рис. 7 – Результати базової CNN з порогом 0.004

Deep CNN архітектура

Deep CNN архітектура представляє глибший варіант CNN з більшою кількістю фільтрів та додатковим повнозв'язним шаром для більш складного моделювання взаємозв'язків між вилученими ознаками (Рис. 8). Вона містить більше шарів, більше фільтрів у згорткових шарах (64→128→256) для виявлення більшої кількості патернів різного рівня

абстракції та варіативні розміри ядер (3, 5, 3) для вилучення ознак різних масштабів.

Кращі результати при тестуванні моделі показав підхід із знаходженням оптимального порогу. Найоптимальнішим виявився поріг 0.035 (Рис. 9). Вдалось досягти наступних результатів: precision – 89%, recall – 93%, F1 Score – 90%, ROC AUC – 93%.

Layer (type)	Output Shape	Param #
conv1d_18 (Conv1D)	(None, 120, 64)	256
max_pooling1d_8 (MaxPooling1D)	(None, 60, 64)	0
conv1d_19 (Conv1D)	(None, 56, 128)	41,088
max_pooling1d_9 (MaxPooling1D)	(None, 28, 128)	0
conv1d_20 (Conv1D)	(None, 26, 256)	98,560
global_average_pooling1d_5 (GlobalAveragePooling1D)	(None, 256)	0
dense_10 (Dense)	(None, 128)	32,896
dropout_5 (Dropout)	(None, 128)	0
dense_11 (Dense)	(None, 64)	8,256
dropout_6 (Dropout)	(None, 64)	0
dense_12 (Dense)	(None, 1)	65

Рис. 8 – Структура Deep CNN

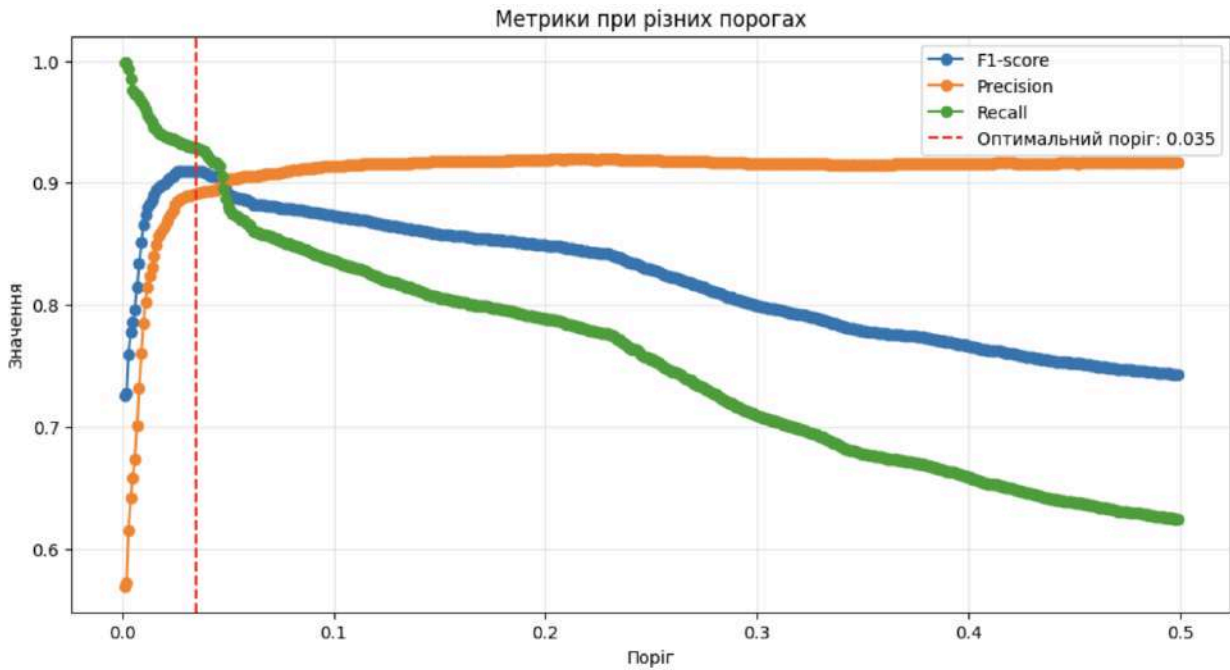


Рис. 9 – Результати моделі Deep CNN для різних порогів

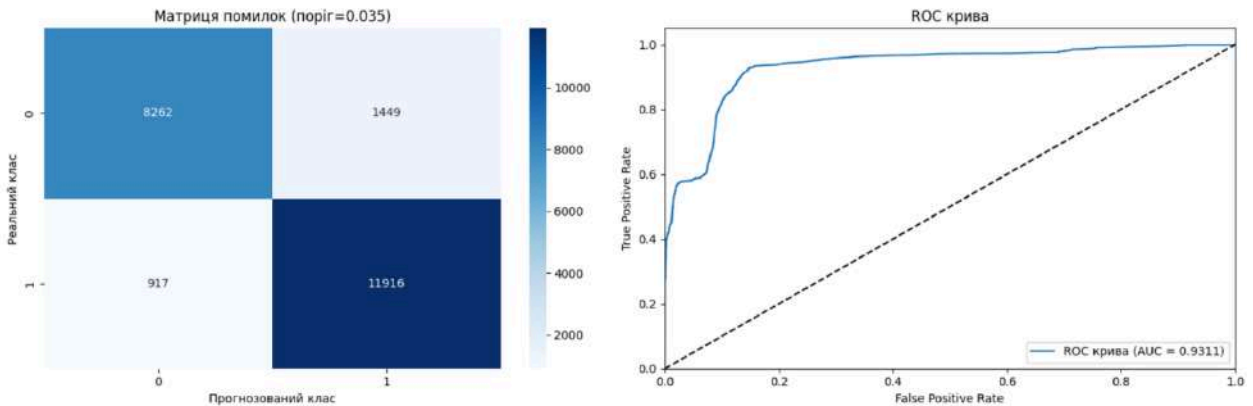


Рис. 10 – Результати моделі Deep CNN з найкращим порогом

Residual CNN

Residual CNN архітектура впроваджує механізм залишкових з'єднань (residual connections), що дозволяє ефективніше тренувати глибші мережі шляхом полегшення поширення градієнтів (Рис. 11).

Layer (type)	Output Shape	Param #	Connected to
input_layer_6 (InputLayer)	(None, 122, 1)	0	-
conv1d_21 (Conv1D)	(None, 122, 32)	128	input_layer_6[0]...
activation_8 (Activation)	(None, 122, 32)	0	conv1d_21[0][0]
conv1d_22 (Conv1D)	(None, 122, 32)	3,104	activation_8[0][...]
activation_9 (Activation)	(None, 122, 32)	0	conv1d_22[0][0]
max_pooling1d_10 (MaxPooling1D)	(None, 61, 32)	0	activation_9[0][...]
conv1d_24 (Conv1D)	(None, 61, 64)	6,208	max_pooling1d_10...
activation_10 (Activation)	(None, 61, 64)	0	conv1d_24[0][0]
conv1d_25 (Conv1D)	(None, 61, 64)	12,352	activation_10[0]...
conv1d_23 (Conv1D)	(None, 61, 64)	2,112	max_pooling1d_10...
add_2 (Add)	(None, 61, 64)	0	conv1d_25[0][0], conv1d_23[0][0]
activation_11 (Activation)	(None, 61, 64)	0	add_2[0][0]
max_pooling1d_11 (MaxPooling1D)	(None, 30, 64)	0	activation_11[0]...
global_average_poo... (GlobalAveragePool...)	(None, 64)	0	max_pooling1d_11...
dense_13 (Dense)	(None, 64)	4,160	global_average_p...
dropout_7 (Dropout)	(None, 64)	0	dense_13[0][0]
dense_14 (Dense)	(None, 1)	65	dropout_7[0][0]

Рис. 11 – Структура Residual CNN

Для цієї моделі найкращим виявився поріг 0.003 (Рис. 18). Вдалося досягти наступних результатів: precision – 89%, recall – 90%, F1 Score – 90%, ROC AUC – 92%.

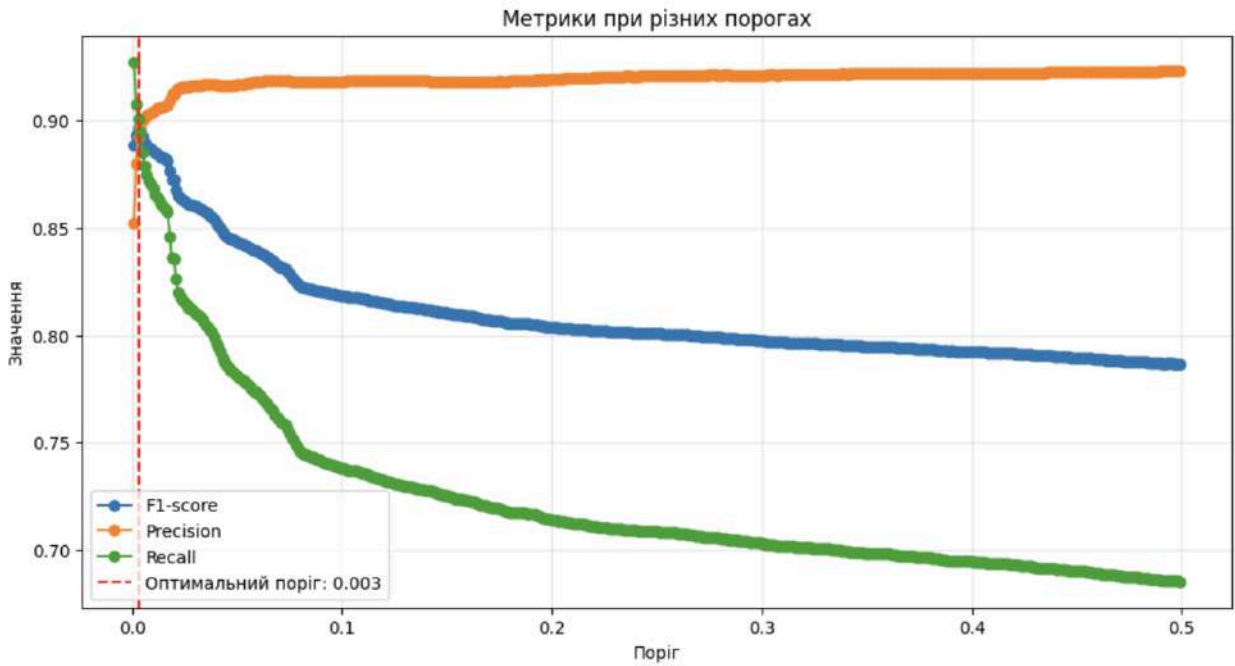


Рис. 12 – Результати Residual CNN при різних порогах

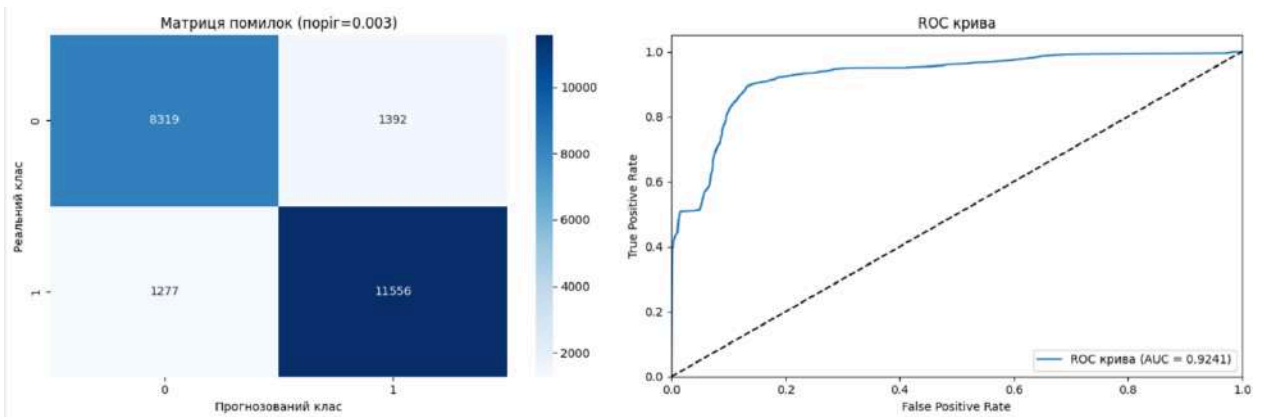


Рис. 13 – Результати Residual CNN за оптимального порогу

Порівняння результатів

Результати продуктивності кожної з моделей на тестових вибірках представлені в Таблиці 1, а також на Рис. 14 та Рис. 15.

Всі три моделі показали відмінний результат з F1 Score ~90% та ROC-AUC 92+%, що свідчить про хорошу здатність CNN архітектур виявляти мережеві аномалії.

Таблиця 1 – Порівняння результатів CNN моделей на тестовій вибірці

	optimal threshold	precision	recall	F1	ROC AUC
Базова CNN	0,004	88,7%	91,9%	90,2%	93,0%
Deep CNN	0,035	89,1%	92,9%	90,9%	93,1%
Residual CNN	0,003	89,2%	90,0%	89,6%	92,4%

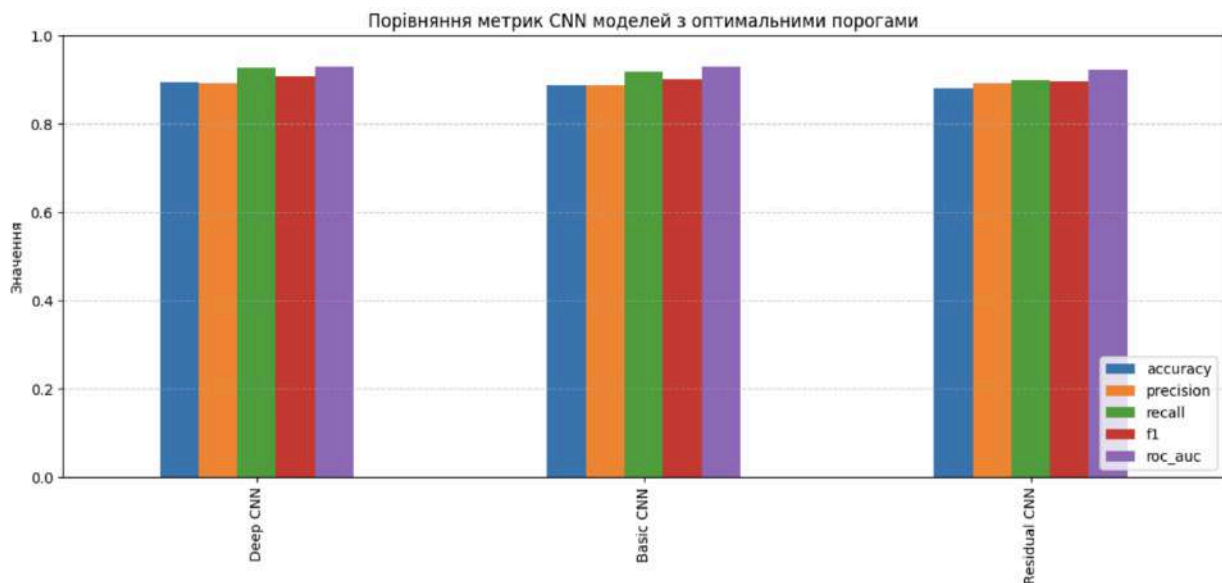


Рис. 14 – Порівняння продуктивності CNN моделей

Найкращою за F1-Score, ROC-AUC та Recall виявилась Deep CNN. Модель демонструє як найкращу загальну продуктивність прогнозів, так і найкращий баланс між хибно позитивними та хибно негативними прогнозами. Значення F1 Score 90,9% та ROC AUC 93,1% прийнято [9] вважати відмінним.

Порівнюючи значення recall на оптимальних порогах із стандартним значенням при порозі 0.5 (Рис. 16), підтверджує тезу про потребу в високій чутливості моделей для виявлення аномалій мережевого середовища. При оптимізації порогу відбулось покращення F1 Score в середньому на 40% – з 62-68% до 90-93%.

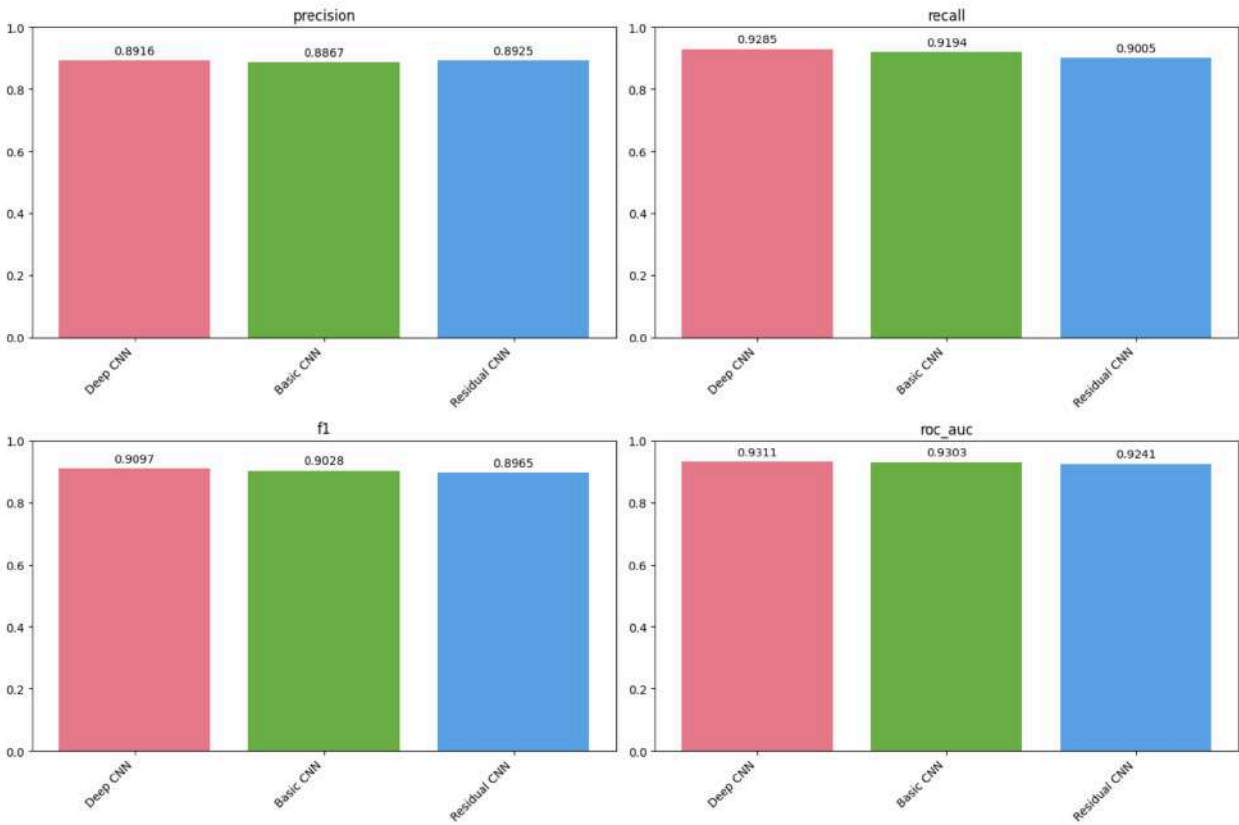


Рис. 15 – Порівняння продуктивності CNN моделей – 2

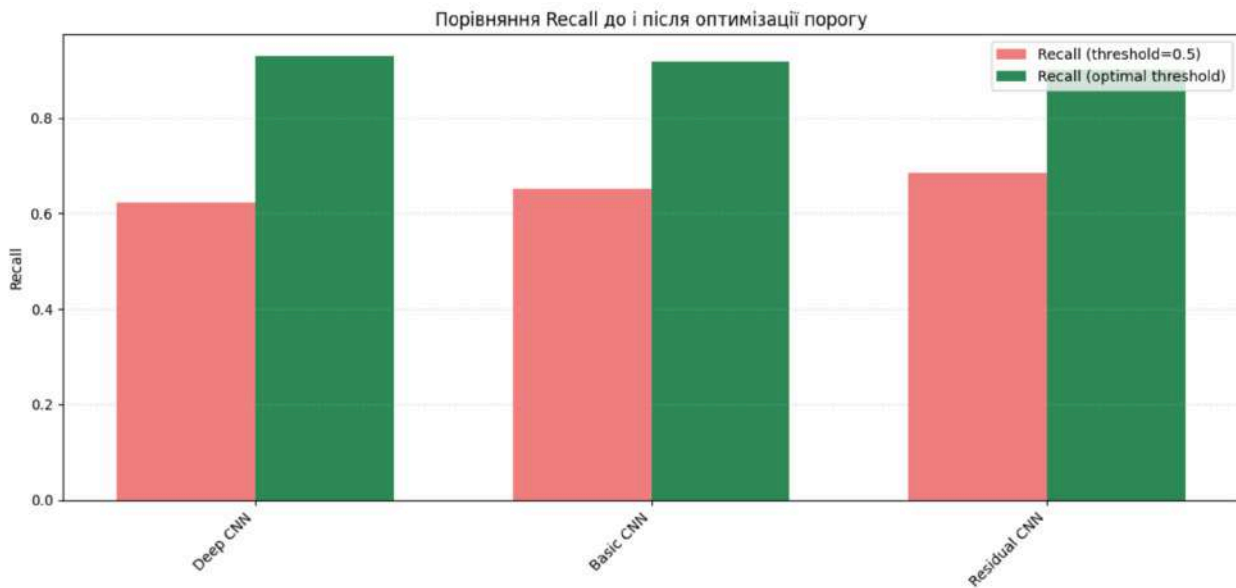


Рис. 16 – Порівняння recall при стандартному та оптимальних порогах класифікації

Загалом перевага Deep CNN над іншими моделями демонструє що глибинні моделі дійсно краще справляються із виявленням складних закономірностей у даних, а також закладає підґрунтя для подальших досліджень. Потенційно розробка складніших глибоких архітектур CNN може призвести до кращих результатів у виявленні аномалій.

3.1.5 Експериментальне дослідження LSTM архітектур

Базова LSTM

Базова LSTM архітектура представляє класичну одношарову реалізацію мережі довгої короткочасної пам'яті, спеціально спроектованої для ефективного навчання на часових послідовностях з довготривалими залежностями. (Рис. 17)

Layer (type)	Output Shape	Param #
lstm_3 (LSTM)	(None, 64)	47,872
dense_4 (Dense)	(None, 32)	2,080
dropout_2 (Dropout)	(None, 32)	0
dense_5 (Dense)	(None, 1)	33

Рис. 17 – Архітектура базової LSTM

Для LSTM моделей так само використовувався підхід з пошуком оптимального порогу класифікації, проте довелось зменшити діапазон пошуку значень в меншу сторону. Для LSTM найкращий результат дав поріг на рівні 0.00057 (Рис. 18): accuracy - 86,0%, precision - 88,0%, recall - 87,4%, F1 Score - 87,7%, ROC AUC - 89,7%.

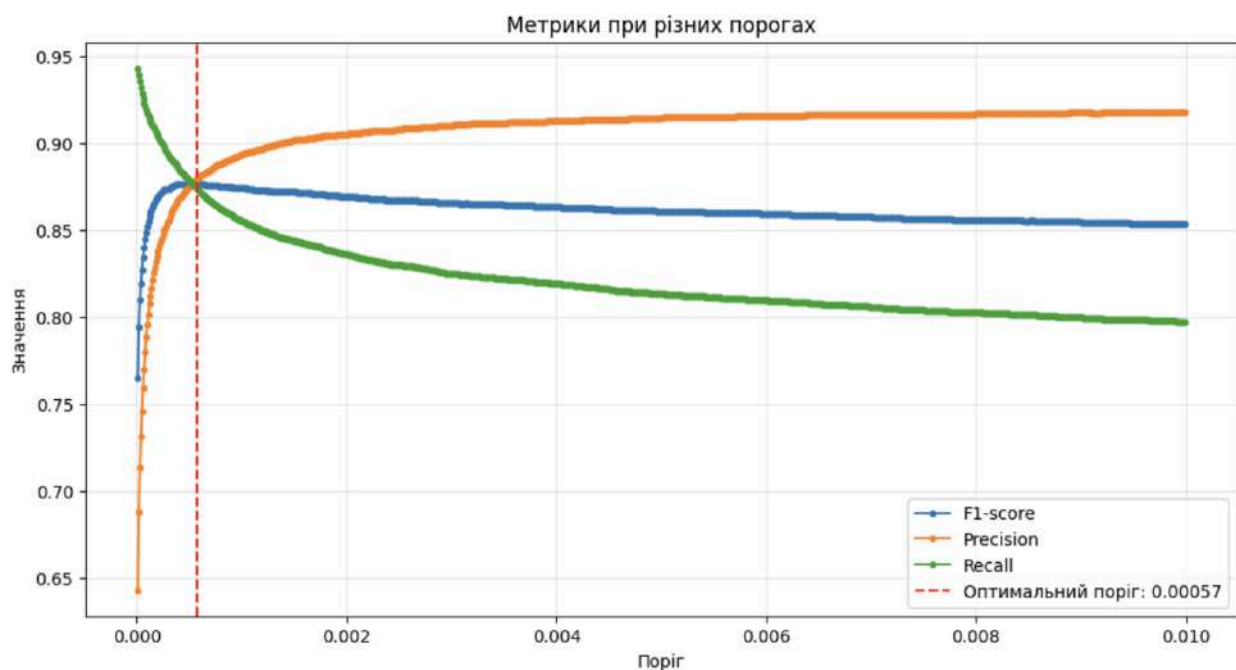


Рис. 18 – Результати базової LSTM для різних порогів класифікації

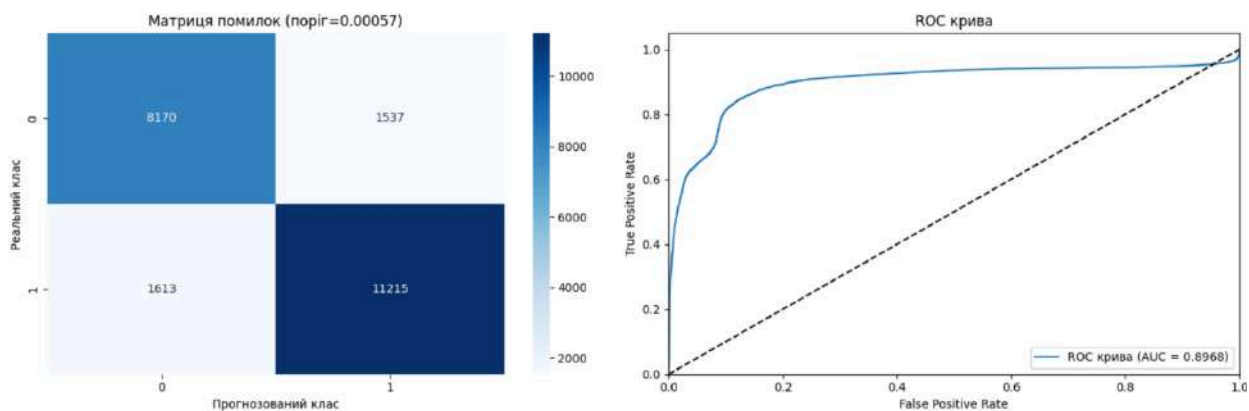


Рис. 19 – Результати базової LSTM з оптимальним порогом

Bidirectional LSTM

Двонаправлена LSTM архітектура розширює концепцію односпрямованої LSTM шляхом аналізу послідовності в обох напрямках, забезпечуючи більш комплексне розуміння контексту даних (Рис. 20).

Layer (type)	Output Shape	Param #
bidirectional_2 (Bidirectional)	(None, 10, 128)	95,744
bidirectional_3 (Bidirectional)	(None, 64)	41,216
dense_6 (Dense)	(None, 32)	2,080
dropout_3 (Dropout)	(None, 32)	0
dense_7 (Dense)	(None, 1)	33

Рис. 20 – Архітектура Bidirectional LSTM

Для Bidirectional LSTM найоптимальнішим виявився поріг класифікації на рівні 0.003 (Рис. 21): accuracy - 86,8%, precision - 88,3%, recall - 88,5%, F1 Score - 88,4%, ROC AUC - 92,4%.

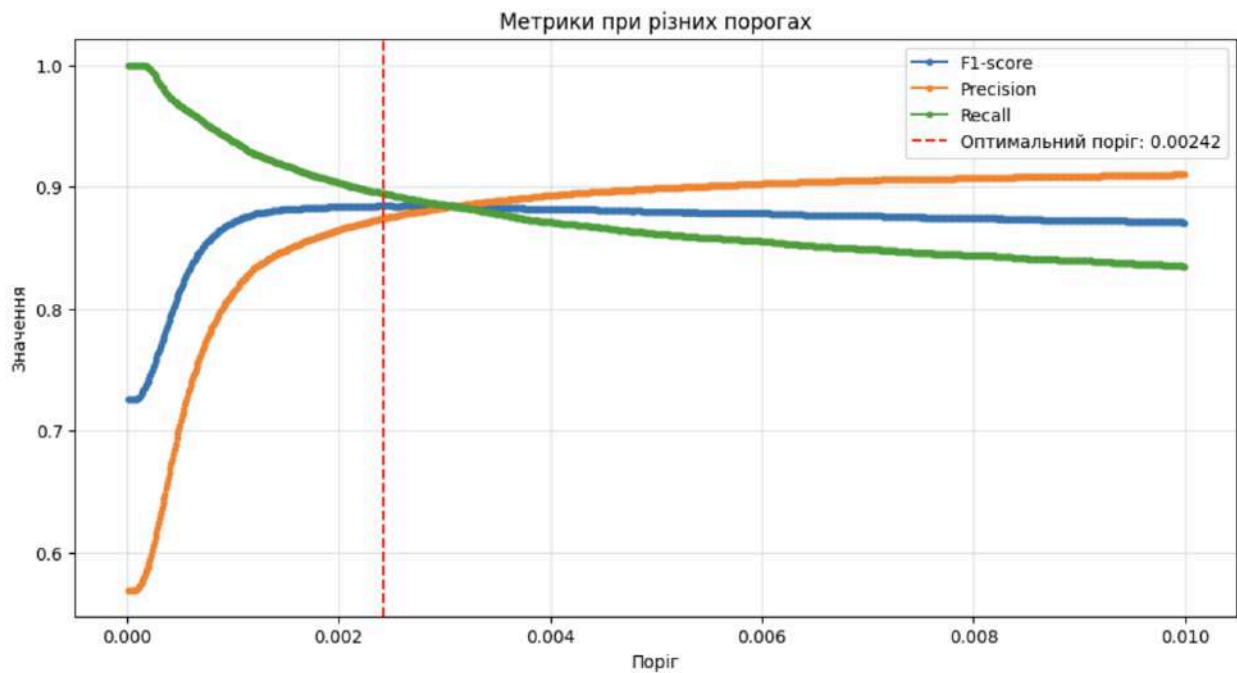


Рис. 21 – Результати Bidirectional LSTM для різних порогів

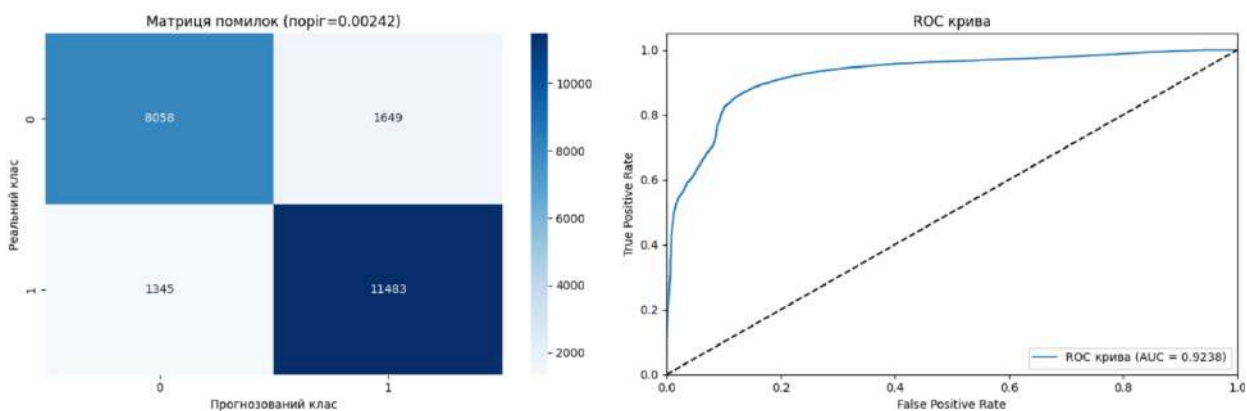


Рис. 22 – Результати Bidirectional LSTM для оптимального порогу класифікації

GRU

Класична архітектура GRU, що пропонує альтернативний підхід до обробки послідовних даних через використання модифікованих рекурентних блоків з меншою кількістю параметрів порівняно з LSTM. (Рис. 23). Також додано шар BatchNormalization для стабілізації розподілу активацій.

Layer (type)	Output Shape	Param #
gru (GRU)	(None, 10, 64)	36,096
gru_1 (GRU)	(None, 32)	9,408
batch_normalization (BatchNormalization)	(None, 32)	128
dense_8 (Dense)	(None, 32)	1,056
dropout_4 (Dropout)	(None, 32)	0
dense_9 (Dense)	(None, 1)	33

Рис. 23 – Архітектура GRU

Для GRU оптимальним виявився поріг класифікації 0.00031 (Рис. 24): accuracy - 86,2%, precision - 85,2%, recall - 91,7%, F1 Score - 88,2%, ROC AUC - 0,91%.

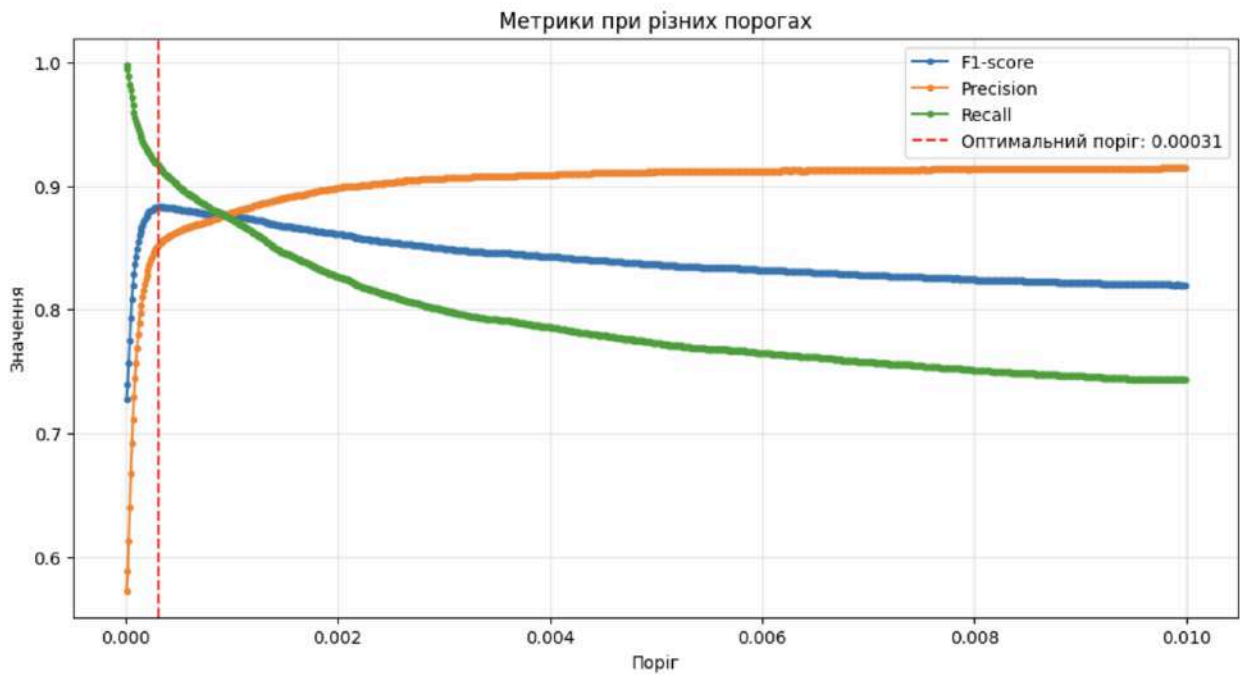


Рис. 24 – Результати GRU для різних порогів

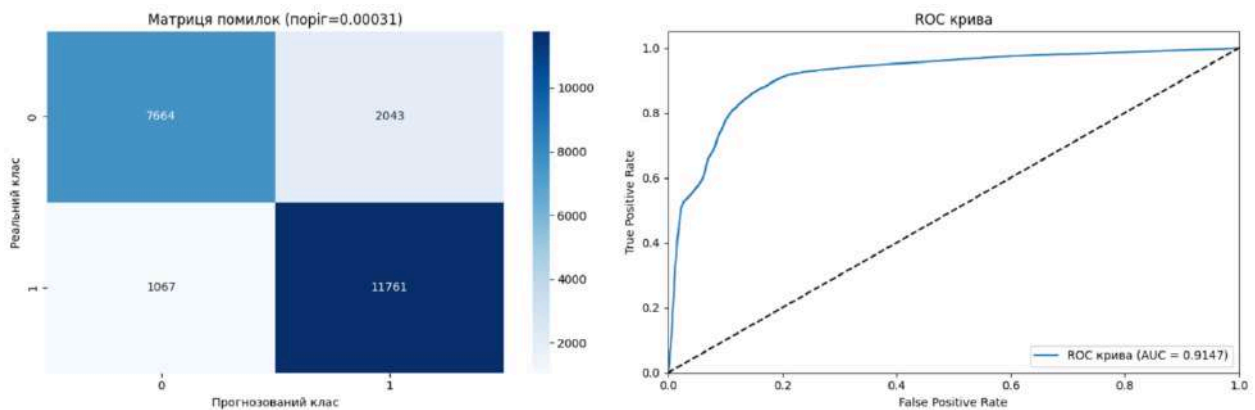


Рис. 25 – Результати GRU з оптимальним порогом класифікації

LSTM з механізмом уваги

LSTM з механізмом уваги (LSTM with Attention) представляє найскладнішу рекурентну архітектуру, яка доповнює двонаправлену LSTM спеціалізованим механізмом уваги, що обчислює ваги значущості для кожного часового кроку в послідовності, дозволяючи моделі "фокусуватися" на найбільш інформативних частинах трафіку (Рис. 26).

Layer (type)	Output Shape	Param #	Connected to
input_layer_6 (InputLayer)	(None, 10, 122)	0	-
bidirectional_5 (Bidirectional)	(None, 10, 128)	95,744	input_layer_6[0]...
dense_11 (Dense)	(None, 10, 1)	129	bidirectional_5[...]
flatten (Flatten)	(None, 10)	0	dense_11[0][0]
activation (Activation)	(None, 10)	0	flatten[0][0]
repeat_vector (RepeatVector)	(None, 128, 10)	0	activation[0][0]
permute (Permute)	(None, 10, 128)	0	repeat_vector[0]...
multiply (Multiply)	(None, 10, 128)	0	bidirectional_5[...] permute[0][0]
lambda (Lambda)	(None, 128)	0	multiply[0][0]
dense_12 (Dense)	(None, 32)	4,128	lambda[0][0]
dropout_5 (Dropout)	(None, 32)	0	dense_12[0][0]
dense_13 (Dense)	(None, 1)	33	dropout_5[0][0]

Рис. 26 – Архітектура LSTM з Attention

Для LSTM з механізмом уваги оптимальним виявився поріг класифікації близько 0.004 (Рис. 27): accuracy - 85,1%, precision - 86,3%, recall - 87,8%, F1 Score - 87,0%, ROC AUC - 89,4%.

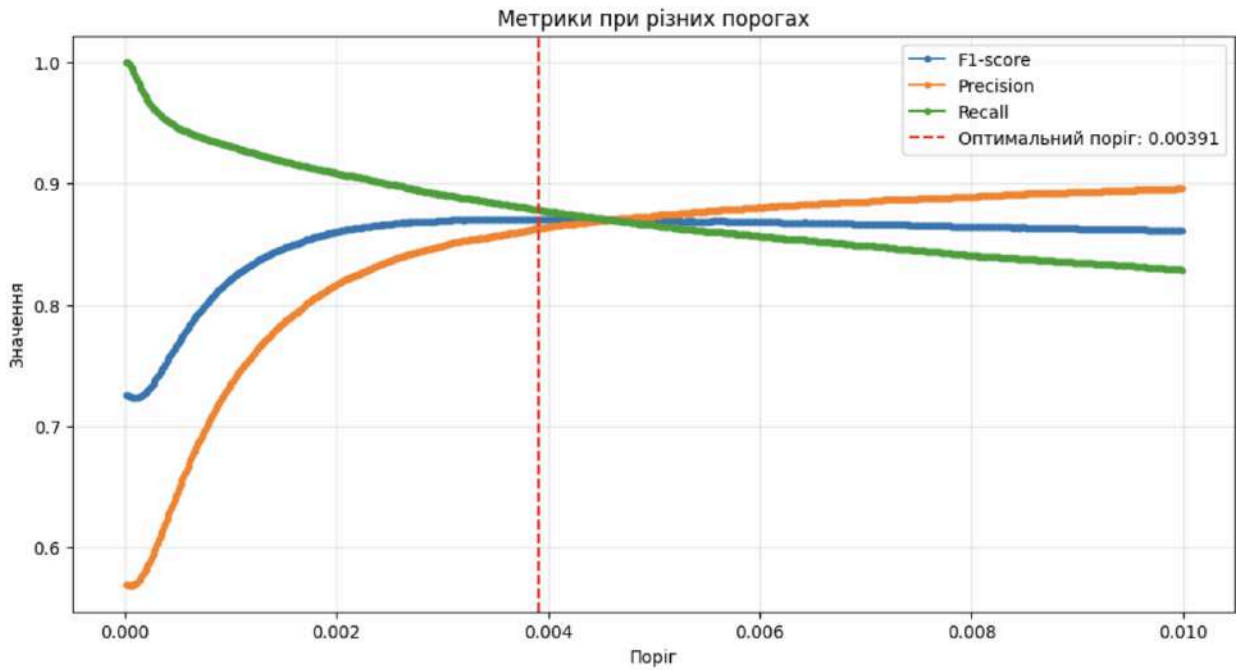


Рис. 27 – Результати LSTM з Attention для різних порогів

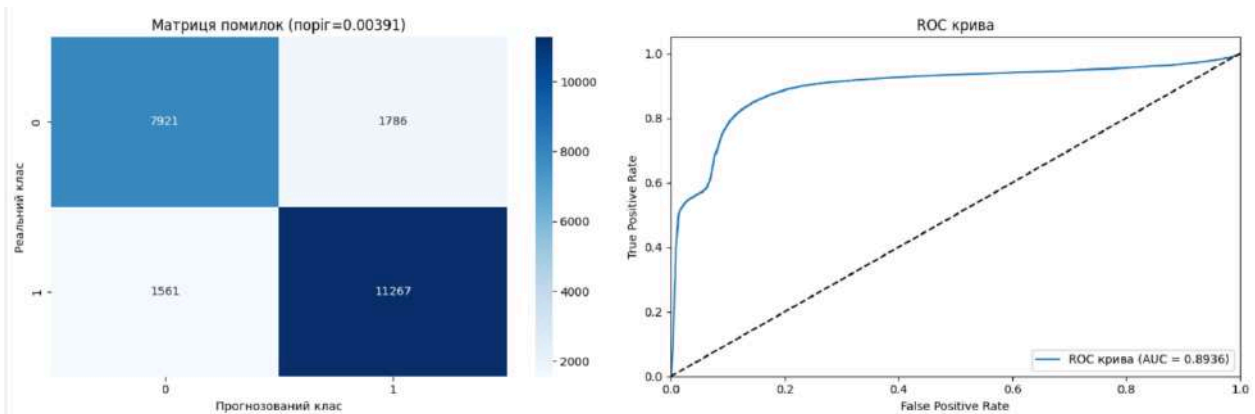


Рис. 28 – Результати LSTM з Attention за оптимального порогу

Порівняння LSTM архітектур

Результати продуктивності кожної з LSTM моделей на тестових вибірках представлені в Таблиці 2, а також на Рис. 29 та Рис. 30. Усі протестовані архітектури LSTM продемонстрували високу здатність виявляти аномалії в мережевому трафіку, із значеннями F1-міри в діапазоні 87,0-88,4%, та ROC AUC в діапазоні 89,6-92,4%б що підтверджує доцільність застосування рекурентних нейронних мереж для цієї задачі.

Bidirectional LSTM показала найкращі результати за F1 та ROC AUC серед усіх LSTM-моделей, що підтверджує важливість врахування контексту в обох напрямках послідовності для ефективного виявлення аномалій. Проте, GRU забезпечує найвищу повноту (recall) та має більшу обчислювальну ефективність через меншу кількість параметрів порівняно з LSTM.

Таблиця 2 – Порівняння результатів LSTM моделей

	threshold	precision	recall	F1	ROC AUC
Basic LSTM	0.0006	87,9%	87,4%	87,7%	89,6%
Bidirectional LSTM	0.0024	87,4%	89,5%	88,4%	92,4%
GRU	0.0003	85,1%	91,7%	88,3%	91,4%
LSTM with Attention	0.0039	86,3%	87,8%	87,0%	89,4%

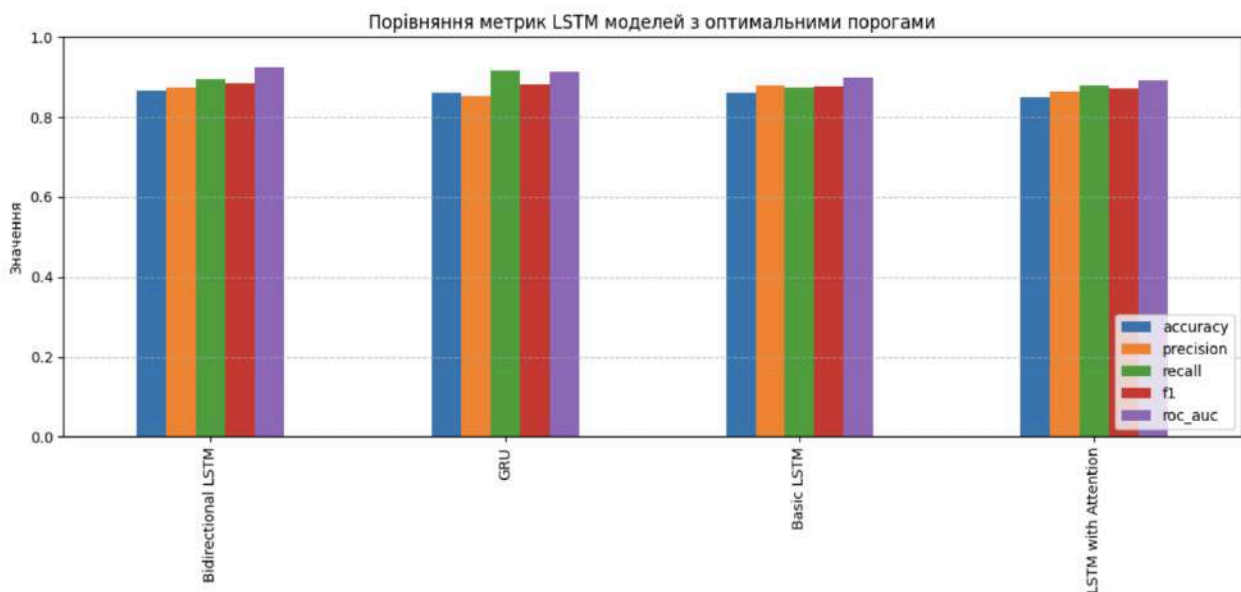


Рис. 29 – Порівняння результатів моделей LSTM з оптимальними порогоми

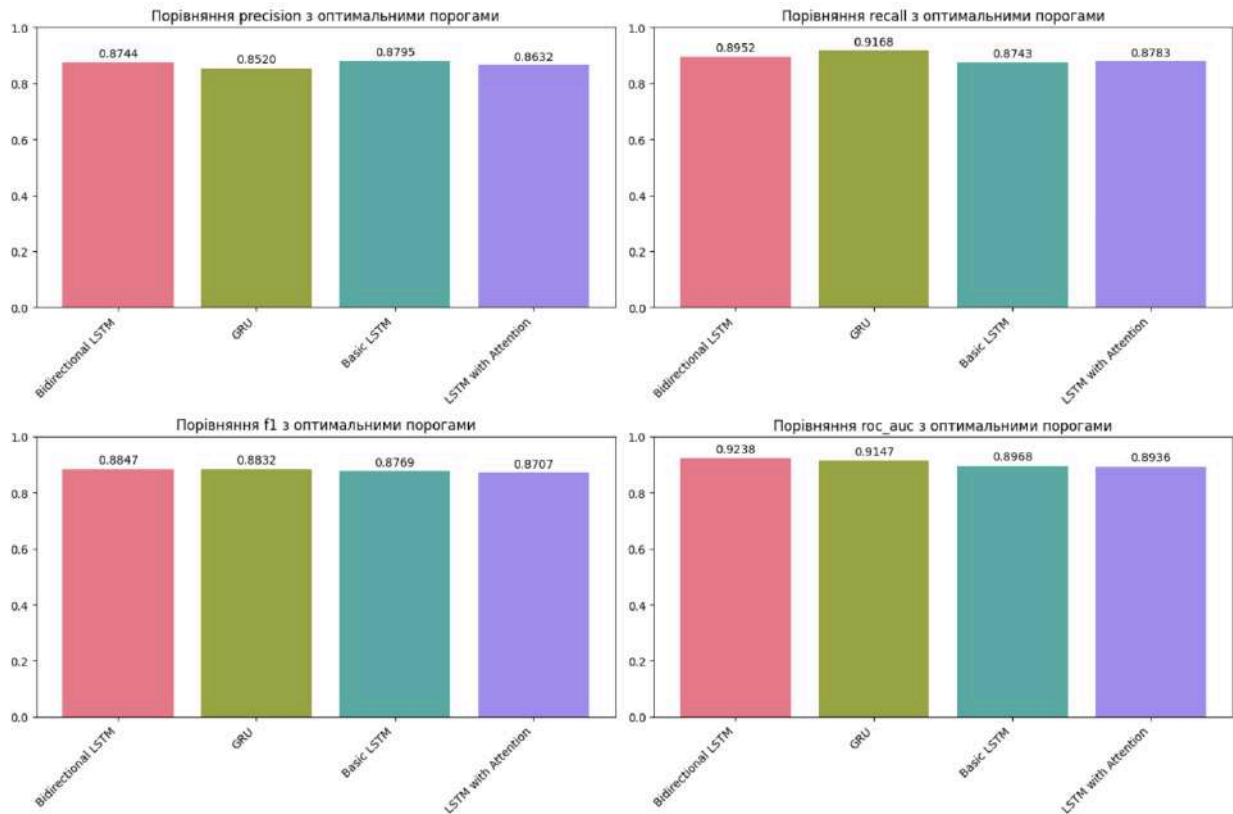


Рис. 30 – Порівняння результатів моделей LSTM з оптимальними порогоми

3.1.6 Експериментальне дослідження архітектур автоенкодерів

Dense Autoencoder

Дана архітектура представляє базовий повнов'язний автоенкодер, що складається з енкодера та декодера (Рис. 31). Енкодер складається з послідовності шарів Dense (128 → 64 → 32) що зменшують розмірність для отримання компактного представлення, між ними — нормалізація (Batch Normalization). Декодер має зворотню структуру (32 → 64 → 128 → 122), що відновлюють початкову розмірність даних.

Знаходження оптимального порогу класифікації для моделі автоенкодера відбувалось на основі значення помилки реконструкції. Найоптимальнішим виявився поріг на рівні 0.079 (Рис. 32): precision - 88,5%, recall - 95,1%, F1 Score - 91,7%, ROC AUC - 93,6%.

Layer (type)	Output Shape	Param #
input_layer_4 (InputLayer)	(None, 122)	0
dense_28 (Dense)	(None, 128)	15,744
batch_normalization_10 (BatchNormalization)	(None, 128)	512
dense_29 (Dense)	(None, 64)	8,256
dense_30 (Dense)	(None, 32)	2,080
dense_31 (Dense)	(None, 64)	2,112
batch_normalization_11 (BatchNormalization)	(None, 64)	256
dense_32 (Dense)	(None, 128)	8,320
dense_33 (Dense)	(None, 122)	15,738

Рис. 31 – Архітектура Dense Autoencoder

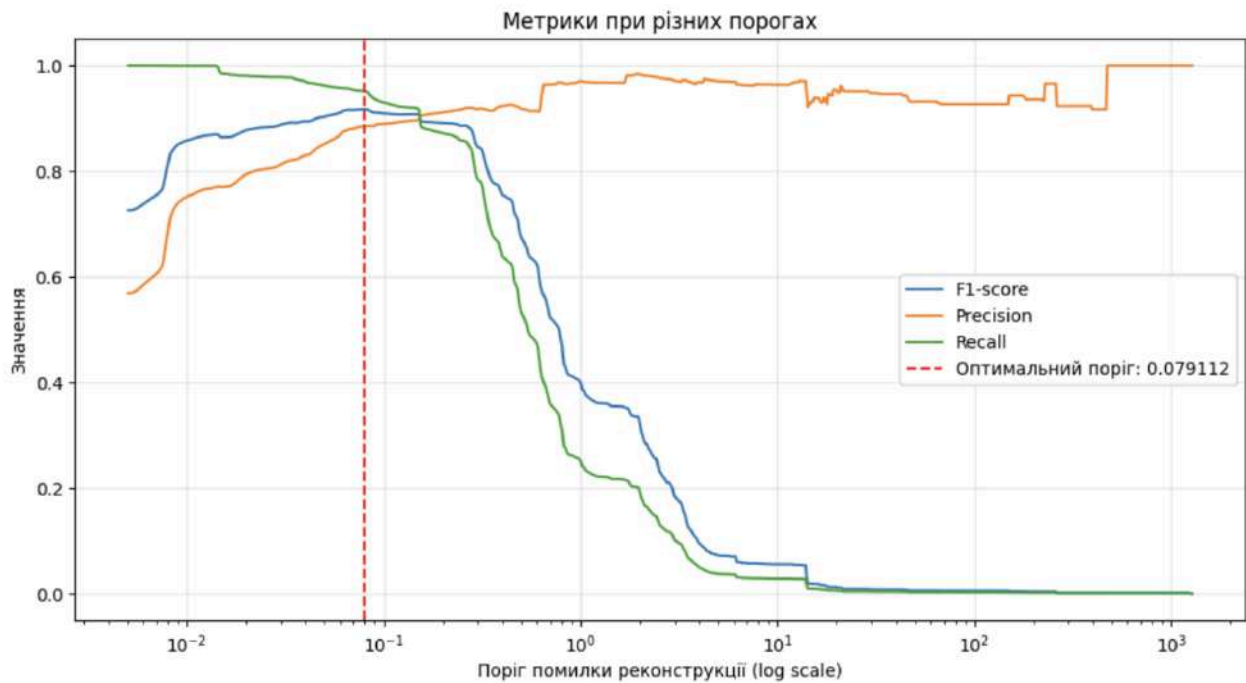


Рис. 32 – Результати Dense Autoencoder для різних порогів

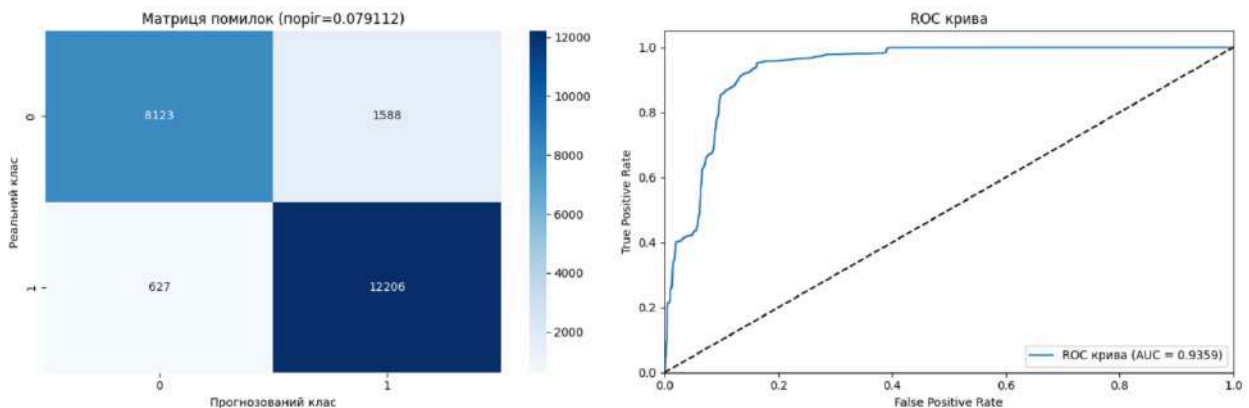


Рис. 33 – Результат Dense Autoencoder з оптимальним порогом

CNN Autoencoder

CNN автоенкодер використовує згорткові шари замість повнов'язних, але так само має симетричну структуру з енкодером і декодером (Рис. 34). Енкодер: виділяє просторові ознаки (фільтри 32 → 64), стабілізує навчання

(BatchNormalization) та зменшує розмірність, зберігаючи лише важливі ознаки в шарі MaxPooling (124 → 62 → 31). Декодер: відновлює розмірність (31 → 62 → 124) та повертає початкові ознаки.

Особливості моделі:

- *Різні розміри фільтрів*: Архітектура поєднує фільтри різного розміру (3×1 та 5×1), що дозволяє моделі захоплювати патерни різного масштабу в мережевому трафіку.
- *Пакетна нормалізація (BatchNormalization)*: Після кожного згорткового шару застосовується пакетна нормалізація, що стабілізує навчання, прискорює збіжність та зменшує ризик перенавчання.
- *Глибока, але вузька структура*: Архітектура має декілька шарів (глибина), але кожен шар має відносно небагато фільтрів (32-64), що забезпечує баланс між репрезентативною потужністю та ефективністю обчислень.

Layer (type)	Output Shape	Param #
input_layer_10 (InputLayer)	(None, 122, 1)	0
zero_padding1d_1 (ZeroPadding1D)	(None, 124, 1)	0
conv1d_19 (Conv1D)	(None, 124, 32)	128
batch_normalization_20 (BatchNormalization)	(None, 124, 32)	128
max_pooling1d_7 (MaxPooling1D)	(None, 62, 32)	0
conv1d_20 (Conv1D)	(None, 62, 64)	6,208
batch_normalization_21 (BatchNormalization)	(None, 62, 64)	256
max_pooling1d_8 (MaxPooling1D)	(None, 31, 64)	0
conv1d_21 (Conv1D)	(None, 31, 32)	6,176
conv1d_22 (Conv1D)	(None, 31, 64)	6,208
up_sampling1d_7 (UpSampling1D)	(None, 62, 64)	0
conv1d_23 (Conv1D)	(None, 62, 32)	6,176
up_sampling1d_8 (UpSampling1D)	(None, 124, 32)	0
conv1d_24 (Conv1D)	(None, 124, 1)	97
cropping1d (Cropping1D)	(None, 122, 1)	0

Рис. 34 – Архітектура CNN Autoencoder

Для CNN автоенкодера оптимальним виявився поріг класифікації на рівні 0.0055 (Рис. 35). Вдалось досягти precision - 85,9%, recall - 97%, F1 Score - 91,1%, ROC AUC - 95,3%.

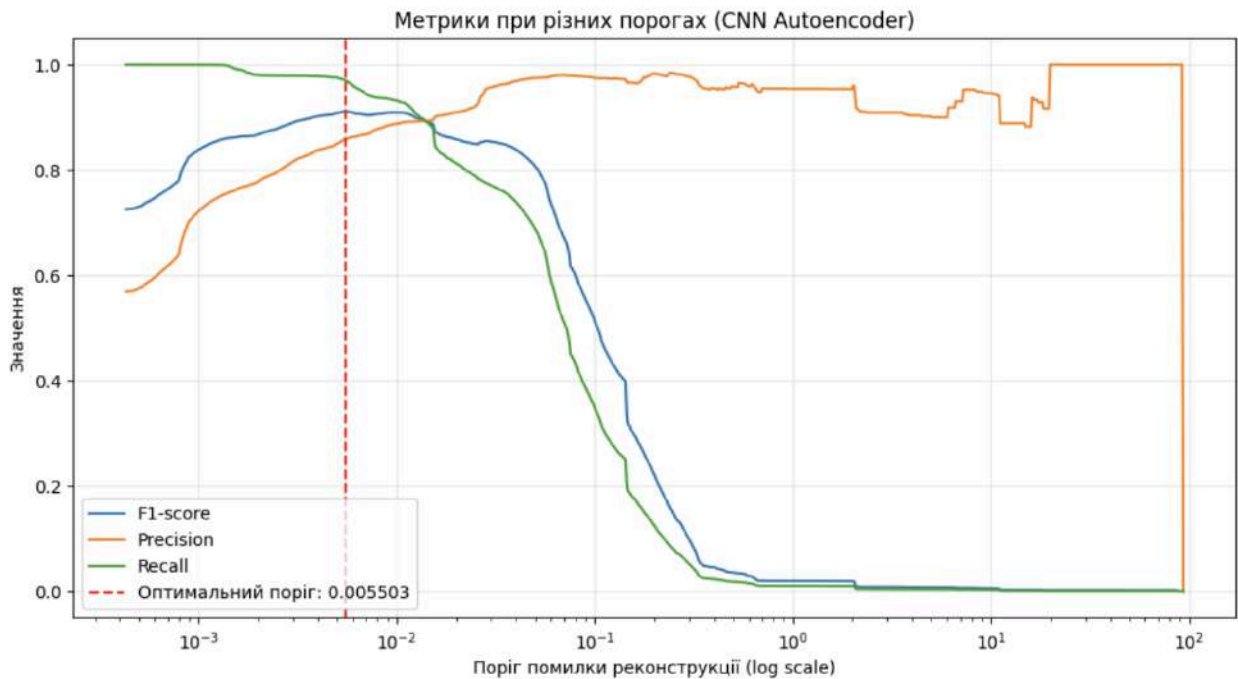


Рис. 35 – Результати CNN Autoencoder для різних порогів

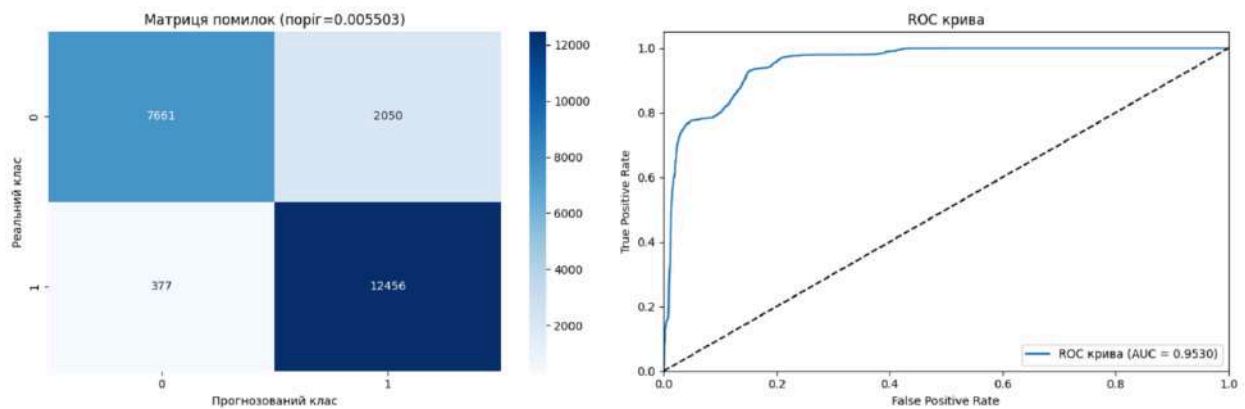


Рис. 36 – Результати CNN Autoencoder для оптимального порогу

GRU Autoencoder

В якості LSTM автоенкодера, що використовує LSTM шари замість повнов'язних, було вирішено використати GRU через його вищу обчислювальну ефективність. Модель є стандартним GRU Autoencoder та складається з GRU шарів, які стискають вхідну послідовність у внутрішнє представлення, та декодера, який розгортає це представлення назад у послідовність (Рис. 37).

Layer (type)	Output Shape	Param #
gru_4 (GRU)	(None, 10, 64)	36,096
gru_5 (GRU)	(None, 32)	9,408
repeat_vector_3 (RepeatVector)	(None, 10, 32)	0
gru_6 (GRU)	(None, 10, 32)	6,336
gru_7 (GRU)	(None, 10, 64)	18,816
time_distributed_3 (TimeDistributed)	(None, 10, 122)	7,930

Рис. 37 – Архітектура моделі GRU Autoencoder

Для GRU автоенкодера найоптимальнішим виявилось порогове значення 0.349 (Рис. 38), за якого вдалось досягти вражаючих, майже ідеальних результатів: precision - 97,3%, recall - 99,8%, F1 Score - 98,5%, ROC AUC - 99,1%.

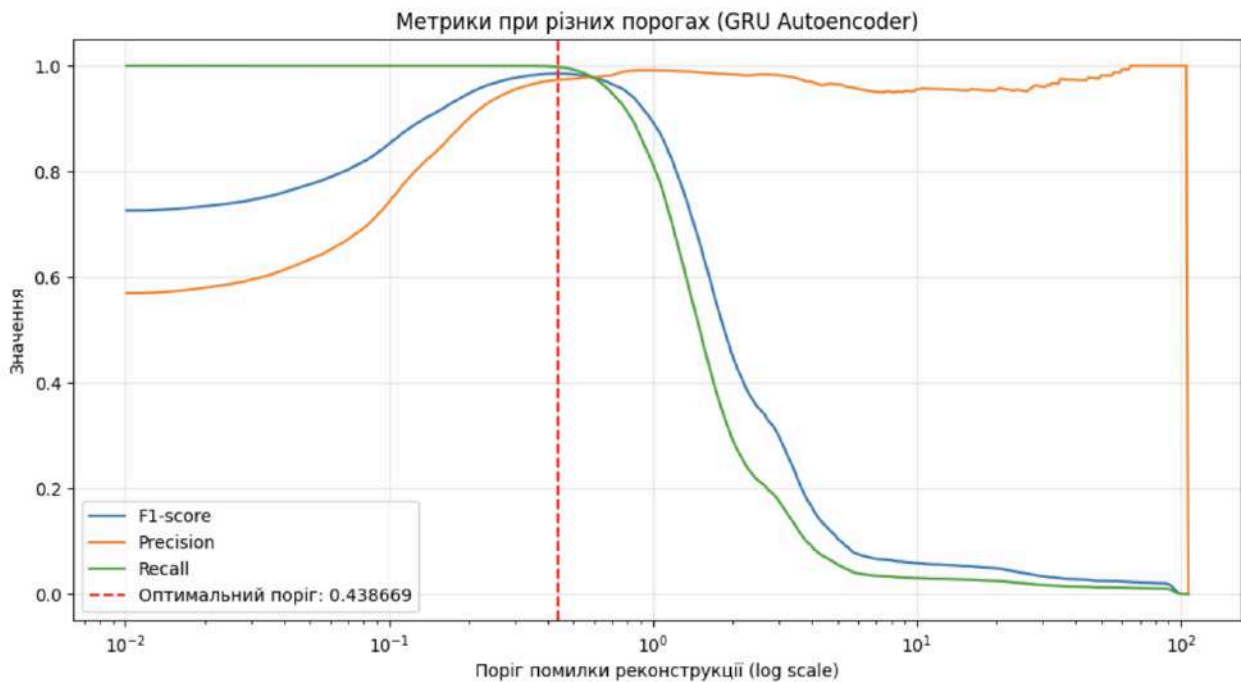


Рис. 38 – Результати GRU Autoencoder для різних порогових значень

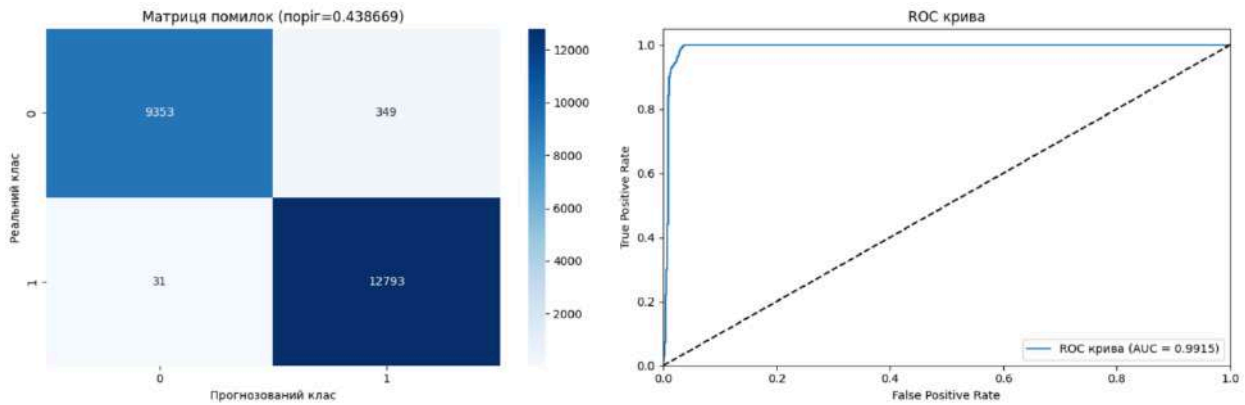


Рис. 39 – Результати GRU Autoencoder для оптимального порогового значення

Порівняння автоенкодер моделей

Порівняння всіх автоенкодерів наведено в Таблиці 3.

Таблиця 3 – Порівняння результатів автоенкодер моделей

	threshold	precision	recall	F1	ROC AUC
Dense Autoencoder	0.1190	88,5%	95,3%	91,8%	93,4%
CNN Autoencoder	0.0055	85,9%	97,0%	91,1%	95,3%
GRU Autoencoder	0.4386	97,3%	99,8%	98,5%	99,1%

Всі автоенкодери продемонстрували хороший F1 Score та дуже високі значення recall, що свідчить про їхню відмінну здатність виявляти аномалії в мережевих даних. Найкращий результат показав GRU Autoencoder, досягнувши рекордних для всіх моделей, та майже ідеальних показників ROC AUC 99,1% та recall 99,8%. Порівняно гірше, проте все ще на дуже високому рівні із ROC AUC 95,5% впорався CNN Autoencoder,

3.1.7 Аналіз та порівняння ефективності різних архітектур нейронних мереж на наборі NSL-KDD

У Таблиці 4 представлено результати дослідження архітектур нейронних мереж на наборі NSL-KDD.

Таблиця 4 – Порівняння результатів CNN моделей на тестовій вибірці

	optimal threshold	precision	recall	F1	ROC AUC
Базова CNN	0.004	88.7%	91.9%	90.2%	93.0%
Deep CNN	0.035	89.1%	92.9%	90.9%	93.1%
Residual CNN	0.003	89.2%	90.0%	89.6%	92.4%
Basic LSTM	0.0006	87.9%	87.4%	87.7%	89.6%
Bidirectional LSTM	0.0024	87.4%	89.5%	88.4%	92.4%
GRU	0.0003	85.1%	91.7%	88.3%	91.4%
LSTM with Attention	0.0039	86.3%	87.8%	87.0%	89.4%
Dense Autoencoder	0.1190	88.5%	95.3%	91.8%	93.4%
CNN Autoencoder	0.0055	85.9%	97.0%	91.1%	95.3%
GRU Autoencoder	0.4386	97.3%	99.8%	98.5%	99.1%

Найкращий *F1-score* демонструє *GRU Autoencoder* (98.5%), значно випереджаючи інші моделі, такі як *Dense Autoencoder* (91,8%), *CNN Autoencoder* (91,1%) та *Deep CNN* (90,9%).

Найвищий *Recall* (здатність виявляти аномалії) показав *GRU Autoencoder* (99,8%), за ним ідуть *CNN Autoencoder* (97,0%) та *Dense Autoencoder* (95,3%). Ці показники значно перевищують результати інших архітектур.

Найкраща Precision (точність визначення аномалій) спостерігається у *GRU Autoencoder* (97,3%). За ним слідують *Residual CNN* (89,2%) та *Deep CNN* (89,1%).

Найвищий ROC AUC зафіксовано у *GRU Autoencoder* (99,1%), що свідчить про найкращу загальну дискримінаційну здатність цієї моделі при різних порогових значеннях, перевершуючи *CNN Autoencoder* (95,3%) та *Dense Autoencoder* (93,4%).

Оптимальні пороги класифікації суттєво відрізняються між різними архітектурами, що підтверджує важливість адаптивного підходу до вибору порогу для кожної конкретної моделі. Наприклад, для стандартної *GRU* оптимальний поріг становить лише 0,0003, для *Dense Autoencoder* він дорівнює 0,1190, тоді як для *GRU Autoencoder* цей показник є найвищим і становить 0,4386.

CNN архітектури продемонстрували високу ефективність у виявленні аномалій завдяки здатності виявляти локальні патерни в даних. *Deep CNN* виявилася найефективнішою *CNN-архітектурою* з F1-score 90,9%, що підтверджує гіпотезу про те, що глибші моделі можуть виявляти більш складні взаємозв'язки в даних. Цікаво, що додавання залишкових з'єднань у *Residual CNN* не привело до очікуваного покращення результатів, що може вказувати на відсутність проблеми градієнтного загасання для даного набору даних. *CNN-архітектури* демонструють високу precision (наприклад, *Residual CNN* - 89,2%, *Deep CNN* - 89,1%), що робить їх цінними у випадках, коли важливо мінімізувати хибні спрацьовування. Однак, за ключовими показниками, такими як F1-score, recall, а також precision та ROC AUC, вони поступаються передовим автоенкодерам, зокрема *GRU Autoencoder*.

LSTM архітектури показали нижчу загальну ефективність порівняно з *CNN* та автоенкодерами, з F1-score в діапазоні 87,0% (*LSTM with Attention*) – 88,4% (*Bidirectional LSTM*) та 88,3% для стандартної *GRU*. Це може свідчити про те, що хоча часові залежності і присутні, їх моделювання стандартними рекурентними шарами не дає значних переваг над згортковими або автоенкодерними підходами для даного набору даних. Серед *LSTM-архітектур*

Bidirectional LSTM показала найкращий результат (F1-score 88,4%), що вказує на користь врахування контексту в обох напрямках послідовності. Дещо несподівано, додавання механізму уваги в LSTM with Attention не привело до покращення результатів, що потребує подальшого дослідження. Цікаво, що стандартна GRU показала одну з найнижчих precision (85,1%), але досить високий recall (91,7%), що свідчить про її схильність класифікувати більше зразків як аномалії, іноді жертвуючи точністю. Це суттєво контрастує з GRU Autoencoder, який демонструє виняткову точність (97,3%) та recall (99,8%).

Автоенкодера продемонстрували найвищу загальну ефективність серед усіх досліджених архітектур, причому *GRU Autoencoder* встановив новий стандарт продуктивності. GRU Autoencoder показав найкращий F1-score (98,5%), найвищий recall (99,8%), найвищу precision (97,3%) та найвищий ROC AUC (99,1%). За ним слідує Dense Autoencoder з F1-score 91,8% та CNN Autoencoder з F1-score 91,1%, recall 97,0% та ROC AUC 95,3%. Особливо важливим є те, що автоенкодера досягають таких результатів, навчаючись лише на нормальних даних, що робить їх особливо ефективними для виявлення невідомих типів аномалій. Винятково висока здатність GRU Autoencoder виявляти аномалії (найвищий recall) при одночасно найвищій точності (precision) робить його найбільш підходящою моделлю для систем безпеки, де пропуск атаки є критичним, а мінімізація хибних спрацьовувань – бажаною. Суттєво вищий поріг класифікації для GRU Autoencoder (0,4386), а також для Dense Autoencoder (0,1190) порівняно з більшістю інших моделей, вказує на інший характер розподілу помилок реконструкції. Це потенційно робить ці моделі автоенкодерів більш стабільними до незначних варіацій у даних та здатними чіткіше відокремлювати аномалії.

3.2 Дослідження узагальнення та трансферності моделей нейронних мереж

Трансферність та адаптивність моделей між середовищами є важливою для розробки production-level систем виявлення аномалій що можна було б легко

адаптувати під різні середовища без втрати продуктивності. Крім того, з часом в межах навіть одного середовища має місце концептуальний дрейф, і система має вміти адаптуватися до оновлених характеристик середовища.

Аби дослідити трансферність моделей нейронних мереж, в даній роботі пропонується проведення експериментів з новим набором даних на уже навчених моделях. Такий підхід імітує зміну мережевого середовища та дає можливість оцінити адаптивність моделей. Також пропонується дослідження ефективності донавчання (fine-tuning) моделі – наскільки змінюється якість прогнозів моделі при донавчанні на нових мережевих даних, та на якому їх обсязі.

Для експерименту були обрані ті моделі з кожної архітектури, що показали найкращі F1 Score та ROC AUC при тестуванні на наборі NSL-KDD - Deep CNN (F1 - 90,9%, ROC AUC - 93,1%), Bidirectional LSTM (F1 - 88,4%, ROC AUC - 92,4%), CNN Autoencoder (F1 - 91,1%, ROC AUC - 95,3%) та GRU Autoencoder (F1 - 98,5%, ROC AUC - 99,1%).

3.2.1 Вибір та підготовка набору даних

Для цього експерименту був обраний набір UNSW-NB15, що був створений у 2015 році в Лабораторії кібербезпеки Австралійського центру кібербезпеки (ACCS) при Університеті Нового Південного Уельсу. На відміну від NSL-KDD, який базується на даних 1999 року, UNSW-NB15 містить сучасніші типи мережевого трафіку та атак, що робить його більш репрезентативним для сучасних мережевих середовищ [67]. Таким чином сценарій тестування моделей, натренованих на наборі NSL-KDD є схожим на реальні сценарії трансферу між мережевими середовищами чи концептуального дрейфу.

UNSW-NB15 має наступну структуру:

- 175,341 записів у тренувальній вибірці.
- 82,332 записів у тестовій вибірці.

- 47 ознак (42 ознаки + 2 мітки + 3 ідентифікатори). Серед ознак базові характеристики з'єднань, ознаки потоку, часові ознаки та дані про з'єднання та вміст.
- Нормальний трафік та 9 різних типів атак: Fuzzers, analysis, backdoors, DoS, exploits, generic, reconnaissance, shellcode, worms.

UNSW-NB15 має суттєві відмінності від NSL-KDD:

- Більша кількість ознак (47 проти 41).
- Розширений набір сучасних атак (9 типів проти 4 категорій у NSL-KDD) (Рис. 40).
- Інше співвідношення між нормальним і аномальним трафіком (Рис. 41).

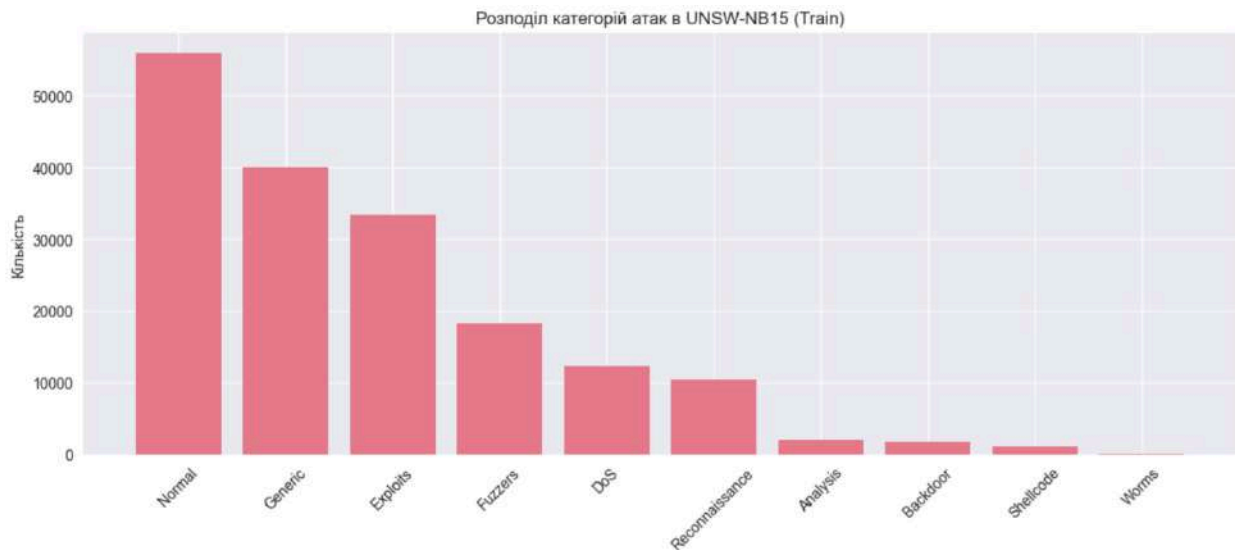


Рис. 40 – Розподіл категорій атак в UNSW-NB15

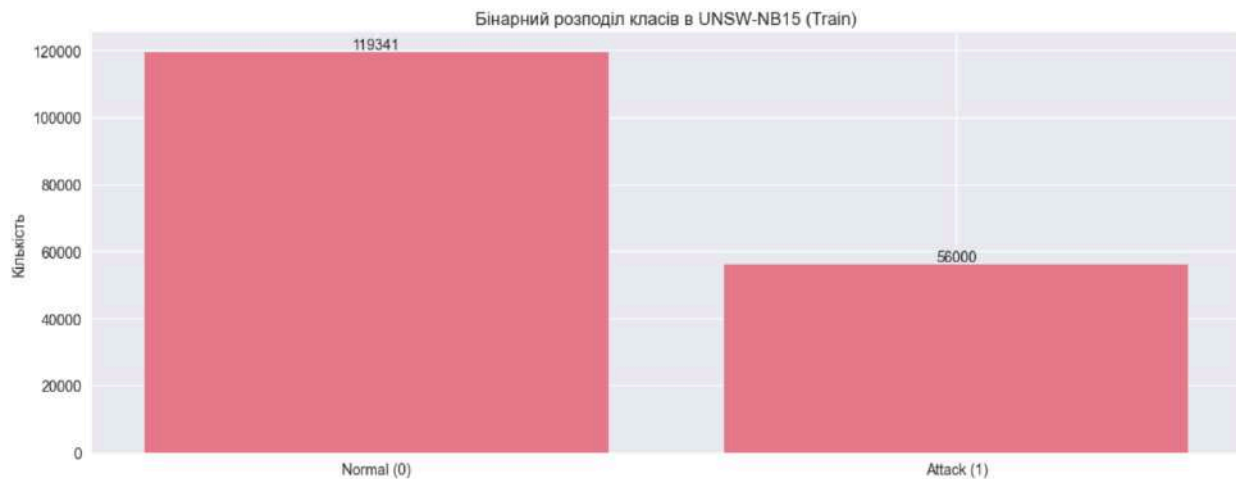


Рис. 41 – Розподіл під нормальним та аномальним трафіком в UNSW-NB15

- Різні набори категоріальних ознак: у UNSW-NB15 це proto, service та state, тоді як у NSL-KDD - protocol_type, service та flag.
- Більша різноманітність значень категоріальних ознак, наприклад, proto має 133 унікальних значення, тоді як protocol_type в NSL-KDD має лише 3.

Для забезпечення сумісності з моделями, навченими на NSL-KDD, було проведено ряд трансформацій даних UNSW-NB15: обробка категоріальних ознак шляхом застосування one-hot encoding, нормалізація числових ознак із застосуванням StandardScaler, бінаризація цільової змінної (0 – нормальний трафік, 1 – аномалія), а також підготовка даних для різних архітектур.

3.2.2 Експеримент з прямим перенесенням моделей без донавчання

Перший експеримент був спрямований на дослідження прямої трансферності моделей – тобто, наскільки ефективно моделі, навчені на NSL-KDD, можуть виявляти аномалії в UNSW-NB15 без будь-якого додаткового навчання. Це імітує сценарій, коли модель, навчена в одному мережевому середовищі, встановлюється в іншому середовищі без адаптації.

Методологія експерименту

Для кожної з чотирьох обраних моделей (Deep CNN, Bidirectional LSTM, CNN Autoencoder, GRU Autoencoder) було проведено наступні кроки:

1. Завантаження попередньо навченої моделі з NSL-KDD.
2. Створення відповідного адаптаційного шару для роботи з даними UNSW-NB15.
3. Застосування адаптованої моделі до тестових даних UNSW-NB15 без будь-якого донавчання.
4. Оцінка ефективності за метриками: F1 Score, ROC AUC, Precision, Recall
5. Для автоенкодерів: обчислення помилок реконструкції та визначення оптимального порогу класифікації.

Ключовою проблемою при перенесенні моделей між NSL-KDD (122 ознаки) та UNSW-NB15 (183 ознаки) *стала різна розмірність вхідних даних*. Один із можливих підходів полягав у приведенні ознакових просторів двох датасетів до єдиної структури шляхом вибору лише спільних ознак або їх підмножини. Це дозволило б провести пряме донавчання без структурних змін, що було б легшим та швидшим. Проте, становлення точної семантичної відповідності між ознаками різних датасетів може бути нетривіальним завданням, і ігнорування унікальних ознак UNSW-NB15 призводить до втрати потенційно важливої інформації для виявлення аномалій. В реальних сценаріях використання систем виявлення аномалій різні джерела даних часто мають різний набір доступних характеристик, тоді потрібен адаптивний підхід який би краще відображав реальні умови.

Тож для вирішення проблеми розмірності було розроблено спеціальні адаптаційні шари для кожної архітектури:

Для Deep CNN: Створено адаптаційний шар з використанням згорткового шару, який проєціює вхідні 183 ознаки на розмірність, яку очікує оригінальна модель (122 ознаки). Це реалізовано за допомогою додаткового згорткового шару з ядром розміром 1, який виконує роль повнозв'язного шару, але зберігає просторову структуру даних. Схема оновленої архітектури представлена на Рис. 42.

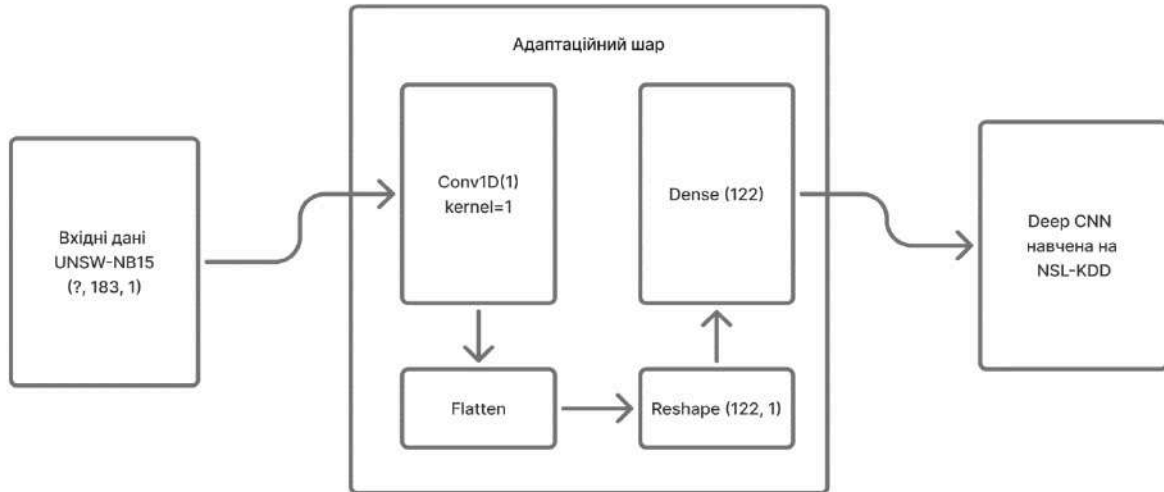


Рис. 42 – Адаптаційна архітектура Deep CNN для UNSW-NB15

Для *Bidirectional LSTM*: Використано шар TimeDistributed з Dense-шаром, який адаптує кожен крок послідовності окремо, зберігаючи часову структуру. Схема оновленої архітектури представлена на Рис. 43.

На відміну від класифікаційних моделей (Deep CNN і Bidirectional LSTM), для яких можна було створити проміжний адаптаційний шар, адаптація автоенкодерів потребувала комплексного підходу через необхідність збереження однакової розмірності входу та виходу.



Рис. 43 – Адаптаційна архітектура Bidirectional LSTM для UNSW-NB15

CNN Autoencoder: Створено нову модель зі структурою, аналогічною оригінальній, але для роботи з 183 ознаками. Збережено ключові елементи:

кількість згорткових фільтрів, стратегії пулінгу та нормалізації. Архітектура залишилась симетричною: енкодер стискає дані, декодер відновлює їх до початкової розмірності. Ваги спільних компонентів ініціалізовані значеннями з оригінальної моделі. Схема оновленої архітектури представлена на Рис. 44.

GRU Autoencoder: Збережено часову структуру з 10 кроками, але з більшою кількістю ознак на кожен крок. Архітектура "енкодер-RepeatVector-декодер" залишилась незмінною. Вихідний шар адаптовано для реконструкції 183 ознак замість 122. Ваги GRU клітин адаптовані для роботи з новою розмірністю входу. Схема оновленої архітектури зображена на Рис. 45.

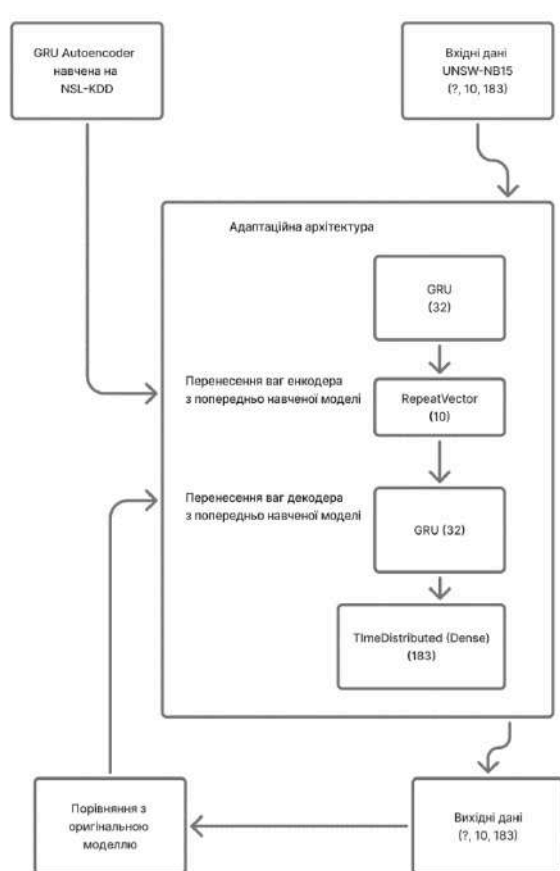


Рис. 44 – Адаптаційна архітектура архітектура

CNN Autoencoder для UNSW-NB15

Аналіз результатів експерименту

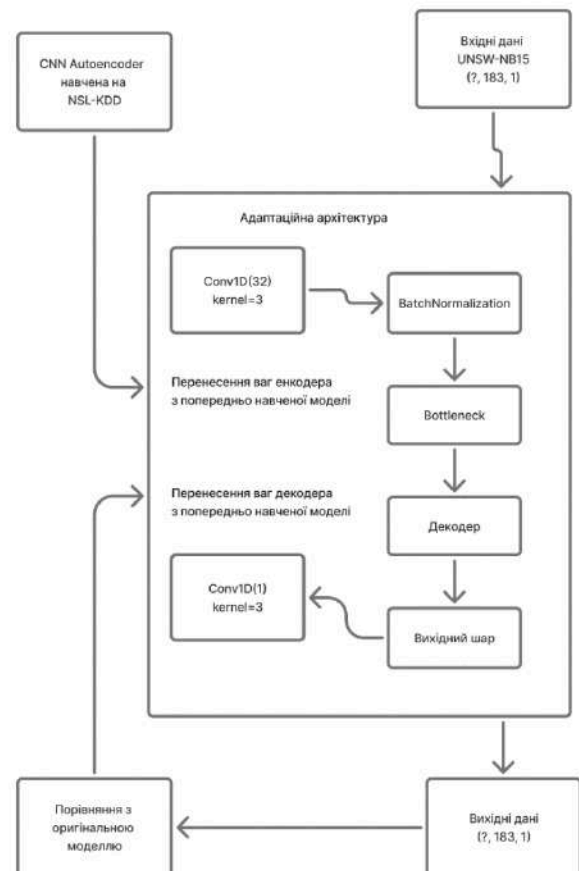


Рис. 45 – Адаптаційна

GRU Autoencoder для UNSW-NB15

Результати тестування моделей на UNSW-NB15 без донавчання наведені в Таблиці 5.

Таблиця 5 – Результати тестування моделей на наборі UNSW-NB15 без донавчання

Модель	F1 Score	ROC AUC	Precision	Recall	Попіг
Deep CNN	0.7102	0.4292	0.5506	1.0000	0.001
Bidirectional LSTM	0.7105	0.7338	0.7422	0.6815	0.4730
CNN Autoencoder	0.7191	0.5224	0.5651	0.9883	0.057
GRU Autoencoder	0.7107	0.5942	0.5514	0.9994	0.069

Аналіз результатів експерименту виявив наступні закономірності:

1. *Загальна ефективність*: Усі моделі продемонстрували посередню ефективність при прямому перенесенні, з F1 Score в діапазоні 0.71-0.72, що значно нижче за їх продуктивність на оригінальному датасеті NSL-KDD (0.87-0.98).
2. *Bidirectional LSTM* показала найкращу здатність до прямого перенесення з ROC AUC 0.7338, що демонструє її кращу здатність до узагальнення на нові дані порівняно з іншими моделями.
3. *Баланс Precision і Recall*: Спостерігається цікава закономірність - CNN і автоенкодера схильні до дуже високого Recall (0.99-1.0) при низькій Precision (0.55-0.57), тоді як Bidirectional LSTM демонструє більш збалансовані показники (Precision 0.74, Recall 0.68).
4. *Найнижча трансферність у Deep CNN*: Модель Deep CNN, яка показувала відмінні результати на NSL-KDD, продемонструвала найнижчий ROC AUC (0.4292) при прямому перенесенні, що свідчить про її сильну спеціалізацію на характеристиках початкового датасету.

5. *Автоенкодера зберігають чутливість до аномалій:* Обидва автоенкодера зберегли здатність виявляти майже всі аномалії ($\text{Recall} > 0.98$), але з великою кількістю хибних спрацювань (низька Precision), що є скоріш за все результатом різного розподілу нормального трафіку в двох датасетах.

Експеримент показав, що пряме перенесення моделей між різними мережевими середовищами має обмежену ефективність. Рекурентні моделі (Bidirectional LSTM) демонструють кращу трансферність, що може бути пов'язано з їх здатністю моделювати часові залежності, які є більш універсальними між різними середовищами.

3.2.3 Експеримент з донавчанням моделей на датасеті UNSW-NB15

З метою визначення впливу донавчання (fine-tuning) моделі на її продуктивність, а також визначення оптимального обсягу даних для цієї задачі, було проведено серію експериментів, у яких варіювався відсоток даних з UNSW-NB15, що використовувався для донавчання попередньо навчених на NSL-KDD моделей. Концептуально процес донавчання ґрунтується на припущенні, що попередньо навчена модель вже набула певних корисних представлень і абстракцій, які можуть бути застосовані до нової предметної області. При донавчанні відбувається модифікація параметрів (ваг) моделі з метою їх адаптації до нових даних, зберігаючи при цьому базові абстракції високого рівня.

Методологія експерименту

В експерименті брали участь чотири моделі, що показали найкращі результати в попередніх експериментах на NSL-KDD: *Deep CNN*, *Bidirectional LSTM*, *CNN Autoencoder*, *GRU Autoencoder*.

Підхід до формування вибірок: Для кожної моделі було створено серію навчальних вибірок різного розміру шляхом випадкового відбору з повного набору даних UNSW-NB15. Розмір вибірок варіювався відповідно до наступних відсотків від загального обсягу даних: 1%, 5%, 10%, 25% та 50%. Це дозволило дослідити, як змінюється ефективність моделей при збільшенні кількості доступних для донавчання даних.

Процедура донавчання: Для кожної моделі та для кожного розміру вибірки було проведено процедуру донавчання з однаковими гіперпараметрами: 10 епох, розмір батчу 64, оптимізатор adam, функція втрат: для класифікаторів (*Deep CNN*, *Bidirectional LSTM*) – `binary_crossentropy`, для автоенкодерів (*CNN Autoencoder*, *GRU Autoencoder*) – `mean squared error (MSE)`.

Оцінка ефективності: Після донавчання кожна модель оцінювалася на тестовій вибірці UNSW-NB15 за допомогою тих самих метрик, що використовувалися в попередніх експериментах.

Реалізація експерименту: Для застосування моделей на даних UNSW-NB15 були використані адаптери з попереднього експерименту, для яких далі було застосовано донавчання на вибірках різного розміру. Особливу увагу було приділено аналізу ефективності донавчання при обмеженому обсязі даних, що має важливе практичне значення в реальних сценаріях, де розмічені дані часто є обмеженим ресурсом. Були проведені експерименти з різними відсотками від загального набору даних UNSW-NB15: від дуже обмежених (1%) до субстантивних (50%) вибірок. Це дозволило оцінити, наскільки різні архітектури чутливі до кількості доступних даних для донавчання. Для сумісності результатів для всіх експериментів використовувалися фіксовані гіперпараметри: розмір міні-батчу 64, кількість епох 10, швидкість навчання 0.001.

Аналіз результатів експерименту

Результати експерименту для кожного з обсягів даних наведені в Таблиці 6 та на Рис. 46.

Таблиця 6 – Результати експерименту з донавчанням моделей на різних обсягах даних

Модель	% даних	F1 Score	ROC AUC	Precision	Recall	Попіг
Deep Cnn	1%	0.7102	0.5	0.5506	1	0.001
Deep Cnn	5%	0.8449	0.8904	0.7375	0.9889	0.19
Deep Cnn	10%	0.8525	0.9317	0.8139	0.8951	0.606
Deep Cnn	20%	0.8778	0.9519	0.9192	0.8399	0.668
Deep Cnn	30%	0.8247	0.9553	0.9864	0.7086	0.001
Deep Cnn	50%	0.8052	0.776	0.7698	0.8439	0.001
Bidirectional Lstm	1%	0.9599	0.9893	0.9735	0.9467	0.689
Bidirectional	5%	0.9692	0.993	0.9728	0.9657	0.304

Lstm						
Bidirectional Lstm	10%	0.9736	0.9956	0.9812	0.966	0.604
Bidirectional Lstm	20%	0.9812	0.9978	0.9833	0.9791	0.49
Bidirectional Lstm	30%	0.9701	0.9955	0.974	0.9663	0.677
Bidirectional Lstm	50%	0.9792	0.9975	0.9822	0.9762	0.317
Cnn Autoencoder	1%	0.7281	0.5086	0.5894	0.9523	0.06
Cnn Autoencoder	5%	0.7173	0.5463	0.5784	0.9439	0.03
Cnn Autoencoder	10%	0.7445	0.6888	0.6181	0.936	0.003
Cnn Autoencoder	20%	0.7102	0.6515	0.5506	1	0.001
Cnn Autoencoder	30%	0.7102	0.5995	0.5506	1	0.001
Cnn Autoencoder	50%	0.7394	0.6299	0.5902	0.9896	0.001
Gru Autoencoder	1%	0.7271	0.6331	0.5736	0.9928	0.048
Gru Autoencoder	5%	0.7349	0.6649	0.5849	0.9883	0.048
Gru Autoencoder	10%	0.7349	0.6511	0.5836	0.9922	0.038
Gru Autoencoder	20%	0.7387	0.6756	0.5889	0.9905	0.045
Gru Autoencoder	30%	0.7368	0.6752	0.5856	0.9932	0.039
Gru Autoencoder	50%	0.7504	0.729	0.6153	0.9615	0.068

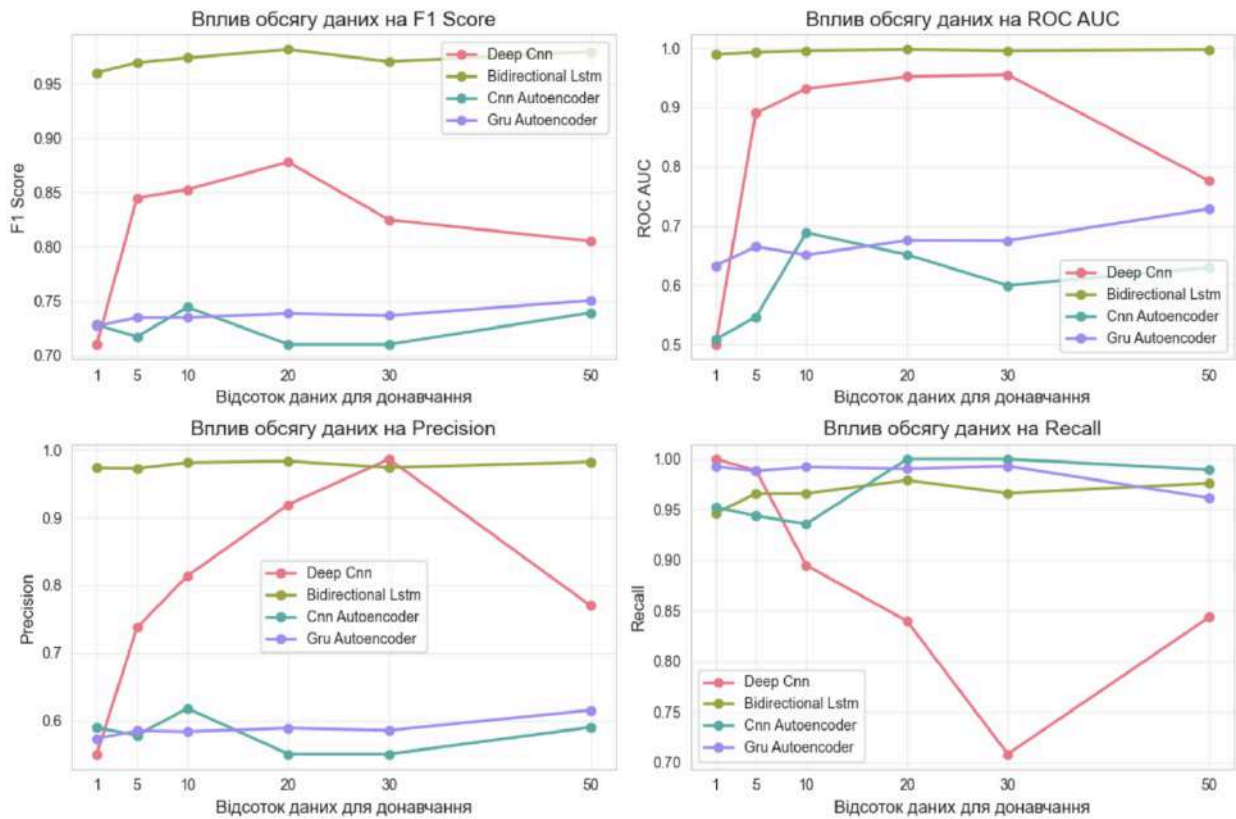


Рис. 46 – Результати експерименту з донавчанням моделей на різних обсягах даних

Deep CNN демонструє найбільш виражену залежність ефективності від кількості даних для донавчання, що проявляється у наступних закономірностях:

1. *Різке покращення при переході від 1% до 5% даних* – F1-міра зростає з 0.7102 до 0.8449, а ROC AUC з 0.5000 до 0.8904. Це вказує на те, що навіть 5% даних суттєво підвищують здатність моделі адаптуватися до нового розподілу.
2. *Оптимальний обсяг даних спостерігається при використанні 20% вибірки*, де модель досягає максимального F1 Score 0.8778 і високого ROC AUC 0.9519.
3. *Баланс між precision і recall змінюється зі збільшенням обсягу даних* – спостерігається перехід від високого recall і низької precision при 1% (0.5506/1.0000) до більш збалансованих показників при 20% (0.9192/0.8399).

4. *Нелінійний характер залежності* – при збільшенні обсягу даних понад 20% спостерігається погіршення F1 Score і зміна балансу між precision та recall, що може свідчити про перенавчання моделі на особливостях нового датасету та втрату переваг трансферного навчання.

Bidirectional LSTM демонструє найвищу ефективність серед усіх досліджених архітектур з наступними особливостями:

1. *Висока базова трансферність* – навіть з 1% даних модель досягає F1 Score 0.9599 і ROC AUC 0.9893, що суттєво перевершує інші архітектури.
2. *Поступове покращення з збільшенням обсягу даних* – спостерігається стабільне зростання F1 Score з максимумом 0.9812 при 20% даних.
3. *Стабільність метрик при різних обсягах даних* – малі коливання F1 Score (від 0.9599 до 0.9812) свідчать про високу стійкість моделі до обсягу даних для донавчання.
4. *Збалансованість metrics* – висока precision і recall для всіх обсягів даних (>0.94), що вказує на ефективне виявлення аномалій без значного зростання хибних спрацювань.
5. *Відсутність значної деградації при збільшенні обсягу даних* – навіть при 50% даних модель зберігає високу ефективність (F1 Score 0.9792).

CNN Autoencoder і GRU Autoencoder демонструють суттєво відмінну поведінку порівняно з класифікаційними моделями:

1. *Обмежене покращення з збільшенням даних* – для обох автоенкодерів спостерігається лише незначне підвищення F1 Score при збільшенні обсягу даних (для CNN Autoencoder з 0.7281 до 0.7445, для GRU Autoencoder з 0.7271 до 0.7504).
2. *Висока чутливість при низькій точності* – характерною особливістю автоенкодерів залишається висока повнота (recall >0.93) при низькій точності (precision $\sim 0.55-0.62$), що підтверджує схильність до хибних спрацювань.

3. *Краща адаптивність GRU Autoencoder* – порівняно з CNN Autoencoder, GRU Autoencoder демонструє більш стабільне покращення метрик з збільшенням обсягу даних, особливо помітне при 50% (F1 Score 0.7504, ROC AUC 0.7290).
4. *Нестабільність порогів* – для автоенкодерів спостерігається значна варіація оптимальних порогів класифікації, що вказує на складність налаштування цих моделей для практичного застосування.

Порівняння результатів донавчання на різних обсягах даних дозволяє зробити наступні **висновки**:

1. *Bidirectional LSTM демонструє найкращу трансферність серед усіх моделей при будь-якому обсязі даних*. Навіть з 1% даних ця архітектура досягає F1 Score, який перевершує результати інших моделей при 50% даних.
2. *Deep CNN потребує більшого обсягу даних для ефективної адаптації*, але демонструє значний потенціал при використанні оптимальної кількості даних (20%).
3. *Автоенкодери показують обмежену здатність до адаптації через донавчання*, що може бути пов'язано з їх специфічним підходом до навчання (моделювання лише нормальних зразків) та різним розподілом даних між датасетами.
4. *Різні темпи адаптації – класифікаційні моделі (особливо CNN) демонструють більш стрімке покращення при збільшенні обсягу даних*, ніж автоенкодери.

3.2.4 Оцінка обчислювальної ефективності моделей

Поряд з точністю та здатністю до трансферного навчання, обчислювальна ефективність є критично важливим критерієм вибору архітектури нейронної мережі для практичного застосування в системах виявлення аномалій. Особливо це

актуально для систем реального часу, де швидкість обробки даних безпосередньо впливає на можливість своєчасного реагування на потенційні загрози.

Методологія проведення експерименту

Для комплексної оцінки обчислювальної ефективності досліджуваних моделей був проведений експеримент, що охоплював три ключові аспекти:

1. Аналіз параметричної складності моделей: визначення загальної кількості параметрів у кожній моделі, оцінка розміру моделі в мегабайтах, співвідношення між параметричною складністю та продуктивністю моделі
2. Вимірювання швидкості інференсу: час обробки одиничного зразка даних (важливо для системи реагування в реальному часі), час обробки пакету зразків (1000 зразків) для оцінки здатності до пакетної обробки, ефективність розпаралелювання обчислень при збільшенні розміру батчу
3. Оцінка швидкості навчання: час донавчання на обмеженій вибірці (1000 зразків), середній час на одну епоху навчання, співвідношення між швидкістю навчання та покращенням моделі

Аналіз результатів експерименту

Результати експерименту для всіх чотирьох моделей наведені в Таблиці 7 та на Рис. 47.

Таблиця 7 – Результати експерименту з обчислювальної ефективності моделей

Модель	Розмір (МБ)	К-ть параметрів	інференс – 1 зразок (мс)	інференс – 1000 зразків (с)	навчання – епоха
Deep CNN	0.66	166,529	107.341	0.132	1.282
Bidirectional LSTM	0.64	168,257	92.019	1.151	2.809
CNN Autoencoder	0.07	19,601	41.513	0.133	1.438

GRU Autoencoder	0.13	33,207	83.321	0.203	0.735
-----------------	------	--------	--------	-------	-------

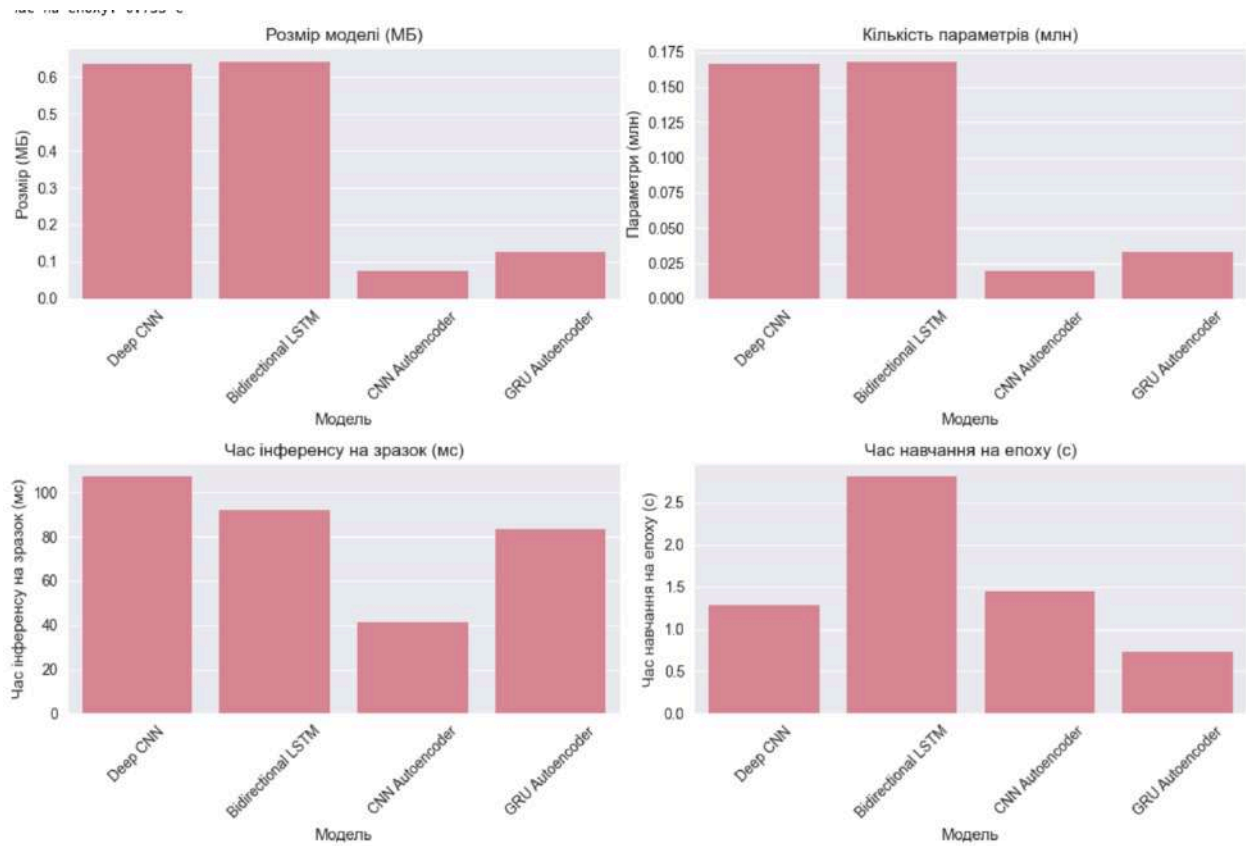


Рис. 47 – Результати експерименту з оцінювальної ефективності моделей

Найбільш збалансованою архітектурою з точки зору обчислювальної ефективності та продуктивності є *Bidirectional LSTM*, яка демонструє: найкращу точність виявлення аномалій, середні показники швидкості інференсу для одиничних зразків, та прийнятний розмір моделі, проте має найгірший час для пакетної обробки та найдовший час навчання.

Найбільш ефективною з точки зору розміру та швидкості є *CNN Autoencoder*: найменший розмір моделі (0.07 МБ), найкраща швидкість інференсу для одиничних зразків та ефективна пакетна обробка, проте посередні результати виявлення аномалій.

Оптимальною для обробки в реальному часі можна вважати Deep CNN, яка: демонструє найкращий час пакетної обробки (0.132 с) та має хорошу здатність до адаптації через донавчання, проте повільніша при обробці одиничних зразків.

Найефективнішою для частого донавчання є GRU Autoencoder: найшвидший час навчання (0.735 с/епоха), компактний розмір (0.13 МБ) та середня швидкість інференсу, проте попередні експерименти показали обмежену здатність до адаптації через донавчання, тож доцільність такого донавчання буде сумнівною.

3.2.5. Аналіз дослідження трансферності нейромережевих моделей

Проведене дослідження трансферності моделей виявлення аномалій на основі нейронних мереж дозволило отримати комплексне розуміння їх поведінки при перенесенні між різними доменами даних. Аналіз охоплював основні аспекти трансферного навчання: пряму трансферність без донавчання, ефективність донавчання на різних обсягах даних та обчислювальну складність моделей. На основі результатів експериментів можна зробити наступні ключові висновки.

Порівняльний аналіз архітектур

1. *Рекурентні архітектури на основі LSTM демонструють виняткову здатність до трансферу як без донавчання, так і з мінімальним донавчанням. Bidirectional LSTM, як представник цієї архітектури, показала найкращі результати (F1 Score 71,05% без донавчання і 95,99% з донавчанням на 1% даних), що свідчить про фундаментальну здатність рекурентних архітектур ефективно моделювати часові залежності, які зберігаються навіть при зміні структури даних.*
2. *Згорткові архітектури (CNN) потребують суттєвого донавчання для ефективної адаптації. Без донавчання результати посередні, але при використанні достатнього обсягу даних (близько 20%) ці архітектури демонструють значне покращення, що вказує на їхню спроможність*

адаптуватися до нових патернів даних за умови наявності адекватної вибірки для донавчання.

3. *Автоенкодер* виявляють обмежену здатність до трансферу з незначним покращенням при донавчанні. Це стосується як згорткових (CNN Autoencoder), так і рекурентних (GRU Autoencoder) варіантів, що вказує на фундаментальне обмеження цього архітектурного підходу: сильну залежність від конкретного розподілу нормальних даних у навчальній вибірці.
4. *Трансферність архітектури* не корелює безпосередньо з її ефективністю на оригінальному датасеті. Автоенкодер, які показали найкращі результати на NSL-KDD, виявились найменш трансферними, тоді як рекурентні архітектури продемонстрували найкращу здатність до адаптації.

Вплив обсягу даних на ефективність трансферу

Оптимальний обсяг даних для донавчання класифікаційних моделей становить близько 20%, після чого не спостерігається значного покращення або навіть відбувається погіршення результатів.

Рекурентні архітектури продемонстрували найвищу ефективність навіть при мінімальному донавчанні (1-5% даних), що свідчить про їхню здатність швидко адаптуватися до нових розподілів даних.

Автоенкодерні архітектури демонструють значно гіршу здатність до донавчання, що вказує на фундаментальне обмеження підходу, заснованого на моделюванні лише нормальних зразків даних, при перенесенні на нові домени.

Баланс між точністю та обчислювальною ефективністю

Рекурентні архітектури (LSTM, GRU) забезпечують оптимальний баланс між точністю та обчислювальною складністю для одиничних зразків, але демонструють обмеження при пакетній обробці та більший час навчання через послідовну природу обчислень.

Згорткові архітектури (CNN) оптимальні для пакетної обробки великих обсягів даних завдяки ефективному розпаралелюванню операцій, що робить їх привабливими для офлайн-аналізу та систем, що працюють з великими потоками даних.

Автоенкодерні архітектури компактніші за розміром та ефективніші з точки зору обчислень, але демонструють гірші показники точності при трансфері. Зокрема, CNN-базовані автоенкодери надзвичайно компактні (до 10 разів менше параметрів порівняно з класифікаційними моделями), а GRU-базовані автоенкодери демонструють найшвидший час навчання серед усіх архітектур

GRU Autoencoder демонструє найшвидший час навчання (0.735 с/епоха), що може бути критичним для сценаріїв з частим оновленням моделі, але з обмеженою здатністю до підвищення якості через донавчання.

3.3 Рекомендації щодо вибору архітектури та побудови системи виявлення аномалій

Отримані під час дослідження результати стали основою для формування комплексу прикладних рекомендацій з оптимізації архітектур нейронних мереж і створення дієвих систем виявлення аномалій. Ці рекомендації враховують як результати порівняльного аналізу архітектур, так і дослідження їх трансферності, обчислювальної ефективності та особливостей застосування в різних умовах.

3.3.1 Критерії вибору нейромережевої архітектури

Проведені дослідження демонструють, що не існує універсальної «найкращої» архітектури для всіх сценаріїв виявлення аномалій. Вибір оптимальної архітектури нейронної мережі для системи виявлення мережеских аномалій повинен ґрунтуватися на комплексному аналізі наступних факторів.

Характер аномалій та типи даних

Точкові аномалії: Згідно як з теоретичним дослідженням літературних джерел, так і за результатами експериментів, автоенкодери демонструють високу

ефективність для виявлення ізольованих аномалій завдяки здатності моделювати нормальний розподіл даних з точністю до окремих спостережень.

Контекстуальні аномалії: Як підтверджено нашими експериментами з Bidirectional LSTM, рекурентні архітектури більш ефективні для виявлення аномалій, залежних від контексту, завдяки моделюванню часових залежностей.

Колективні аномалії: Поєднання CNN для просторового аналізу та LSTM для моделювання послідовностей оптимально для виявлення колективних аномалій.

Стабільність мережевого середовища

Стабільні середовища: У середовищах з низькою динамікою змін автоенкодері демонструють найвищу точність (GRU Autoencoder: F1 Score 98,5%, ROC AUC 99,1%).

Динамічні середовища: Для середовищ із частими змінами характеристик трафіку рекурентні архітектури демонструють кращу адаптивність, що підтверджується експериментами з донавчанням (Bidirectional LSTM: F1 Score 95,99% при донавчанні на 1% даних).

Доступність розмічених даних

При наявності великого обсягу розмічених даних (>10000 зразків з мітками) оптимальними є класифікаційні підходи (CNN-based, LSTM-based).

При обмеженій кількості розмічених даних або їх відсутності кращими є автоенкодерні підходи.

Співвідношення важливості хибних спрацювань та пропущених аномалій

Для систем де пропуск аномалії є критичним, рекомендуються автоенкодері (особливо LSTM-based Autoencoder), які демонструють найвищий recall навіть при зміні середовища.

У системах, де важлива мінімізація хибних спрацювань, перевагу мають класифікаційні моделі, особливо CNN.

Обчислювальні ресурси та вимоги до швидкодії:

Системи реального часу: Згідно з проведеними експериментами, CNN архітектури забезпечують найкращу пропускну здатність при пакетній обробці (Deep CNN: 0,132 с на 1000 зразків).

Системи з обмеженими ресурсами: Автоенкодери демонструють найкращий баланс між розміром моделі та точністю (CNN Autoencoder: 0,07 МБ, F1 Score 91,1% на оригінальних даних, GRU Autoencoder: 0,13 МБ F1 Score 98,5%).

3.3.2 Архітектурні особливості моделей та їх вплив на ефективність

CNN

Глибина мережі: Збільшення глибини до 3-4 згорткових шарів суттєво покращує здатність CNN виявляти складні патерни, що підтверджується перевагою Deep CNN (F1 Score 90,9%) над базовою CNN (F1 Score 90,0%).

Розмір та кількість фільтрів: Використання різних розмірів фільтрів (3×1, 5×1) дозволяє виявляти патерни різних масштабів, а поступове збільшення кількості фільтрів (32→64→128→256) підвищує здатність до вилучення ієрархічних ознак, що підтверджується експериментами з Deep CNN.

Залишкові з'єднання: Всупереч теоретичним очікуванням, додавання залишкових з'єднань (Residual CNN) не призвело до істотного покращення результатів порівняно з Deep CNN, що свідчить про відсутність проблеми зникаючого градієнта для даної задачі та наборів даних.

Рекурентні нейронні мережі

Використання стандартної RNN не рекомендується через проблему зникаючого градієнта.

Двонаправленість: Bidirectional LSTM показала найкращі результати серед рекурентних архітектур (F1 Score 88,4%, ROC AUC 92,4%), що підтверджує важливість аналізу послідовності в обох напрямках для виявлення аномалій.

Вибір між LSTM та GRU: Хоча теоретично GRU має перевагу в обчислювальній ефективності, експерименти показали, що стандартна GRU поступається Bidirectional LSTM за F1 Score (88,3% проти 88,4%), але має вищий recall (91,7% проти 89,5%), що може бути критичним для систем, де пропуск аномалій є більш критичним, ніж хибні спрацювання.

Механізм уваги: Додавання механізму уваги до LSTM не привело до покращення результатів (F1 Score 87,0% порівняно з 88,4% для Bidirectional LSTM).

Автоенкодера

Базова архітектура: Всі досліджені автоенкодера показали високий рівень recall (95,1-99,8%), що підтверджує їх теоретичну здатність ефективно виявляти відхилення від вивченого нормального розподілу.

Рекурентні vs згорткові: GRU Autoencoder демонструє значно вищу точність (F1 Score 98,5%, ROC AUC 99,1%) порівняно з CNN Autoencoder (F1 Score 91,1%, ROC AUC 95,3%), що підтверджує перевагу рекурентних архітектур для моделювання динамічних особливостей мережевого трафіку.

Розмірність латентного простору: Зменшення розмірності до 1/4-1/8 від початкової (як у GRU Autoencoder) забезпечує оптимальний баланс між стисненням та збереженням інформації.

3.3.3 Підходи до трансферного навчання моделей

Результати експериментів з трансферного навчання між NSL-KDD та UNSW-NB15 дозволяють сформулювати наступні рекомендації.

Базова трансферність: Згідно з результатами експериментів, усі досліджені архітектури демонструють обмежену здатність до прямого трансферу без

донавчання (F1 Score 0,71-0,72), що свідчить про значущість відмінностей між різними мережевими середовищами і потребу в донавчанні моделей в динамічних середовищах.

Розмірність даних: Розбіжність у розмірності ознакових просторів різних датасетів (122 для NSL-KDD vs 183 для UNSW-NB15) вимагає спеціальних методів адаптації архітектури.

Рекурентні архітектури: Bidirectional LSTM демонструє найвищу базову трансферність (F1 Score 71,05% без донавчання) та найкращу адаптивність при мінімальному донавчанні (F1 Score 95,99% з донавчанням на 1% даних).

Згорткові архітектури: CNN потребують більшого обсягу даних для ефективною адаптації, але демонструють значне покращення при донавчанні на 20% даних (Deer CNN: F1 Score 87,78%, ROC AUC 95,19%).

Автоенкодеру: Як CNN Autoencoder, так і GRU Autoencoder демонструють обмежену здатність до адаптації через донавчання (F1 Score 74,45% та 75,04% при донавчанні на 50% даних), що вказує на їх високу специфічність до розподілу нормальних даних у вихідному датасеті.

Оптимальні підходи: Для класифікаційних моделей оптимальним є донавчання на 20% нових даних із нижчою швидкістю (learning rate) (0,0001-0,001) порівняно з початковим навчанням. Для адаптації до різної розмірності даних ефективним є використання проміжних адаптаційних шарів (згорткових з ядром 1×1 для CNN, TimeDistributed для LSTM).

3.3.4 Рекомендації щодо побудови повноцінної системи виявлення мережових аномалій

На основі аналізу архітектурних підходів, представлених у літературі, та результатів власних експериментів, рекомендується багаторівнева архітектура системи виявлення аномалій, що забезпечує баланс між точністю, адаптивністю та продуктивністю.

Рівень збору та попередньої обробки даних: Модулі збору даних з різних джерел, фільтрації шуму, екстракції ознак та нормалізації. Якісна попередня обробка даних критично важлива для ефективності нейромережових моделей. Наприклад, використання StandardScaler для числових ознак та One-Hot кодування для категоріальних ознак, як показано в експериментах з NSL-KDD та UNSW-NB15.

Рівень виявлення аномалій: Вибір конкретної нейромережової архітектури залежить від контексту. Рекомендації наведені в розділі 3.2. Також можливе використання ансамблю нейромережових моделей різних архітектур. Поєднання моделей, що спеціалізуються на різних типах аномалій, підвищує загальну ефективність системи.

Рівень адаптації та навчання: Модулі збору позитивних та негативних прикладів, донавчання моделей, валідації та оновлення для запобігання концептуальному дрейфу.

Рівень аналізу та реагування: Модулі агрегації та кореляції виявлених аномалій, пріоритизації, генерації сповіщень та звітів. Рекомендується імплементація адаптивних порогів класифікації, що враховують специфіку конкретного мережевого середовища. Експериментально підтверджено, що оптимізація порогу за F1 Score забезпечує найкращий баланс між precision та recall, що особливо важливо для виявлення аномалій в різних мережових контекстах.

ВИСНОВКИ

У даній роботі було проведено комплексне дослідження систем виявлення аномалій на основі нейронних мереж, фокусуючись на аналізі мережевого трафіку. Робота поєднала теоретичний аналіз із експериментальним дослідженням різних архітектур нейронних мереж. Теоретичний аналіз дозволив систематизувати знання про типи мережевих аномалій (точкові, контекстуальні, колективні) та дослідити основні архітектури нейронних мереж, що застосовуються для їх виявлення. Аналіз літератури виявив невирішені питання щодо порівняльної ефективності різних архітектур, їх трансферності та балансу між обчислювальною ефективністю та точністю виявлення. В експериментальній частині було розроблено, навчено та протестовано дев'ять моделей нейронних мереж на наборі даних NSL-KDD. Результати експериментів показали, що GRU Autoencoder демонструє найвищу, майже ідеальну точність виявлення аномалій (F1 Score 98,5%, ROC AUC 99,1%). Серед CNN моделей найкращі результати показала Deep CNN (F1 Score 90,9%, ROC AUC 93,1%), а серед рекурентних архітектур – Bidirectional LSTM (F1 Score 88,4%, ROC AUC 92,4%). Експерименти з трансферного навчання на наборі даних UNSW-NB15 показали, що Bidirectional LSTM демонструє найкращу трансферність як без донавчання, так і з мінімальним донавчанням. CNN архітектури потребують більше нових даних для ефективної адаптації, а автоенкодери показали обмежену здатність до адаптації через донавчання. На основі проведеного дослідження було сформульовано рекомендації щодо вибору архітектури нейронної мережі та побудови системи виявлення аномалій з урахуванням різних сценаріїв застосування.

Виконана робота має як наукову цінність для систематизації знань і подальших досліджень, так і практичну цінність для розробників систем виявлення мережевих аномалій, надаючи обґрунтовані рекомендації щодо вибору архітектури нейронної мережі та її налаштування залежно від конкретних вимог та обмежень.

Для подальшого розвитку дослідження перспективними напрямками є:

1. Розробка та тестування гібридних архітектур, що поєднують переваги різних типів нейронних мереж, наприклад, CNN для вилучення просторових ознак з LSTM для моделювання часових залежностей.
2. Поглиблене дослідження методів доменної адаптації для підвищення трансферності моделей між різними мережевими середовищами.
3. Дослідження механізмів інтерпретації результатів нейромережових моделей для підвищення довіри до автоматичних систем виявлення аномалій.
4. Валідація результатів у реальних мережових середовищах з природним розподілом аномалій та оцінка ефективності різних архітектур для специфічних типів атак.

ВИКОРИСТАНІ ДЖЕРЕЛА

1. Aleskerov E., Freisleben B., Rao B. Cardwatch: A Neural Network Based Database Mining System for Credit Card Fraud Detection // Proceedings of IEEE Computational Intelligence for Financial Engineering. 1997. P. 220–226.
2. Spence C., Parra L., Sajda P. Detection, synthesis and compression in mammographic image analysis with a hierarchical image probability model // Proceedings of the IEEE Workshop on Mathematical Methods in Biomedical Image Analysis. Washington, DC, USA: IEEE Computer Society, 2001. P. 3.
3. García-Teodoro P., Díaz-Verdejo J., Maciá-Fernández G., Vázquez E. Anomaly-Based Network Intrusion Detection: Techniques, Systems, and Challenges // Computers & Security. 2009. Vol. 28, no. 1–2.
4. Grubbs F. E. Procedures for detecting outlying observations in samples // Technometrics. 1969. Vol. 11, no. 1. P. 1–21.
5. Chandola V., Banerjee A., Kumar V. Anomaly detection: A survey // ACM Computing Surveys. 2009. Vol. 41.
6. Information Journal [Електронний ресурс]. Режим доступу: https://www.mdpi.com/journal/information/special_issues/2WK033A8UH
7. TechMagic Blog. AI Anomaly Detection [Електронний ресурс]. Режим доступу: <https://www.techmagic.co/blog/ai-anomaly-detection>
8. Pang G., Shen C., Cao L., Hengel A. V. D. Deep learning for anomaly detection: A review // ACM Computing Surveys. 2021. Vol. 54, no. 2. P. 1–38.
9. Parimala V. Introductory Chapter: Anomaly Detection – Recent Advances, AI and ML Perspectives and Applications. 2024. DOI: 10.5772/intechopen.113968.
10. Bishop C. M., Bishop H. Deep Learning: Foundations and Concepts. Springer International Publishing, 2023.
11. Bhuyan M. H., Bhattacharyya D. K., Kalita J. K. Network Anomaly Detection: Methods, Systems and Tools // IEEE Communications Surveys & Tutorials. 2014. Vol. 16, no. 1.
12. Гайдур Г. І., Гахов С. О., Бригинець А. А. Виявлення мережевих аномалій з використанням алгоритмів нейронних мереж // Державний університет телекомунікацій, Київ. 2023. № 1 (78). DOI: 10.31673/2412-4338.2023.010416.

13. Колодчак О. М. Сучасні методи виявлення аномалій в системах виявлення вторгнень. Національний університет "Львівська політехніка", 2012.
14. Parhizkari S. Anomaly Detection in Intrusion Detection Systems. IntechOpen, 2023. DOI: 10.5772/intechopen.112733.
15. Нікітенко А. О. Системи виявлення мережевих вторгнень на основі нейронних мереж глибокого навчання // Наукові праці ДонНТУ. 2023. № 2 (37).
16. Filho J. E. de A., Brandão L. C. P., Fernandes B., Maciel A. M. A Review of Neural Networks for Anomaly Detection // IEEE Access. 2022.
17. Naseer S. et al. Enhanced Network Anomaly Detection Based on Deep Neural Networks // IEEE Access. 2018. Vol. 6. P. 48231-48246. DOI: 10.1109/ACCESS.2018.2863036.
18. Ahmed M., Mahmood A. N., Hu J. A survey of network anomaly detection techniques // Journal of Network and Computer Applications. 2016. Vol. 60. P. 19-31.
19. Bhattacharyya D. K., Kalita J. Network Anomaly Detection: A Machine Learning Perspective. 2013.
20. Ahmad Z., Khan A. S., Shiang C. W., Abdullah J., Ahmad F. Network intrusion detection system: A systematic study of machine learning and deep learning approaches // Transactions on Emerging Telecommunications Technologies. 2021. Vol. 32, no. 1.
21. Liu Y., Zhang H., Li L., Guan Y. A deep learning based approach for network intrusion detection // 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). 2020. P. 287-294.
22. Kwon D., Kim H., Kim J., Suh S. C., Kim I., Kim K. J. A survey of deep learning-based network anomaly detection // Cluster Computing. 2019. Vol. 22, no. 1. P. 949-961.
23. Diro A. A., Chilamkurti N. Distributed attack detection scheme using deep learning approach for Internet of Things // Future Generation Computer Systems. 2018. Vol. 82. P. 761-768.

24. Yu Y., Long J., Cai Z. Network intrusion detection through stacking dilated convolutional autoencoders // Security and Communication Networks. 2018. Vol. 2018. P. 1-10.
25. Wang W., Sheng Y., Wang J., Zeng X., Ye X., Huang Y., Zhu M. HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection // IEEE Access. 2017. Vol. 6. P. 1792-1806.
26. Wang W., Jiang Y., Luo Y. A graph neural network approach for detecting network anomalies // Proceedings of the ACM International Conference on Information and Knowledge Management. 2021. P. 3113-3122.
27. LeCun Y., Bengio Y., Hinton G. Deep Learning // Nature. 2015. Vol. 521. P. 436-44.
28. Karpathy A. Connecting Images and Natural Language: дис. ... PhD: Stanford University, 2016.
29. Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press, 2016.
30. Wang W., Zhu M., Wang J., Zeng X., Yang Z. End-to-end encrypted traffic classification with one-dimensional convolution neural networks // 2018 IEEE International Conference on Intelligence and Security Informatics (ISI). 2018. P. 43-48.
31. Hochreiter S., Schmidhuber J. Long short-term memory // Neural Computation. 1997. Vol. 9, no. 8. P. 1735-1780.
32. Kim J., Kim M., Kim H. Automatic analysis of security log using LSTM-RNN algorithm // Applied Sciences. 2019. Vol. 9, no. 21.
33. Gated Recurrent Unit Networks [Электронный ресурс] // GeeksforGeeks. Режим доступа: <https://www.geeksforgeeks.org/gated-recurrent-unit-networks/>
34. Li Y., Xu Y., Liu Z., Hou H., Zheng Y., Xin Y., Zhao Y., Cui L. Robust detection for network intrusion of industrial IoT based on multi-CNN fusion // Measurement. 2019. Vol. 154.
35. Vincent P., Larochelle H., Lajoie I., Bengio Y., Manzagol P.-A. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion // Journal of Machine Learning Research. 2010. Vol. 11. P. 3371–3408.

36. Masci J., Meier U., Cirean D., Schmidhuber J. Stacked convolutional auto-encoders for hierarchical feature extraction // Artificial Neural Networks and Machine Learning ICANN 2011. 2011. P. 52–59.
37. Linda O., Vollmer T., Manic M. Neural Network-based Intrusion Detection System for Critical Infrastructures // 2009 International Joint Conference on Neural Networks. 2009. P. 1827-1834.
38. Kingma D. P., Welling M. Auto-encoding variational Bayes // International Conference on Learning Representations. 2014.
39. Yang Y., Zheng K., Wu C., Yang Y. Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network // Sensors. 2020. Vol. 20, no. 9.
40. Barford P., Kline J., Plonka D., Ron A. A signal analysis of network traffic anomalies // Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurement. 2002. P. 71-82.
41. Junior G., Rodrigues J., Carvalho L., Al-Muhtadi J., Proença M. A comprehensive survey on network anomaly detection // Telecommunication Systems. 2019. Vol. 70.
42. Casas P., Mazel J., Owezarski P. Unsupervised network intrusion detection systems: Detecting the unknown without knowledge // Computer Communications. 2012. Vol. 35, no. 7. P. 772-783.
43. Barford P., Plonka D. Characteristics of network traffic flow anomalies // Proceedings of the 1st ACM SIGCOMM workshop on internet measurement. 2001. P. 69–73.
44. Jung J., Krishnamurthy B., Rabinovich M. Flashcrowds and denial of service attacks // Proceedings of the 11th international conference on World Wide Web. 2002. P. 293.
45. Beth Dataset [Электронный ресурс] // Kaggle. Режим доступа: <https://www.kaggle.com/datasets/katehighnam/beth-dataset>
46. Zhao X., Leng X., Wang L. et al. Efficient anomaly detection in tabular cybersecurity data using large language models // Scientific Reports. 2025. Vol. 15. P. 3344.

47. Zhou Y., Cheng G., Jiang S., Dai M. Building an efficient intrusion detection system based on feature selection and ensemble classifier // Computer Networks. 2018. Vol. 174.
48. Soltani N., Kenari F. G., Seno S. A. H. A deep autoencoder-based approach for internet traffic classification // 2020 10th International Conference on Computer and Knowledge Engineering (ICCKE). 2020. P. 559-564.
49. Credit Card Fraud Detection [Электронный ресурс] // Kaggle. Режим доступа: <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud>
50. Medical Anomaly Detection [Электронный ресурс] // Kaggle. Режим доступа: <https://www.kaggle.com/datasets/arktis2022/medical-anomaly-detection>
51. Sträter L. P. J., Salehi M., Gavves E., Snoek C. G. M., Asano Y. M. GeneralAD: Anomaly Detection Across Domains by Attending to Distorted Features. Cornell University, 2024.
52. Jain Y., Dabouei A., Xu M. Cross-Domain Learning for Video Anomaly Detection with Limited Supervision. Cornell University, 2024.
53. Wang C. Robust and Cross-domain Anomaly Detection and Mitigation. University of Nevada, 2024.
54. Cho K. et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation // Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2014. P. 1724-1734.
55. Hornik K. Approximation capabilities of multilayer feedforward networks // Neural Networks. 1991. Vol. 4, no. 2. P. 251–257.
56. Thottan M., Ji C. Anomaly detection in IP networks // IEEE Transactions on Signal Processing. 2003. Vol. 51, no. 8. P. 2191-2204.
57. Marnierides A. K., Pezaros D. P., Hutchison D. Detection and mitigation of abnormal traffic behavior in autonomic networked environments // ACM Computing Surveys. 2015. Vol. 47, no. 1. P. 1-51.
58. Nguyen H. X., Roughan M. Methods for determining capacity bottlenecks in dynamic wireless networks // International Journal of Network Management. 2013. Vol. 23, no. 6. P. 447-462.

59. Iliofotou M., Pappu P., Faloutsos M., Mitzenmacher M., Singh S., Varghese G. Network monitoring using traffic dispersion graphs (TDGs) // Proceedings of the 2007 ACM SIGCOMM Internet Measurement Conference. 2011. P. 315-320.
60. Lof A., Nelson R. Annotating network trace data for anomaly detection research // 2014 IEEE 39th conference on local computer networks workshops (LCN workshops). 2014. P. 679–684.
61. Nasr M., Bahramali A., Houmansadr A. DeepCorr: Strong flow correlation attacks on Tor using deep learning // Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. 2018. P. 1962-1976.
62. Wang W., Du X., Wang N. Building a deep learning based intrusion detection system for IoT network // IEEE Access. 2019. Vol. 7. P. 104033-104045.
63. Vinayakumar R., Alazab M., Soman K. P., Poornachandran P., Al-Nemrat A., Venkatraman S. Deep learning approach for intelligent intrusion detection system // IEEE Access. 2019. Vol. 7. P. 41525-41550.
64. Ring M. et al. A survey of network-based intrusion detection datasets. 2019.
65. NSL-KDD Dataset [Электронный ресурс]. Режим доступа: <https://www.kaggle.com/datasets/hassan06/nslkdd>
66. KDD Cup 1999 Dataset [Электронный ресурс]. Режим доступа: <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>
67. UNSW-NB15 Dataset [Электронный ресурс]. Режим доступа: <https://research.unsw.edu.au/projects/unswnb15-dataset>
68. Fast Forward Labs [Электронный ресурс]. Режим доступа: <https://ff12.fastforwardlabs.com/>
69. F1 Score in Machine Learning [Электронный ресурс] // Encord Blog. Режим доступа: <https://encord.com/blog/f1-score-in-machine-learning/>

ДОДАТОК

Опис програмної реалізації

1. Використані технології

Основні бібліотеки та інструменти:

- Python 3.11 - мова програмування
- TensorFlow/Keras 2.x - фреймворк для нейронних мереж
- Pandas - робота з табличними даними
- NumPy - числові обчислення
- Scikit-learn - метрики оцінки та допоміжні функції машинного навчання
- Matplotlib, Seaborn - візуалізація результатів
- Jupyter Lab/Notebook - середовище розробки

2. Структура проекту

- Основні notebook'и експериментів у кореневій папці
- Папка data/ для збереження оброблених даних (*не включається в архів з вихідним кодом через розмір*)
- Папка models/ для збережених моделей
- Папки nsl-kdd/ та unsw-nb15/ з вихідними наборами даних (*не включається в архів з вихідним кодом через розмір*)

3. Основні notebook'и та їх призначення

1. nsl-kdd-exploratory-analysis.ipynb - дослідницький аналіз NSL-KDD
2. nsl-kdd-processing.ipynb - попередня обробка набору даних NSL-KDD для подальших експериментів
3. cnn-nsl-kdd-experiments.ipynb - експерименти з CNN архітектурами
4. lstm-nsl-kdd-experiments.ipynb - експерименти з рекурентними архітектурами
5. autoencoders-nsl-kdd-experiments.ipynb - експерименти з автоенкодерами
6. unsw-nb-processing.ipynb - попередня обробка набору UNSW-NB15
7. unsw-nb15-experiments.ipynb - експерименти з трансферного навчання
8. computational-comparison.ipynb - аналіз обчислювальної ефективності моделей

4. Організація даних та результатів

data/prepared/ - підготовлені дані NSL-KDD у форматі prz

data/processed/ - оброблені CSV файли

data/unsw_prepared/ - підготовлені дані UNSW-NB15

models/ - збережені навчені моделі за архітектурами:

- cnn/ - моделі згорткових мереж

- lstm/ - рекурентні моделі

- autoencoder/ - автоенкодери

- adapted/ - адаптовані моделі для трансферного навчання

results/ - результати експериментів, метрики, графіки:

- computational_complexity/ - результати аналізу ефективності
- transfer_learning/ - результати трансферного навчання
- data_volume_impact/ - вплив обсягу даних на донавчання

5. Порядок виконання експериментів

1. nsl-kdd-exploratory-analysis.ipynb
2. nsl-kdd-processing.ipynb
3. cnn-nsl-kdd-experiments.ipynb
4. lstm-nsl-kdd-experiments.ipynb
5. autoencoders-nsl-kdd-experiments.ipynb
6. unsw-nb-processing.ipynb
7. unsw-nb15-experiments.ipynb
8. computational-comparison.ipynb

Повний архів з вихідним кодом надається в електронному вигляді.