Хмельницький національний університет
Факультет інформаційних технологій
Кафедра кібербезпеки

КВАЛІФІКАЦІЙНА РОБОТА

Яцишиної Катерини Олександрівни

на здобуття ступеня вищої освіти Бакалавра

Система виявлення вторгненнь на основі аномалій для пристроїв IoT

Галузь знань 12 – Інформаційні технології
Спеціальність 125 – Кібербезпека
Освітня програма Кібербезпека

Шифр КРБКБ.2101136.21.01.16 ПЗ

Виконала студентка 4 курсу група КБ-21-1 Катерина ЯЦИШИНА

Керівник доктор технічних наук, професор Михайло КАСЯНЧУК

Нормоконтролер старший викладач Сергій МОСТОВИЙ

До захисту допускаю:
Завідувач кафедри кібербезпеки Юрій КЛЬОЦ

6 06 2025 р.

Хмельницький 2025

втрат.Матриця невідповідностей для OtherNet і IoTNet із збалансованою за класами функцією втрат.

6 Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7 Дата видачі завдання 16 лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
Вибір і затвердження теми кваліфікаційної роботи	Січень-Лютий	
Ознайомлення з предметною областю	Лютий	
Дослідження існуючих рішень	Лютий	
Постановка задачі	Березень	
Визначення загальних принципів рішення задачі	Березень	
Деталізація принципів рішення задачі	Квітень	
Розробка проєктних рішень	Квітень	
Апробація проєктних рішень	Травень	
Оформлення пояснювальної записки згідно вимог	Травень	
Оформлення графічної частини	Червень	
Захист КР	Червень	

Студентка Катерина ЯЦИШИНА

Керівник кваліфікаційної роботи Михайло КАСЯНЧУК

АНОТАЦІЯ

Тема кваліфікаційної роботи: Система виявлення вторгнень на основі аномалій для пристроїв IoT.

Автор роботи: Яцишина Катерина Олександрівна.

Керівник роботи: Касянчук Микола Миколайович

Пояснювальна записка: 64 с., 8 додатків, 17 рисунків, 7 таблиць, 40 джерел.

Графічна частина: 8 плакатів.

ІНТЕРНЕТ РЕЧЕЙ, ГЛИБОКЕ НАВЧАННЯ, ВИЯВЛЕННЯ ВТОРГНЕНЬ, АНОМАЛІЇ, НЕЙРОМЕРЕЖА.

Кваліфікаційна робота бакалавра присвячена розробці та аналізу існуючих рішень систем виявлення вторгнень, що окремо розглядається саме на основі пристроїв IoT.

В роботі проведено аналіз методів виявлення аномалій у трафіку пристроїв Інтернету речей (IoT) з метою розробки ефективної системи виявлення вторгнень. Визначено основні загрози для безпеки IoT-пристроїв, досліджено сучасні підходи до моніторингу та аналізу аномалій у мережевих з'єднаннях. В результаті розроблено концептуальну модель системи виявлення вторгнень, яка базується на методах машинного навчання та поведінковому аналізі. Отримано набір рекомендацій щодо впровадження запропонованої системи для підвищення рівня кібербезпеки IoT-інфраструктури.

02. 08.2025 _____

ABSTRACT

Subject of qualification work: Anomaly-Based Intrusion Detection System for IoT Devices.

Author: Yatsyshyna Kateryna Oleksandrivna..

Head of work:Kasianchuk Mykola Mykolaiovych.

Explanatory note: 64 p., 8 appendices, 17 figures, 7 tables, 40 sources

Graphic part: 8 posters.

INTERNET OF THINGS, DEEP LEARNING, INTRUSION DETECTION, ANOMALY, NETWORKS.

The bachelor's qualification thesis is dedicated to the development and analysis of existing intrusion detection systems, specifically focusing on IoT devices.

The study analyzes anomaly detection methods in Internet of Things (IoT) device traffic to develop an effective intrusion detection system. The key security threats to IoT devices are identified, and modern approaches to monitoring and anomaly analysis in network connections are explored. As a result, a conceptual model of an intrusion detection system based on machine learning methods and behavioral analysis has been developed. A set of recommendations for implementing the proposed system to enhance IoT infrastructure cybersecurity has been provided.

02. 06.2025

ЗМІСТ

ВСТУП

Стрімкий розвиток інформаційних технологій та поширення пристроїв Інтернету речей (IoT) є однією з ключових тенденцій сучасної цифрової трансформації. Завдяки мільярдам підключених датчиків, виконавчих механізмів та обчислювальних пристроїв, системи IoT інтегруються в багато сфер життя - від промисловості та енергетики до охорони здоров'я та «розумних» будинків. Незважаючи на величезний потенціал і численні переваги, пов'язані з автоматизацією, оптимізацією процесів і збором великих обсягів даних, розширення екосистеми IoT також створює значні нові виклики кібербезпеці.

Пристрої IoT часто мають обмежені обчислювальні ресурси, є менш енергоефективними і працюють у дуже неоднорідному середовищі, що робить їх вразливими до різноманітних кібератак. Традиційні методи захисту, розроблені для корпоративних мереж і персональних комп'ютерів, не завжди ефективно застосовуються до специфіки середовищ IoT. Це призвело до зростання кількості успішних кібератак, що призводять до порушення конфіденційності, цілісності та доступності даних, значних фінансових втрат і збоїв у роботі критично важливих систем.

Одним з найбільш перспективних напрямків безпеки IoT є розробка та впровадження систем виявлення вторгнень на основі аномалій (IDS). На відміну від сигнатурних систем виявлення вторгнень (СВВ), які спираються на відомі шаблони атак, аномальні СВВ можуть виявляти невідомі загрози («нульового дня») та аномальну поведінку, що особливо важливо в динамічному та непередбачуваному середовищі IoT.

Актуальність теми дослідження, зростаюча кількість і складність кібератак на пристрої Інтернету речей та специфічні особливості їх поведінки (обмежені ресурси, різноманітні протоколи, великі обсяги даних) вимагають розробки нових ефективних підходів до виявлення вторгнень. Існуючі рішення часто не забезпечують достатнього рівня захисту або є занадто ресурсоємними для реалізації в пристроях IoT. Тому розробка і дослідження СВВ на основі аномалій,

специфічних для IoT, є важливим завданням, яке може допомогти поліпшити

загальну кібербезпеку інфраструктури IoT.

Мета даної кваліфікаційної роботи є розробка та дослідження СВВ на основі аномалій для IoT-пристроїв. Для досягнення поставленої мети були поставлені наступні задачі дослідження:

- проаналізувати сучасний стан та особливості розвитку систем IoT, існуючі загрози та типові атаки на пристрої IoT;

- дослідити принципи роботи та архітектуру систем виявлення вторгнень на основі аномалій, а також методи машинного навчання та статистичного аналізу, що використовуються при їх побудові;

- розробити архітектуру та обрати найкращий алгоритм для ефективного виявлення аномалій у мережевому трафіку та поведінці з урахуванням обмежених ресурсів iot-пристроїв;

- реалізувати ключові компоненти розробленої системи виявлення вторгнень у програмному забезпеченні;

- провести експериментальне дослідження ефективності розробленої системи з використанням спеціальних наборів даних трафіку iot та оцінити її продуктивність.

Об'єкт дослідження Процеси взаємодії та функціонування пристроїв в середовищі Інтернету речей.

Особливістю даного дослідження є його орієнтація на специфіку середовища IoT, що вимагає врахування обмежень на ресурси та енергоспоживання конкретних пристроїв, а також особливостей мережевого трафіку. Розробка системи базується на застосуванні сучасних методів машинного навчання, оптимізованих для ефективного виявлення аномалій у поведінці IoT-систем. Практична ефективність запропонованого рішення була підтверджена експериментальними дослідженнями, проведеними на спеціальних наборах даних трафіку IoT. Крім того, запропонована архітектура системи має модульну структуру, що дає можливість подальшого вдосконалення та інтеграції нових методів аналізу для протидії еволюціонуючим кіберзагрозам.

1 АНАЛІЗ АТАК НА ІОТ ПРИСТРОЇ

1.1 Типи атак

З поширенням IoT виникають нові ризики для кібербезпеки. IoT-пристрої мають обмежені ресурси, що ускладнює впровадження ефективних засобів захисту. Крім того, різноманіття архітектур і протоколів підвищує ймовірність виникнення вразливостей. Однією з найпоширеніших загроз є атаки типу DDoS, що спрямовані на виведення з ладу пристроїв або сервісів через перевантаження трафіком [1].

DDoS-атаки бувають об’ємними, протокольними та прикладного рівня. Об’ємні атаки передбачають надсилання масового трафіку, що вичерпує пропускну здатність мережі жертви. Протокольні атаки, наприклад SYN-флуд або UDP-флуд, експлуатують слабкості протоколів TCP, UDP, ICMP і викликають перевантаження системних ресурсів. Атаки прикладного рівня, зокрема HTTP флуд, націлені на виснаження серверів через численні запити до вебдодатків [3, 5]. Економічні наслідки DDoS-атак можуть бути значними: зупинка бізнес процесів, втрата доходу, зниження продуктивності персоналу та витрати на відновлення. За даними Ponemon Institute, середній збиток від такої атаки може досягати десятків тисяч доларів на годину [2]. DDoS також використовуються у кібервійні: державні чи пов’язані з ними угруповання намагаються паралізувати критичну інфраструктуру противника.

Окрім DDoS, суттєву загрозу становлять експлойти — програмні або апаратні інструменти, що використовують вразливості для проникнення в систему. Вони класифікуються за способом використання: локальні (наприклад, Dirty Cow, CVE-2016-5195) дозволяють отримати підвищені привілеї.

Також існують експлойти, спрямовані на клієнтські додатки (браузери, медіаплеєри), мережеві сервіси (Heartbleed), вебзастосунки (SQL-ін’єкції, XSS) і навіть апаратне забезпечення (Meltdown, Spectre). Для виявлення та нейтралізації таких загроз використовують СВВ (IDS/IPS), регулярні оновлення, багаторівневий контроль доступу та аналіз поведінки шкідливого ПЗ [4, 7].

У середовищі IoT, де переважають слабозахищені та ресурсно обмежені пристрої, критичні уразливості стають серйозною загрозою стабільній роботі як

окремих компонентів, так і всієї мережі. Нижче було розглянуто види таких атак.

Heartbleed є помилкою в реалізації бібліотеки OpenSSL, яка широко використовується для забезпечення захищеної передачі даних через протокол TLS/SSL (рис.1.1)[4,7]. Через неправильну реалізацію механізму "heartbeat" (перевірки зв'язку) стала можливою експлуатація помилки, що дозволяє зчитування до 64 КБ пам’яті сервера стороннім користувачем.

Рисунок 1.1 – Стандартний SSL-профіль у конфігурації F5 BIG-IP

Це може призвести до витоку важливих даних, зокрема закритих ключів, логінів, паролів, а також приватних повідомлень (рис. 1.2)[10].

Рисунок 1.2 – Апаратний захист SSL-ключів

Компанія F5 Networks розробила низку захисних механізмів для протидії Heartbleed. Зокрема, контролер доставки додатків F5 BIG-IP забезпечує захист шляхом аналізу SSL-сесій. Він інтегрує апаратне шифрування ключів (через HSM-модулі), а також дозволяє реалізацію спеціальних iRules — скриптів для

фільтрації шкідливого трафіку, навіть без використання апаратного прискорення. Такі рішення є дієвими в захисті високонавантажених IoT-сервісів, що працюють через SSL [10].

Meltdown — апаратна уразливість, яка дозволяє неавторизованим процесам читати захищені області оперативної пам’яті. Основу цієї атаки становить спекулятивне виконання інструкцій — механізм, за допомогою якого сучасні процесори намагаються передбачити результати обчислень для підвищення продуктивності. Однак уразливість у логіці доступу до пам’яті дозволяє отримати конфіденційні дані з кешу, які мали бути недоступними. У першу чергу атака зачіпає процесори Intel, але частково вразливими є також деякі моделі AMD та ARM [11].

Spectre використовує схожий принцип, але замість порушення меж доступу між ядром і користувацьким простором, ця атака маніпулює логікою передбачення переходів у коді, змушуючи процесор виконувати інструкції, які розкривають чутливі дані інших процесів. Уразливості Spectre класифіковані за ідентифікаторами CVE-2017-5753 та CVE-2017-5715 і належать до класу атак через сторонні канали, що ускладнює їх виявлення та запобігання [11].

У відповідь на ризики, Microsoft випустила патчі навіть для систем, підтримка яких була завершена: Windows XP, Windows Server 2003 та інші. Американське Агентство з кібербезпеки (CISA) підтвердило можливість створення експлойту BlueKeep у лабораторних умовах і попередило про потенційну масову атаку на державні інфраструктури [12].

MITM-атаки (рис 1.3)[13] становлять одну з найпоширеніших загроз у контексті захисту переданих даних. Вони передбачають несанкціоноване втручання зловмисника у комунікацію між клієнтом і сервером із метою перехоплення, зміни або підміни інформації. Особливо вразливими є користувачі публічних мереж Wi-Fi, а також IoT-пристрої з незахищеними або стандартними налаштуваннями підключення.

Сценарії MITM можуть включати: підміну IP-адрес (DNS/ARP-спуфінг), зниження рівня шифрування (SSL stripping), підміну HTTPS-сертифікатів або

навіть зараження клієнтських пристроїв троянами, що змінюють відображення вмісту сайтів (MITB-атаки). Уражені користувачі ризикують втратити облікові дані, фінансову інформацію або стати об’єктом фішингових кампаній.

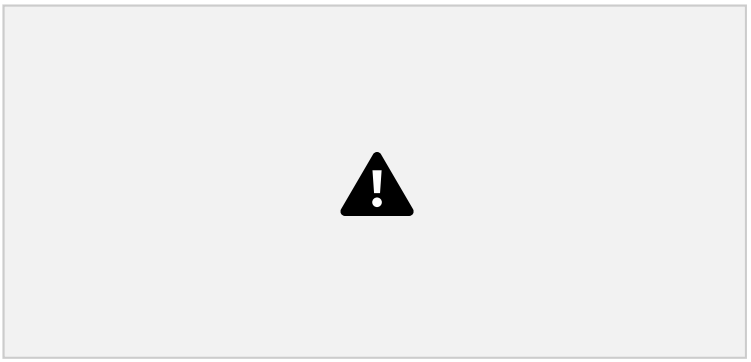

Рисунок 1.3 – Приклад атаки «Людина посередині»

Для захисту від MITM-атак користувачам рекомендується використовувати VPN, уникати відкритих мереж, перевіряти справжність сертифікатів, а також оновлювати прошивку маршрутизаторів. У корпоративному середовищі актуальним є застосування мережевих фільтрів, сегментації та моніторингу активності в режимі реального часу[13].

Брутфорс — це атака, що базується на переборі всіх можливих комбінацій для виявлення правильного пароля чи ключа. У контексті IoT та промислових систем, ця методика часто використовується для доступу до інтерфейсів управління, панелей адміністрування або пристроїв з відкритими портами (наприклад, Telnet або SSH). Особливу загрозу становлять пристрої, у яких залишені паролі за замовчуванням або реалізована слабка політика автентифікації. Успішна атака може призвести до повного контролю над пристроєм, модифікації його прошивки, викрадення даних або залучення в ботнет. Для протидії необхідно впроваджувати механізми обмеження кількості спроб входу, двофакторну автентифікацію та зміну стандартних облікових записів [13].

Брутфорс-атака базується на ресурсномісткому принципі перевірки кожної можливої комбінації символів, поки не буде знайдено правильний варіант.

Спершу зловмисник вибирає ціль, яку хоче зламати, наприклад, обліковий запис, веб-сайт або зашифрований файл, і досліджує формат ключа, його довжину та допустимі символи. Далі він обирає програмне забезпечення для атаки, яке генерує всі варіанти ключа в заданому діапазоні та порівнює хеші цих варіантів із

хешем цілі, якщо він доступний. Коли програма знаходить збіг, атака завершується, і зловмисник отримує доступ до цілі. Брутфорс-атаки часто спрямовані на паролі до облікових записів, шифри для захисту даних, PIN-коди для карток чи телефонів, шифрувальні ключі для доступу до зашифрованих файлів, а також хеші для відновлення вихідних даних. Простий перебір (Brute Force Attack) - цей тип атаки полягає в повному переборі всіх можливих комбінацій символів, починаючи від найпростіших і завершуючи найбільш складними. Наприклад, для злому пароля з 4 цифр, порушник почне з “0000” і закінчить на “9999”, перевіряючи кожну можливу комбінацію. Простий перебір є базовим, але дуже ресурсомістким методом, особливо коли йдеться про довгі або складні паролі.

Атаки за словником (Dictionary Attack) - цей метод використовує заздалегідь підготовлений список слів або фраз, який називається "словником", для перевірки можливих паролів або інших секретних даних. Це більш ефективний підхід, ніж простий перебір, оскільки він орієнтується на ймовірні варіанти паролів, які користувачі найчастіше обирають.

Гібридні атаки (Hybrid Attacks) - цей тип атаки поєднує простий перебір і використання словника. Спочатку порушник пробує паролі з "словника", а якщо це не приносить результату, переходить до повного перебору всіх можливих варіантів. Такий підхід дозволяє заощадити ресурси, оскільки спочатку перевіряються найбільш ймовірні варіанти, а потім – менш вірогідні.

Атаки на багатокористувацькі системи (Credential Stuffing) - цей метод включає використання вкрадених облікових даних (логінів і паролів), отриманих через попередні витоки інформації, для спроби входу в різні акаунти на численних сайтах або сервісах. Порушник автоматизує процес введення цих облікових даних на багатьох платформах, використовуючи одні й ті самі комбінації логінів і

паролів.

Є пару інструментів для подібних атак. Hydra, John the Ripper та Medusa — потужні інструменти для проведення брутфорс-атак, кожен з яких має свої особливості. Hydra є багатопотоковим інструментом, який здійснює перебір

паролів для різних протоколів аутентифікації, таких як SSH, FTP, HTTP і Telnet. Вона підтримує як однопотоковий, так і багатопотоковий режими, що дозволяє швидко виконувати атаки, а також має можливість налаштування параметрів, таких як кількість потоків і затримки між спробами. John the Ripper, у свою чергу, є одним з найпопулярніших інструментів для злому паролів, що підтримує різні формати хешування. Він використовує методи перебору, атаки за словником і комбіновані атаки, що робить його універсальним для злому паролів на різних платформах.

Medusa також є багатопотоковим інструментом для брутфорс-атак, здатним працювати з різними протоколами аутентифікації. Як і Hydra, Medusa надсилає запити з різними комбінаціями імен користувачів і паролів, перевіряючи відповіді серверів. Вона забезпечує високу продуктивність, що робить її ефективною для атак на великі мережі та сервери.

Використання складних паролів - є важливим елементом захисту від брутфорс-атак. Паролі повинні бути довгими (не менше 12 символів) і містити різноманітні символи, включаючи великі та малі літери, цифри та спеціальні знаки. Варто уникати простих і передбачуваних паролів, таких як "123456" або "password" [36-37].

Двофакторна аутентифікація (2FA) - додає додатковий рівень захисту. Окрім пароля, користувач повинен підтвердити свою особистість, надавши одноразовий код, надісланий через SMS або згенерований спеціальним додатком для автентифікації [38].

Обмеження на кількість спроб входу - допомагає запобігти успішним брутфорс-атакам. Після певної кількості неправильних спроб система може тимчасово заблокувати доступ або обмежити введення пароля, що ускладнює автоматичні атаки [39].

Моніторинг і реєстрація активності входу - дозволяє виявляти підозрілі спроби, зокрема брутфорс-атаки. Аналіз журналів допомагає швидко реагувати на потенційні загрози та вживати необхідні заходи безпеки.

Оновлення програмного забезпечення та безпекових патчів - є важливим заходом для закриття відомих вразливостей, які можуть бути використані в

брутфорс-атаках. Оновлення допомагає зміцнити систему, роблячи її менш уразливою до атак [40].

Застосування цих заходів у комплексі суттєво підвищує безпеку акаунтів та систем, зменшуючи ймовірність успішних брутфорс-атак [14].

1.2 Аномалії мережевого трафіку

Виявлення аномалій передбачає виявлення відмінностей, відхилень і винятків із норми в наборі даних. Це іноді називають виявленням викидів (тобто переглядом набору даних для виявлення будь-яких віддалених або незвичних точок даних, груп даних або діяльності). Наприклад, компанії, що займаються кредитними картками, збирають дані про все, що ми купуємо, включаючи суму грошей, яку ми витрачаємо, де ми їх витрачаємо, на що ми їх витрачаємо, як часто ми робимо покупки тощо

Виявлення аномалій робить ці дані не тільки корисними, але й потужними. Це пов’язано з тим, що алгоритми виявлення аномалій аналізують усі наведені вище дані, щоб виявити шахрайську діяльність кредитної картки протягом кількох секунд після здійснення транзакції.

Звичайно, існує багато застосувань для виявлення аномалій, ніж перераховані вище. Вирішальним моментом є той факт, що виявлення аномалій стає все більш важливим і що це дозволяє використовувати дані у спосіб, яким це не було раніше [15].

Загалом методи такого виявлення поділяються на чотири основні категорії: статистичні, засновані на часових рядах, ескізні методи та підходи на базі

машинного навчання. Останні дві категорії зазвичай вимагають маркованих даних для попереднього навчання моделей, що може бути ресурсомістким процесом, проте забезпечує високу точність виявлення. Натомість статистичні методи та методи часових рядів здебільшого працюють у неконтрольованому режимі, не потребуючи попередньої розмітки даних.

Виявлення аномалій у мережевому трафіку є важливим аспектом

забезпечення інформаційної безпеки та своєчасного виявлення кіберзагроз. Сучасні підходи до вирішення цього завдання включають як класичні методи кластеризації, так і гібридні моделі з використанням глибокого навчання. Один з найпоширеніших підходів використовує алгоритм K-середніх для кластеризації ключового трафіку, який дозволяє виявляти нетипові патерни, що можуть свідчити про потенційні атаки або збої в системі.

Інтеграція K-середніх та множинних згорткових нейронних мереж (Multi CNN) дозволяє автоматично виявляти характерні ознаки аномалій в межах кожного кластера, підвищуючи точність та зменшуючи кількість хибних спрацьовувань [16]. Подальші вдосконалення алгоритмів кластеризації, такі як заміна евклідової метрики на косинусну або використання коаліційного навчання, можуть адаптуватися до розподілених даних високої розмірності, зберігаючи при цьому конфіденційність [17].

Більш складний підхід передбачає багатовекторний аналіз мережевого трафіку, який розглядає інформацію про трафік з різних точок зору, таких як час, тип з'єднання та взаємодія між вузлами. Цей підхід дозволяє краще виявляти складні або замасковані аномалії, особливо в контексті зашифрованих з'єднань [18]. У транспортних системах методи кластеризації (K-середні, ієрархічна кластеризація) ефективно використовуються для виявлення збоїв датчиків, перебоїв у русі або штучно створених заторів [19].

Крім того, в останніх дослідженнях все частіше використовуються моделі на основі графових нейронних мереж (GNN) для виявлення аномалій без попередньої розмітки даних за допомогою самонавчання [20]. Існують також підходи на основі багатомасштабних залишкових класифікаторів, які аналізують

трафік на різних рівнях деталізації та покращують здатність системи виявляти нові та невідомі типи атак [21].

Використання K-середніх є основним методом виявлення кластерів з нетиповою поведінкою в простіших системах виявлення аномалій [22]. Водночас системи, що поєднують K-середні з сучасними механізмами класифікації, постійно підвищують свою ефективність [23].

Кожен із цих підходів має свої переваги й обмеження залежно від потреб

системи, обсягу трафіку та вимог до точності чи швидкості. У поєднанні вони дозволяють створювати потужні гібридні системи виявлення аномалій, здатні адаптуватися до різних умов і загроз.

1.3 Аналіз існуючих рішень

В магістральних мережах трафік на маршрутизаторах дуже великий і постійно змінюється, в той час як аномальний трафік невеликий в порівнянні з нормальним трафіком або змінами в нормальному трафіку. Основною метою виявлення аномалій є виявлення відносно невеликого аномального трафіку серед відносно великого фонового трафіку. Тому швидке і точне виявлення аномалій трафіку є однією з умов забезпечення безпеки ефективної роботи мережі.

Методи виявлення аномалій поділяються на дві категорії: виявлення на основі сигнатур та статистичне виявлення. Сигнатурні методи виявляють аномалії на основі відомих ознак або «сигнатур». Недоліком цього методу є те, що він може виявляти апріорі відомі типи аномалій. Тому цей метод не можна використовувати для виявлення невідомих аномалій. Статистичний метод виявлення аномалій трафіку використовує інший підхід, коли визначається «нормальна» мережева активність, а відхилення від норми ідентифікуються як аномалії. Перевага цього підходу полягає в тому, що він не вимагає попереднього знання характеристик аномалії і тому ефективний для виявлення невідомих аномалій і змін в існуючих відомих аномаліях. Один із статистичних методів

виявлення ґрунтується на аналізі малих хвиль.

Цей метод є одним з найсучасніших інструментів моделювання, що використовується для аналізу як нестаціонарних, так і довгострокових взаємозв'язків і властивостей рядів даних. Малі вейвлет-перетворення використовуються для аналізу трафіку і, на відміну від перетворень Фур'є, можуть виявити великомасштабні особливості як часової, так і частотної динаміки.

Методи виявлення аномалій у мережевому трафіку охоплюють широкий

спектр підходів, кожен з яких має власні переваги, обмеження та рівень гнучкості. Одним із найбільш динамічних і адаптивних є машинне навчання, що дозволяє системам самостійно навчатися на історичних даних та виявляти нетипову поведінку без необхідності точного попереднього визначення ознак. Класифікація, кластеризація та регресійні методи машинного навчання здатні обробляти великі масиви даних і адаптуватися до змін у структурі трафіку. Вони особливо ефективні в умовах динамічного середовища, де традиційні методи не справляються.

Нейронні мережі, як глибша реалізація машинного навчання, мають високу гнучкість завдяки здатності моделювати складні нелінійні залежності між параметрами трафіку. Наприклад, багатошарові перцептрони (MLP), рекурентні мережі (RNN) або згорткові мережі (CNN) можуть бути адаптовані для аналізу як потокових, так і структурованих даних. Вони використовуються як у задачах класифікації, так і в задачах виявлення вторгнень у реальному часі. Автокодери, що є різновидом нейронних мереж, виконують навчання у ненаглядовому режимі та добре підходять для виявлення рідкісних чи нових типів аномалій завдяки здатності виявляти відхилення від реконструйованих шаблонів нормального трафіку.

Сигнатурний аналіз, навпаки, є менш гнучким, але ефективним методом виявлення вже відомих загроз. Його принцип базується на зіставленні вхідного трафіку зі списком заздалегідь визначених шаблонів (сигнатур) атак. Такий підхід дозволяє з високою точністю і швидкістю виявляти відомі атаки, проте він неспроможний ідентифікувати нові або модифіковані загрози, що робить його

вразливим до невідомих атак і ботнетів нового покоління.

Генетичні алгоритми демонструють високу адаптивність завдяки здатності до оптимізації параметрів моделі в умовах складного пошукового простору. Їх часто використовують у поєднанні з іншими алгоритмами (наприклад, SVM або нейронними мережами) для вибору найінформативніших ознак або для тонкого налаштування ваг моделі. Генетичні підходи добре зарекомендували себе в задачах, де складно сформулювати точні правила або обчислити градієнти, однак вони можуть вимагати значних обчислювальних ресурсів.

Ансамблі моделей поєднують результати кількох різних моделей для досягнення більшої точності та надійності. Наприклад, такі методи як випадковий ліс (Random Forest), градієнтне бустування (XGBoost) або навіть гібридні ансамблі, що включають класифікатори та автокодери, здатні охопити різні аспекти аномалій у мережевому трафіку. Вони мають високу стійкість до шуму і

переобучення, що робить їх ефективним інструментом у складних середовищах з великою кількістю змінних.

Таким чином, вибір методу виявлення тісно пов'язаний з контекстом його застосування: реальний час, обсяг даних, необхідна точність, наявність навчальних даних тощо. Гнучкі методи, як-от нейронні мережі, автокодери, генетичні алгоритми та ансамблі, дозволяють будувати адаптивні системи, здатні протистояти сучасним кіберзагрозам і мінімізувати хибнопозитивні спрацьовування.

Було розглянуто декілька методів та рішень за для реалізації виявлень аномалій у пристроїв IoT, які було описано нижче. В цьому переліку розглянуто різні підходи, які було розглянуто попередньо.

Одне з існуючих рішень, яке реалізується на машинній технології для виявлення аномалій було запропоновано в статті [24] на основі гібридного підходу з використанням кластеризації k-середніх. Однією з переваг цієї моделі є висока точність класифікації аномалій, досягнута завдяки етапу попередньої кластеризації. Алгоритм K-середніх дозволяє згрупувати трафік за схожими характеристиками, зменшуючи розмірність простору ознак і фокусуючи увагу

моделі на найбільш релевантних кластерах. Це, у свою чергу, допомагає SMO алгоритму (реалізованому як частина методу опорних векторів – SVM) ефективніше розділяти аномальні та нормальні зразки. Таким чином, комбінація двох етапів – кластеризації та класифікації – покращує продуктивність системи в складних умовах мережевого середовища. Проте дана система має й певні обмеження. По-перше, алгоритм K-середніх чутливий до вибору початкових центрів кластерів і кількості кластерів (параметр k), що може суттєво впливати на результати класифікації. По-друге, цей підхід менш ефективний у виявленні

нових або рідкісних типів атак, які не мають чітко виражених кластерних характеристик або мають незначну представленість у навчальних даних. Крім того, SMO є досить обчислювально затратним, особливо при роботі з великими обсягами даних, що може стати перешкодою для використання моделі в системах, які вимагають обробки в реальному часі.

Модель Gadal et al. [24] демонструє ефективний і точний підхід до виявлення аномалій на основі машинного навчання, проте її практична реалізація вимагає ретельного налаштування та оптимізації параметрів, а також врахування обчислювальних ресурсів при роботі з великими потоками мережевого трафіку.

У статті Thankappan, Rifà-Pous та Garrigues [25] представлено сигнатурно орієнтовану систему виявлення вторгнень (IDS), розроблену спеціально для виявлення міжканальних атак типу "людина посередині" (MITM) у захищених Wi-Fi мережах. Запропонована система використовує набір попередньо визначених сигнатур атак, що дозволяє точно визначати характерні шаблони зловмисної активності в середовищі багатоканального зв’язку. Рішення реалізовано у вигляді гнучкої архітектури, здатної виявляти навіть складні атаки на рівні кадрів Wi-Fi-протоколу. Однією з основних переваг цієї системи є висока точність виявлення відомих атак. Завдяки використанню сигнатур, система може точно і швидко реагувати на визначені шаблони зловмисної поведінки, мінімізуючи кількість хибних спрацювань. Крім того, автори реалізували підтримку багатоканального моніторингу, що є важливою інновацією для виявлення MITM-атак, які часто експлуатують перемикання між каналами для

уникнення фіксації. Система також демонструє хорошу масштабованість і може бути розгорнута на реальних мережевих середовищах без суттєвих змін до інфраструктури. Головним недоліком є обмежена здатність до виявлення нових, раніше невідомих атак. Як і всі сигнатурні IDS, система не здатна розпізнати аномальні дії, які не мають чіткої відповідності наявним сигнатурам. Це знижує її гнучкість у ситуаціях, коли зловмисники використовують модифіковані або нові методи атак, які ще не були каталогізовані. Ще однією проблемою є потреба в постійному оновленні сигнатурної бази, щоб підтримувати актуальність і ефективність захисту, що може вимагати додаткових зусиль з боку

адміністраторів безпеки.

Підхід Thankappan et al. [25] є високоефективним рішенням для конкретного класу загроз у Wi-Fi-мережах, особливо тих, що ґрунтуються на MITM-атаках, однак він вимагає постійної актуалізації та не є універсальним методом для виявлення широкого спектру загроз.

У статті Корнієнка, Герасіної, Тимофєєва, Сафарова та Ковальової [26] розглянуто проблему ідентифікації та прогнозування самоподібного трафіку в інформаційно-комунікаційних мережах, що є важливим завданням для підвищення ефективності систем виявлення атак. Автори акцентують увагу на тому, що сучасний мережевий трафік часто має самоподібні властивості (self similarity), що ускладнює його аналіз традиційними методами. Відповідно, для підвищення точності виявлення аномалій, запропоновано враховувати статистичні характеристики самоподібності при моделюванні трафіку. Серед переваг запропонованого підходу — здатність більш точно прогнозувати поведінку трафіку у звичайному стані, що дає змогу своєчасно виявляти відхилення, які можуть свідчити про потенційну атаку. Це особливо актуально для мереж із високим навантаженням або складною топологією, де традиційні IDS можуть давати багато хибнопозитивних результатів. Залучення самоподібних моделей дозволяє покращити обчислювальну стійкість систем виявлення атак та знизити кількість хибних спрацювань . Однак серед недоліків підходу можна назвати складність точного моделювання та параметризації самоподібного

трафіку, що потребує глибоких знань статистичного аналізу та специфічних обчислювальних ресурсів. Крім того, моделі можуть бути чутливими до змін у середовищі функціонування мережі, що ускладнює їх універсальне застосування у різних сценаріях без попереднього налаштування .

Робота [26] робить значний внесок у напрямок вдосконалення систем виявлення атак через адаптацію до специфіки мережевого трафіку, але її ефективність залежить від точності побудованих моделей самоподібності та умов реального застосування.

У роботі Чжана, Полікарпу та Парізіні [27] представлено покращений

детектор аномалій, спеціально розроблений для нелінійних кіберфізичних систем, що піддаються ризику прихованих атак на цілісність даних. На відміну від класичних методів, цей підхід не лише реагує на явні відхилення у поведінці системи, а й здатен виявляти тонко замасковані порушення, які зловмисники можуть ретельно приховувати в межах допустимих значень сигналів. Основна перевага цього рішення полягає в інтеграції адаптивного нелінійного моделювання зі стратегією виявлення аномалій, що дозволяє реагувати на порушення навіть тоді, коли зловмисник намагається уникнути виявлення за допомогою точного підстроювання шкідливого сигналу. Такий підхід є особливо цінним у сучасних промислових і енергетичних кіберфізичних системах, де контроль і безпека повинні бути гарантовані навіть при наявності "розумного" супротивника. Складність цього підходу полягає у необхідності точного опису динаміки системи — будь-яке спрощення або неправильна модель можуть знизити здатність детектора ідентифікувати аномалії. Крім того, практичне впровадження потребує високої обчислювальної потужності та ретельного калібрування, що обмежує застосування в ресурсно-обмежених середовищах.

На відміну від попереднього підходу, орієнтованого на загальну поведінку трафіку в інформаційних мережах, рішення з [27] фокусується на структурованому аналізі фізичних процесів та їх моделюванні, що відкриває нові горизонти в захисті високоточних технологічних систем від цілеспрямованих та прихованих атак.

У статті Волокити та Меленчукова [28] досліджується ефективність використання згорткових нейронних мереж (CNN) для виявлення атак на розподілені обчислювальні системи, що є актуальним напрямком у сфері кібербезпеки. Авторами було запропоновано архітектуру, яка дозволяє автоматично витягувати суттєві ознаки з мережевого трафіку, не вимагаючи ручної обробки чи експертного втручання. Серед переваг такого підходу — висока здатність до узагальнення та адаптації до нових, раніше невідомих загроз. CNN дозволяє ефективно моделювати складні патерни у поведінці мережі, що робить її особливо корисною в умовах, коли дані мають часову або просторову структуру. Крім того, згорткові мережі добре масштабуються для аналізу

великого обсягу трафіку, що є типовим для розподілених систем. Проте слід зазначити й недоліки такого підходу. По-перше, навчання згорткових моделей потребує великих обсягів маркованих даних, що не завжди доступні в реальному середовищі. По-друге, архітектура CNN може бути вразливою до перенавчання, особливо якщо дані не є репрезентативними або мають шум. Також впровадження такої системи в розподілену мережу вимагає високих обчислювальних ресурсів, що може бути критичним для деяких сценаріїв.

Рішення, описане в [28], демонструє значний потенціал у побудові інтелектуальних та гнучких систем виявлення атак, що можуть самостійно адаптуватися до нових форм загроз, проте його ефективність залежить від правильного налаштування, якості даних та ресурсного забезпечення системи.

У статті [29] запропоновано метод виявлення аномалій на основі хаотичних нейронних мереж (ХНМ), який поєднує чутливість хаотичних систем до початкових умов з можливостями навчання нейронних мереж. Цей підхід дозволяє виявляти складні та замасковані атаки, особливо у випадку нелінійних та імовірнісних даних. Перевагами є висока точність, низький рівень помилкових спрацьовувань та ефективність у складних сценаріях. До недоліків можна віднести складність налаштування параметрів, потребу в значних обчислювальних ресурсах і ризик надмірного пристосування при використанні даних недостатньо високої якості. Загалом, підхід є перспективним у динамічних

і неструктурованих середовищах, але вимагає ретельного налаштування і ресурсів для стабільної роботи.

У роботі [30] запропоновано метод виявлення LDDOS-атак з використанням програмно-визначених мереж (SDN) та машинного навчання: оскільки трафік в мережах SDN управляється централізовано, характеристики трафіку можна ефективно збирати та швидко реагувати на аномалії. Мережа SDN - це централізована мережа. Перевагами такого підходу є гнучкість, динамічність і здатність з високою точністю виявляти низькошвидкісні DDoS-атаки, які важко ідентифікувати за допомогою традиційних IDS. Недоліками є висока чутливість моделі ML до параметрів, потреба у високоякісних маркованих даних та обмеження впровадження SDN у типових мережах. Крім того, інтеграція ML в

мережу ускладнює обслуговування і вимагає додаткових ресурсів. Загалом, поєднання аналітичної потужності ML та гнучкості SDN робить цей підхід перспективним та ефективним у протидії сучасним мережевим загрозам.

У статті [31] запропоновано гібридний метод виявлення аномалій мережевого трафіку на основі нечітких нейронних мереж з ваговими коефіцієнтами (WK-FNN). Підхід поєднує переваги нечіткої логіки (робота з неточними даними) та нейронних мереж (самонавчання та узагальнення). WK FNN ефективні для систем реального часу, вони високо адаптивні до змін трафіку, зменшують кількість хибних спрацьовувань та підтримують онлайн-аналіз. Ефективні в системах реального часу, оскільки зменшують кількість хибних спрацьовувань і підтримують аналіз в режимі он-лайн.

Основним недоліком є складна структура моделі, що ускладнює налаштування і вимагає значних обчислювальних ресурсів, особливо для оптимізації ядра і вагових параметрів. Але, WK-FNN є перспективним інструментом для виявлення аномалій у складних мережевих середовищах.

1.4 Постановка задачі

Виявлення аномалій є важливим елементом кібербезпеки і може ідентифікувати відхилення від нормальної поведінки в мережевому трафіку. Існує чотири основні категорії методів: статистичні, часові ряди, ескізні та методи машинного навчання. Найсучасніші підходи поєднують кластеризацію (наприклад, K-середні) і глибоке навчання (Multi-CNN, GNN, багатомасштабні класифікатори) для кращого виявлення складних або зашифрованих аномалій. Гібридні системи виявлення, що поєднують кілька підходів, забезпечують високу точність і адаптивність до нових загроз.

Методи виявлення аномалій поділяються на сигнатурні та статистичні. Сигнатурні точні, але не виявляють нових атак. Статистичні та методи

машинного навчання (ML, нейронні мережі, автокодери, генетичні алгоритми, ансамблі) є більш гнучкими і здатні ідентифікувати нові загрози, проте можуть вимагати значних обчислювальних ресурсів.

Проаналалізувавши дані підходи до виявлення вторгнень на основі аномалій, можемо бачити, що жоден з них не є ідеально складеним та пропрацьованим. У кожного є певні недоліки. І ні один метод не може огорнути та виявити багато різних аномалій пристроїв IoT.

Метою кваліфікаційної роботи розробка та дослідженні системи виявлення вторгнень на основі аномалій, адаптованої для використання у середовищах Інтернету речей. Досягнення цієї мети вимагає послідовного та системного виконання низки дослідницьких та практичних завдань, спрямованих на створення ефективного рішення для забезпечення кібербезпеки IoT інфраструктур в умовах зростаючих кіберзагроз.Для досягнення цієї мети необхідно наступне завдання:

- проаналізувати сучасний стан та особливості розвитку IoT-систем, а також існуючі загрози та типові атаки на пристрої Інтернету речей. А саме провести огляд архітектур та протоколів комунікації IoT-пристроїв, визначити типові вразливості та вектори атак, спрямованих на IoT-системи, здійснити аналіз

існуючих наборів даних IoT-трафіку для виявлення аномалій та вторгнень. - дослідити принципи функціонування та архітектури систем виявлення вторгнень на основі аномалій, а також методи машинного навчання та статистичного аналізу, що застосовуються для їх побудови. А саме вивчити основні категорії аномалійних СВВ , провести порівняльний аналіз алгоритмів машинного навчання, які можуть бути використані для виявлення аномалій, визначити критерії вибору методів з урахуванням обмежених ресурсів IoT пристроїв.

- розробити архітектуру та обрати оптимальні алгоритми для ефективного виявлення аномалій у мережевому трафіку та поведінці IoT-пристроїв з урахуванням їхніх обмежених ресурсів. А саме сформувати концептуальну архітектуру proposed системи виявлення вторгнень для IoT, вибрати оптимальні методи вилучення ознак (feature extraction) з IoT-трафіку, розробити модель

поведінки, що відображає нормальний стан IoT-пристроїв.

- здійснити програмну реалізацію ключових компонентів розробленої системи виявлення вторгнень. А саме розробити модуль збору та попередньої обробки даних IoT-трафіку, імплементувати обрані алгоритми виявлення аномалій, реалізувати модуль сповіщення про виявлені аномалії.

- провести експериментальні дослідження ефективності розробленої системи на спеціалізованих наборах даних IoT-трафіку та оцінити її продуктивність. А саме обрати репрезентативні набори даних IoT-трафіку, що містять як нормальні, так і аномальні зразки, розробити сценарії тестування та метрики для оцінки ефективності (наприклад, точність, повнота, F1-міра, ROC крива, кількість хибних спрацьовувань), провести серію експериментів з різними параметрами та налаштуваннями системи, здійснити аналіз отриманих результатів та сформулювати висновки щодо ефективності та доцільності використання розробленого підходу.

2 ПРОЕКТУВАННЯ СИСТЕМИ ВИЯВЛЕННЯ ВТОРГНЕННЬ НА ОСНОВІ АНОМАЛІЙ ДЛЯ ПРИСТРОЇВ IOT

2.1 Вибір методу виявлення вторгненнь на основі аномалій

У сфері аналізу даних машинне навчання передбачає створення моделей, здатних виконувати завдання на основі аналізу набору ознак — характеристик, що описують об’єкти. Алгоритми машинного навчання застосовуються до широкого кола задач, таких як класифікація, кластеризація чи регресія. Водночас їхня ефективність значною мірою залежить від типу обраних ознак, які впливають на якість навчання моделі та її здатність узагальнювати інформацію. Однією з передових підгалузей машинного навчання є глибоке навчання, що відзначається здатністю працювати зі складними ієрархіями ознак. На відміну від класичних моделей, глибоке навчання використовує багатошарові структури (нейронні мережі), де кожен шар здійснює перетворення вхідних даних,

поступово виділяючи від простих до більш складних залежностей. Кожен нейрон у таких мережах може бути з’єднаний з нейронами інших шарів, а структура з'єднань визначається типом мережі.

Штучна нейронна мережа (ANN) є базовим прикладом такої архітектури. Вона зазвичай складається з трьох основних шарів: вхідного, прихованого та вихідного. В ANN кожен нейрон одного шару з’єднаний з усіма нейронами сусіднього шару, що називається повнозв’язною структурою. Такий підхід дозволяє моделі враховувати всю інформацію, проте потребує значних обчислювальних ресурсів при великій кількості нейронів.

На відміну від цього, згорткові нейронні мережі (CNN) побудовані на принципі локальності зв’язків. Нейрони в CNN взаємодіють лише з невеликою областю попереднього шару, що дає змогу ефективно обробляти просторові або послідовні структури даних, наприклад, зображення або часові ряди. Завдяки згортковим шарам у таких мережах поступово формується представлення даних на вищих рівнях абстракції — від простих патернів до складних характеристик. Така структура забезпечує високу продуктивність при порівняно невеликій

кількості параметрів, що є перевагою для масштабованих систем. Набір даних UNSW-NB15, представлений у 2015 році лабораторією Cyber Range Австралійського центру кібербезпеки (ACCS), широко використовується для досліджень у сфері виявлення вторгнень (IDS). Він містить як нормальний, так і згенерований за допомогою IXIA PerfectStorm аномальний трафік. Для отримання 35 базових ознак з pcap-файлів було використано такі інструменти, як Argus та Bro-IDS, а ще 12 ознак було сформовано шляхом комбінування наявних. Подальшим розвитком цього підходу став набір NF-UNSW-NB15-v2, створений на базі UNSW-NB15 із застосуванням запропонованого стандарту ознак. Для його побудови використано 43 NetFlow-ознаки, які були витягнуті з тих самих pcap-файлів за допомогою nProbe. Завдяки цьому вдалося досягти суттєвого підвищення точності багатокласової класифікації та зменшити час прогнозування. Саме тому в цьому дослідженні було обрано NF-UNSW-NB15-v2 як основний набір даних [32]. Набір охоплює дев’ять категорій атак (рис.2.1), серед яких: Аналіз, Бекдор, DoS, Експлойти, Фазери, Загальний, Розвідка, Шел

код та Хробаки. Проте розподіл між класами є значно незбалансованим: 96.02% становлять безпечні зразки, тоді як атаки — лише незначну частку.

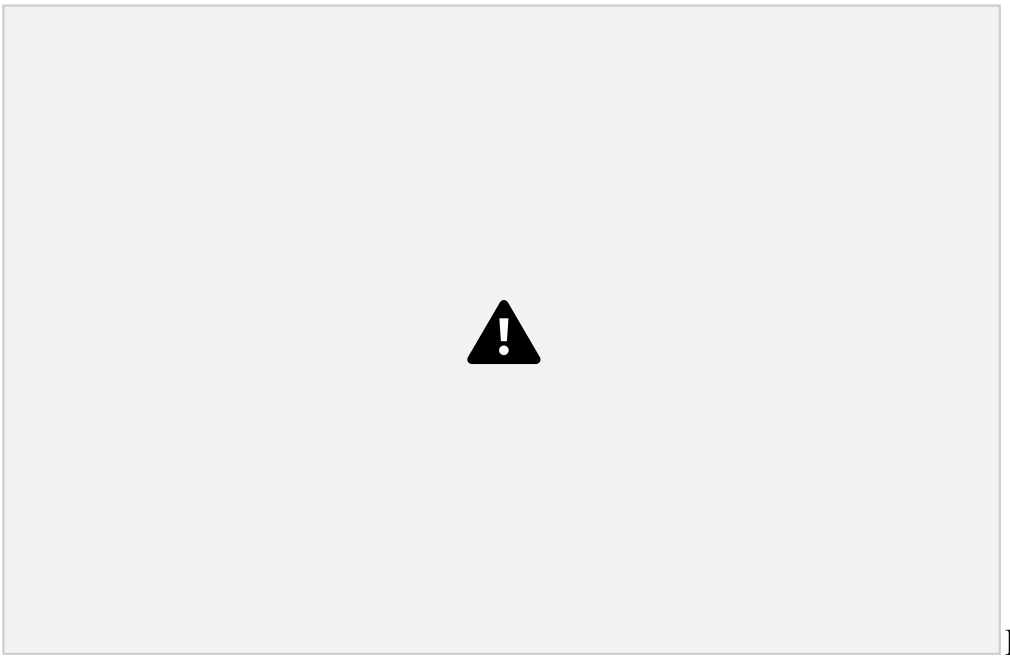

Рисунок 2.1 – Розподіл класів у наборі даних, за винятком безпечних зразків

Наприклад, клас Worm представлений менш ніж 0.01% усіх записів, що суттєво ускладнює навчання моделей і підвищує ризик перенавчання на домінуючому класі. Така ситуація є типовою проблемою у сфері IDS.

Попри ці складнощі, рішення працювати з NF-UNSW-NB15-v2 було зумовлено наявністю уніфікованого стандарту ознак, що забезпечує можливість повторного використання розроблених методів на інших подібних наборах. Це спростило процес побудови моделей та дозволило зосередитись на покращенні результатів у рамках обмежених обчислювальних ресурсів. Хоча набір є меншим за обсягом порівняно з деякими іншими, він містить різноманітні типи атак, що важливо для комплексного навчання моделей виявлення.

Після отримання набору даних NF-UNSW-NB15-v2, наступним етапом стало його попереднє опрацювання для навчання моделей. Цей процес включає п’ять кроків: очищення, перетворення даних, їх розбиття на підвибірки, а також формування вхідних зображень. Ініціально дані надходили у форматі CSV-файлу з 43 ознаками NetFlow, мітками атак і класифікаційним маркером (шкідливий чи ні). Для підвищення якості даних було вилучено записи з некоректними

значеннями (NaN, +inf, -inf), а також шість ознак, що не мали інформативної цінності для задачі класифікації. До таких атрибутів належать: IP-адреси та порти джерела і призначення, а також мінімальний і максимальний TTL потоку.

Після фільтрації було отримано очищений набір із 37 ознак, представлених у числовому форматі (float або int), що зберігаються у таблиці 3.1. Для пришвидшення процесу тренування та тестування моделей вирішено використовувати лише 40% доступного набору даних. Щоб зберегти пропорції кожного класу атак, застосовувалося стратифіковане розбиття. Дані були розділені на три підмножини: навчальну (60%), валідаційну (20%) та тестову (20%). Такий підхід називається 3-way hold-out, і є оптимальним для великих обсягів даних, на відміну від k-кратної перехресної перевірки, яка частіше застосовується для менших наборів. У результаті було сформовано такі підвибірки: 560 927 зразків для навчання, 186 976 — для валідації, і стільки ж — для тестування.

Особливістю реалізованого підходу є трансформація табличних даних у зображення, що дозволяє використовувати ефективність згорткових нейронних мереж (CNN), традиційно успішних у класифікації зображень. Для цього дані спочатку нормалізуються за допомогою min-max масштабування (формула 2.1), а потім ознаки структуруються у формат матриці 7×7. У разі браку ознак використовується доповнення. Наступний крок — це конвертація значень у 8-бітні цілі числа та їх подальше множення на 255. Останній етап — застосування колірної карти, що дозволяє досягти кращої ефективності порівняно з монохромними зображеннями, що було зображено на рисунку 2.2 при виявленні DoS-атак або шкідливого ПЗ [33, 34].

Таким чином, описана методика попередньої обробки дозволила адаптувати класичний NetFlow-набір до форматів, зручних для глибокого навчання, забезпечивши підвищену точність класифікації за допомогою CNN.

$$\text{[illegible]} \quad (2.1)$$

Маємо результат даної формули, що зображано на рисунку 2.2.

Рисунок 2.2 – Процес створення зображення

Представлено таблицю 2.1, яка містить перелік 37 ознак NetFlow, що

використовуються для класифікації мережевого трафіку. Кожен рядок таблиці складається з двох стовпців: назва ознаки (ознака) та її опис українською мовою. Ці ознаки є важливими характеристиками мережевих потоків і широко використовуються в системах виявлення вторгнень, аналізу трафіку, побудові моделей машинного навчання тощо.

Таблиця 2.1 - Остаточні 37 ознак NetFlow, що використовуються для класифікації

Ознака	Опис
1	2
L7_PROTO	Протокол 7-го рівня (числовий)
IN_BYTES	Кількість вхідних байтів
OUT_BYTES	Кількість вихідних байтів
IN_PKTS	Вхідна кількість пакетів
OUT_PKTS	Вихідна кількість пакетів
FLOW_DURATION_MILLISECONDS	Тривалість потоку в мілісекундах
TCP_FLAGS	Сукупність усіх TCP-флагів

CLIENT_TCP_FLAGS	Сукупність усіх клієнтських TCP флагів
SERVER_TCP_FLAGS	Сукупність усіх серверних TCP флагів
DURATION_IN	Тривалість потоку від клієнта до сервера (мс)
DURATION_OUT	Тривалість потоку від клієнта до сервера (мс)
LONGEST_FLOW_PKT	Найдовший пакет (байт) у потоці
SHORTEST_FLOW_PKT	Найкоротший пакет (байт) у потоці
MIN_IP_PKT_LEN	Довжина найменшого IP-пакета потоку, що спостерігався
MAX_IP_PKT_LEN	Довжина найбільшого IP-пакета, зафіксованого у потоці
SRC_TO_DST_SECOND_BYTES	Байт/с від джерела до призначення
DST_TO_SRC_SECOND_BYTES	Байт/с від призначення до джерела
RETRANSMITTED_IN_BYTES	РеТрансмітовані байти TCP-потоку (src->dst)

Кінець таблиці 2.1

1	2
RETRANSMITTED_IN_PKTS	РеТрансмітовані пакети TCP-потоку (src->dst)
RETRANSMITTED_OUT_BYTES	РеТрансмітовані байти TCP-потоку (dst->src)
RETRANSMITTED_OUT_PKTS	РеТрансмітовані пакети TCP-потоку (dst->src)
SRC_TO_DST_AVG_THROUGHPUT	Середня пропускна здатність від джерела до призначення (біт/с)
DST_TO_SRC_AVG_THROUGHPUT	Середня пропускна здатність від призначення до джерела (біт/с)

NUM_PKTS_UP_TO_128_BYTES	Пакети з IP-розміром <= 128 байт
NUM_PKTS_128_TO_256_BYTES	Пакети з IP-розміром > 128 та <= 256 байт
NUM_PKTS_256_TO_512_BYTES	Пакети, розмір IP яких > 256 та <= 512 байт
NUM_PKTS_512_TO_1024_BYTES	Пакети, розмір IP яких > 512 та <= 1024 байти
NUM_PKTS_1024_TO_1514_BYTES	Пакети, розмір IP яких > 1024 та <= 1514 байт
TCP_WIN_MAX_IN	Максимальний розмір TCP вікна (src- >dst)
TCP_WIN_MAX_OUT	Максимальний розмір TCP вікна (dst- >src)
ICMP_TYPE	Тип ICMP * 256 + код ICMP
ICMP_IPV4_TYPE	Тип ICMP
DNS_QUERY_ID	Ідентифікатор транзакції DNS-запиту
DNS_QUERY_TYPE	Тип DNS-запиту (наприклад, 1=A, 2=NS...)
DNS_TTL_ANSWER	TTL першого запису A (за наявності)
FTP_COMMAND_RET_CODE	Код повернення команди клієнта FTP

Перші кілька ознак стосуються загальної інформації про потік: наприклад, PROTOKOL, L7_PROTO, IN_BYTES, OUT_BYTES, які характеризують використовуваний протокол та обсяг переданих байтів і пакетів. Далі наведено ознаки, пов’язані з TCP-прапорами, тривалістю потоків та розмірами пакетів —

CLIENT_TCP_FLAGS, DURATION_IN, LONGEST_FLOW_PKT, тощо. Особливу увагу приділено байтам і пакетам, що передаються між джерелом і призначенням, включно з повторно переданими TCP-байтами та середньою пропускною здатністю. Також зазначені кількісні характеристики IP-пакетів у

різних діапазонах розмірів. Наприкінці таблиці наведено специфічні поля для ICMP, DNS та FTP, що є важливими при аналізі не-TCP трафіку. Ця таблиця є ключовим елементом для розуміння, які саме характеристики потоків NetFlow застосовуються при класифікації мережевих сесій або виявленні аномалій у мережевому трафіку.

2.2 Архітектура моделі

Архітектура моделі OtherNet є незначною модифікацією базової структури згорткової нейронної мережі (ЗНМ) (рис. 2.3а) і призначена для ефективної класифікації мережевого трафіку, зокрема для виявлення шкідливого програмного забезпечення в середовищах IoT та Android. Вона включає три згорткові модулі, за якими слідують два повнозв'язні шари і шар активації softmax, що виконує остаточну класифікацію (рис. 2.3б). Подібні малі моделі вже довели свою ефективність у попередніх дослідженнях [35], що підтверджує доцільність використання компактних архітектур у задачах кібербезпеки.

Кожен згортковий модуль у моделі OtherNet має фільтр 3×3, після якого застосовується шар пакетної нормалізації, що стабілізує навчання. Далі слідує функція активації ReLU, яка додає нелінійність, а також шар max pooling 2×2, що зменшує просторову розмірність вхідних даних, зберігаючи важливі ознаки. У перших двох згорткових шарах кількість каналів послідовно збільшується з 3 до 16, а потім до 32, що дозволяє мережі навчитися дедалі складніших ознак у міру поглиблення.

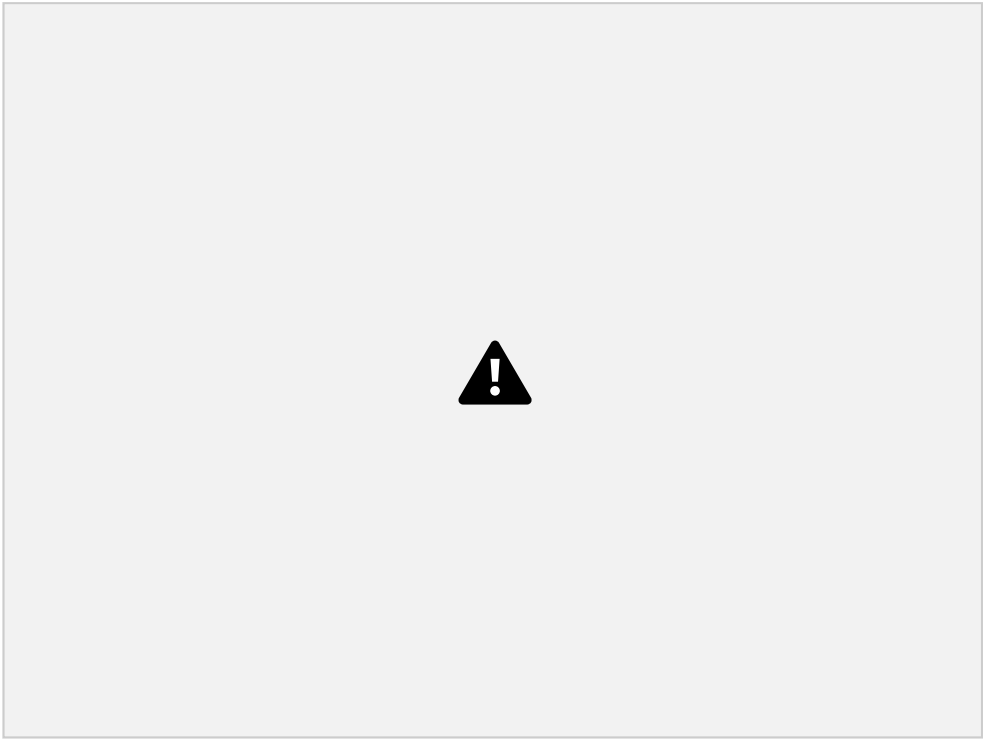

Рисунок 2.3 - Модель OtherNet: а) згортковий модуль для OtherNet, б) структура груп OtherNet

Між третім згортковим модулем і першим повнозв'язним шаром використовується шар вирівнювання (flattening layer), який переводить матрицю ознак у одновимірний вектор. Після першого повнозв'язного шару додається ще один ReLU та шар dropout з імовірністю 0.5 для зменшення перенавчання. Завершується мережа другим повнозв'язним шаром, який виконує остаточне виділення ознак перед класифікацією. Цікаво, що, на відміну від традиційних архітектур, у OtherNet класифікація здійснюється не softmax, а шаром max pooling, що є специфічною особливістю та потенційно може покращити узагальнення моделі. OtherNet є базовою, але функціональною архітектурою, яка завдяки своїй простоті та ефективності служить відправною точкою для порівняння з іншими, більш складними моделями в задачах класифікації шкідливого трафіку.

IoTNet — це друга модель, представлена у дослідженні, яка була розроблена T. Lawrence та L. Zhang спеціально для використання у середовищах з обмеженими обчислювальними ресурсами, таких як IoT-пристрої. Її ключова особливість полягає у заміні стандартних згорток 3×3 на послідовність парних

згорток 1×3 та 3×1, що зменшує кількість параметрів і навантаження на апаратне

забезпечення без втрати продуктивності. Такий підхід дозволяє гнучко адаптувати архітектуру до різних рівнів обчислювальної потужності вбудованих пристроїв.

Архітектура IoTNet побудована з так званих блоків (blocks) і груп блоків, кожна з яких складається з декількох блоків. Як показано на рисунку 2.4a, кожен блок містить згорткові шари, пакетну нормалізацію та ReLU-активатор, а також реалізує skip connection — механізм залишкових зв’язків, який допомагає уникнути затухання градієнта і полегшує навчання глибших мереж. Така структура є натхненною ResNet-архітектурами, де збереження інформації з попередніх шарів покращує якість класифікації.

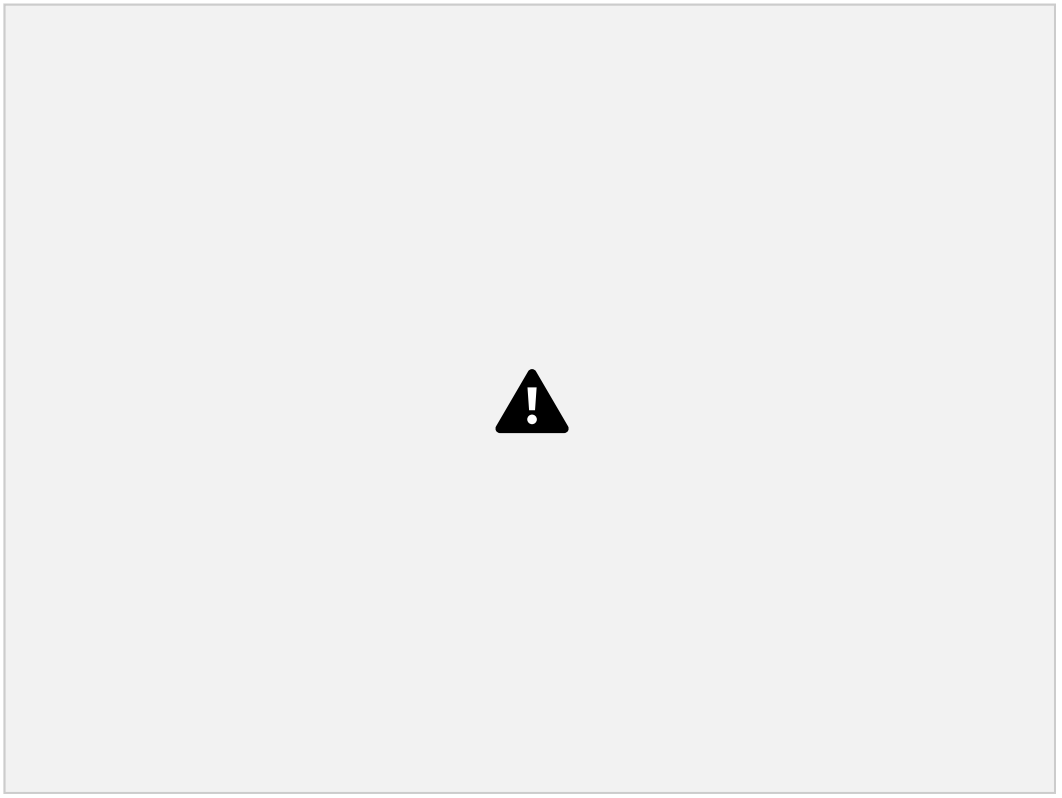

Рисунок 2.4 - Модель IoTNet: a)Згортковий модуль для IoTNet, b) Структура груп IoTNet

Перший згортковий шар 3×3 та перший блок кожної групи визначають ширину мережі — кількість каналів ознак. Для цього застосовується ширинний коефіцієнт k, згідно з методом, запропонованим S. Zagoruyko та N. Komodakis. У

дослідженні використовувались значення n = 3 (кількість блоків у групі) та k = 0.2 (рис 2.4b). Щоб зменшити перенавчання на переважний клас, після кожної

групи блоків у модель додано dropout з імовірністю 0.2. Завершується архітектура повнозв'язним шаром, який відповідає за остаточну класифікацію.

2.3 Вибір гіперпараметрів моделі

Під час проєктування та навчання моделей класифікації особливу увагу було приділено підбору гіперпараметрів, які найбільше впливають на ефективність процесу навчання. Зокрема, було обрано два ключові параметри: швидкість навчання (learning rate) та розмір пакета (batch size). Експерименти проводилися з метою досягнення оптимального балансу між швидкістю збіжності та якістю класифікації, оскільки неправильний вибір цих параметрів може призвести як до перенавчання, так і до повільного або неефективного навчання моделей.

У межах дослідження швидкість навчання варіювалась у межах значень 0.01, 0.001, 0.0001 та 0.00001. Було виявлено, що при значеннях понад 0.0001 моделі фактично не навчалися: вони класифікували всі зразки як один і той самий клас. Навпаки, надто мала швидкість (0.00001) призводила до вкрай повільного

покращення, і навіть після 50 епох результати залишались незадовільними. Оптимальним варіантом виявилася швидкість навчання 0.00005, яка забезпечила поступову збіжність без втрати точності.

Другим важливим параметром був розмір пакета, який впливає як на тривалість навчання епохи, так і на здатність моделі до узагальнення. Згідно з висновками N. S. Keskar та співавторів, надто великі пакети можуть призводити до збіжності у вузькі мінімуми, погіршуючи узагальнення. Під час експериментів тестувалися значення 32, 64, 128 і 256. Найкращий баланс між швидкістю навчання та якістю класифікації був досягнутий при розмірі пакета 128 — цей варіант дозволив значно зменшити час навчання, не погіршуючи результати на

валідації.

Підсумовуючи, у фінальній конфігурації для навчання моделей було обрано

швидкість навчання 0.00005 та розмір пакета 128, що забезпечило найкращу ефективність при роботі з невеликими зображеннями у рамках довідково інформаційної системи.

Дисбаланс у наборі даних є проблемою під час навчання моделей машинного навчання, особливо в задачах класифікації. Коли один або кілька класів домінують за кількістю зразків над іншими, модель має тенденцію переважно правильно класифікувати приклади з мажоритарного класу, ігноруючи менш представлені класи. Така ситуація призводить до зниження загальної точності моделі щодо важливих, але рідкісних класів, що може бути неприпустимим у практичних застосуваннях, таких як виявлення атак або шкідливого програмного забезпечення.

Щоб зменшити вплив дисбалансу класів, існує кілька перевірених методів. Один із них — повторне семплювання, яке включає надлишкову вибірку (oversampling) для збільшення кількості зразків міноритарного класу та/або недостатню вибірку (undersampling) для зменшення кількості зразків мажоритарного класу. Це дозволяє вирівняти розподіл класів у навчальному наборі, покращуючи здатність моделі адекватно навчатися на всіх класах. Проте, такий метод може призвести до перенавчання, особливо при агресивному oversampling.

Другий підхід — навчання з урахуванням вартості (cost-sensitive learning). У цьому випадку моделі призначаються різні штрафи за помилкову класифікацію зразків залежно від їхнього класу. Зазвичай, помилка при класифікації прикладів з міноритарного класу коштує дорожче, ніж помилка для мажоритарного класу. Такий підхід дозволяє моделі приділяти більше уваги менш представленим класам, не змінюючи сам набір даних.

Останній варіант — гібридний підхід, який поєднує повторне семплювання з навчанням із урахуванням вартості. Він дозволяє скористатися перевагами обох методів, мінімізуючи їхні недоліки, і може давати кращі результати, особливо в

умовах сильно вираженого дисбалансу.

У межах цієї дипломної роботи планується оцінити ефективність повторного семплювання, а також двох варіантів навчання з урахуванням

вартості: стандартної функції втрат перехресної ентропії (cross-entropy loss) з вагами класів і збалансованої за класами функції втрат, яка автоматично враховує дисбаланс при розрахунку помилки. Це дозволить обрати оптимальний метод боротьби з дисбалансом для завдання класифікації в системі, що розробляється.

Для усунення дисбалансу в наборі даних під час навчання моделей у цій роботі було застосовано метод повторного семплювання з використанням зваженого випадкового семплера, реалізованого за допомогою бібліотеки PyTorch.

Основна ідея полягає в тому, що кожному зразку в навчальному наборі присвоюється певна вага, яка визначає ймовірність його вибору в процесі формування навчального пакета. Таким чином, часті зразки мають меншу ймовірність потрапити в пакет, тоді як рідкісні — більшу. Визначення ваги для кожного класу здійснюється за допомогою формули 2.2, у якій змінна x позначає відповідний клас. Це математичне вираження дозволяє адаптивно коригувати частоту зразків у залежності від їх належності до того чи іншого класу. У результаті — класи з великою кількістю зразків, такі як "безпечний", вибираються рідше, а класи з меншою кількістю прикладів, наприклад, "хробаки", потрапляють до тренувального набору частіше.

Завдяки цьому підходу досягається більш збалансоване представлення класів під час навчання, що сприяє кращій здатності моделі до узагальнення і точнішій класифікації зразків із міноритарних класів. Таблиця 2.2 містить конкретні значення ваг, використаних у процесі навчання, що дозволяє відтворити експериментальні умови та оцінити вплив повторного семплювання на результати моделі.

(2.2)

Таблиця 2.2 - Ваги класів, використані для зваженого випадкового семплера

Клас	Вага

Аналіз	0.0109
Бекдор	0.0122
Безпечний	0.0000018287
DoS	0.0012
Експлойти	0.0001
Фазери	0.0003
Загальний	0.0014
Розвідка	0.0006
Шел-код	0.0059
Хробак	0.0286

Цей підхід використовує зважену повторну вибірку для покращення балансу між класами в навчальному наборі даних. Основна ідея полягає в зміні ймовірності вибору вибірки: недостатньо представлені класи («черв'як», «бекдор», “шеллкод”) мають високі ваги, а більшість класів (особливо «безпечний») мають надзвичайно низькі ваги. Це дозволяє моделі бачити більше прикладів з рідкісних класів, які зазвичай погано представлені в стандартних випадкових вибірках.

Використання рівняння (2.2) для розрахунку ваг дозволяє адаптивно вирівнювати дисбаланс класів, роблячи вибірку динамічною та обґрунтованою. Такий підхід важливий при навчанні моделей для виявлення аномалій в мережевому трафіку, де важливі типи атак часто становлять невелику частку від загального обсягу даних.

В результаті всі класи більш рівномірно представлені під час навчання, що покращує здатність моделі до узагальнення, особливо при класифікації зразків з невеликої кількості класів. Це зменшує упередженість моделі на користь домінуючих класів і покращує точність, повноту та показники F1 при оцінці ефективності на тестових даних. Таблиця 2.2 з конкретними вагами є важливим елементом для повторення експериментів і порівняння з іншими підходами.

2.4 Висновок до розділу

У другому розділі було детально розроблено систему виявлення вторгнень на основі аномалій, орієнтовану на характеристики пристроїв Інтернету речей (IoT); на основі аналізу характеристик мережевого середовища IoT було обрано підхід, заснований на глибоких згорткових нейронних мережах. Цей вибір був зроблений завдяки здатності ефективно обробляти багатовимірні структуровані дані зі зменшеним мережевим трафіком після відповідних перетворень.

На основі детального аналізу набору даних NF-UNSW-NB15-v2, що охоплює дев'ять категорій атак, генеруються вибірки для навчання, валідації та тестування моделі. Увагу приділено викликам, пов'язаним з дисбалансом класів, що є типовою проблемою при роботі з реальним мережевим трафіком. Для вирішення цієї проблеми запропоновано два підходи: передискретизація та модифіковані функції втрат. Перед навчанням дані були перетворені з табличного формату у візуальний формат зображень 7x7, що дозволило використовувати згорткову архітектуру для класифікації.

В рамках архітектурного проекту було створено дві різні згорткові нейронні мережі: OtherNet та IoTNet. Перша модель була побудована з урахуванням базових вимог щодо точності та швидкості обробки, тоді як друга модель має додатковий згортковий шар, який забезпечує вищу якість класифікації за рахунок збільшення обчислювальної складності. Кожна модель була оптимізована для конкретного випадку використання з урахуванням ресурсних обмежень IoT пристроїв. Крім того, було визначено набір гіперпараметрів, які забезпечують баланс між якістю навчання та стабільністю моделі. Зокрема, обрано оптимізатори, які забезпечують найкращі результати для функції активації, розміру пакета, кількості епох та обраних типів даних. Обидві архітектури використовують оптимізатор Adam, який показав хорошу збіжність у попередніх експериментах.

Результати моделювання чітко показують, що коли класи значно незбалансовані, традиційні методи навчання не дають задовільних результатів, що підтверджує необхідність комбінованого підходу. Розроблена архітектура

враховує обмеженість обчислювальних ресурсів IoT-пристроїв, а також дозволяє робити швидкі висновки на основі вхідного трафіку, що дозволяє розгортати моделі в хмарному середовищі і згодом доставляти їх на кінцеві пристрої.

Отримані результати показують потенціал використання згорткових нейронних мереж для класифікації трафіку в задачах кібербезпеки. Водночас були виявлені обмеження, пов'язані з недостатньою універсальністю моделі для виявлення атак нульового дня. Для покращення результатів у майбутньому рекомендовано тестувати модель з різними конфігураціями гіперпараметрів (у тому числі зі змінною шириною мережі IoT), надалі експериментувати з іншими форматами вхідних зображень та використовувати більш гнучкі структури в процесі перетворення даних.

Таким чином, у цьому розділі було проведено проектування повномасштабної системи виявлення вторгнень, що охоплює всі ключові етапи, від вибору підходу до побудови архітектури моделі та підготовки вхідних даних для навчання. Ці напрацювання створюють міцний фундамент для практичної реалізації системи, описаної в наступному розділі.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМУ НА ОСНОВІ НЕЙРОМЕРЕЖІ

3.1 Оцінка ефективності навчання та валідації нейромережі

Моделі навчалися протягом 100 епох на попередньо розподілених навчальних і валідаційних наборах даних. Результати показують значення втрат для кожної моделі на навчальних і валідаційних даних, точність всіх моделей на валідаційному наборі і середній час навчання для кожної нейронної мережі.

На рисунку 3.1 показано динаміку втрат з часом для навчальної та валідаційної множин. Штриховою лінією показано втрати на навчальній вибірці, а суцільною - на валідаційній вибірці. За цими залежностями втрат можна оцінити, чи є модель перенавченою або недонавченою. Якщо втрати на перевірочній множині перевищують втрати на навчальній множині, то модель є перенавченою, якщо ні - то недонавченою. Як видно з графіків, усі моделі демонструють різке зменшення втрат на початковому етапі, після чого відбувається стабілізація значень приблизно з 20-го періоду.

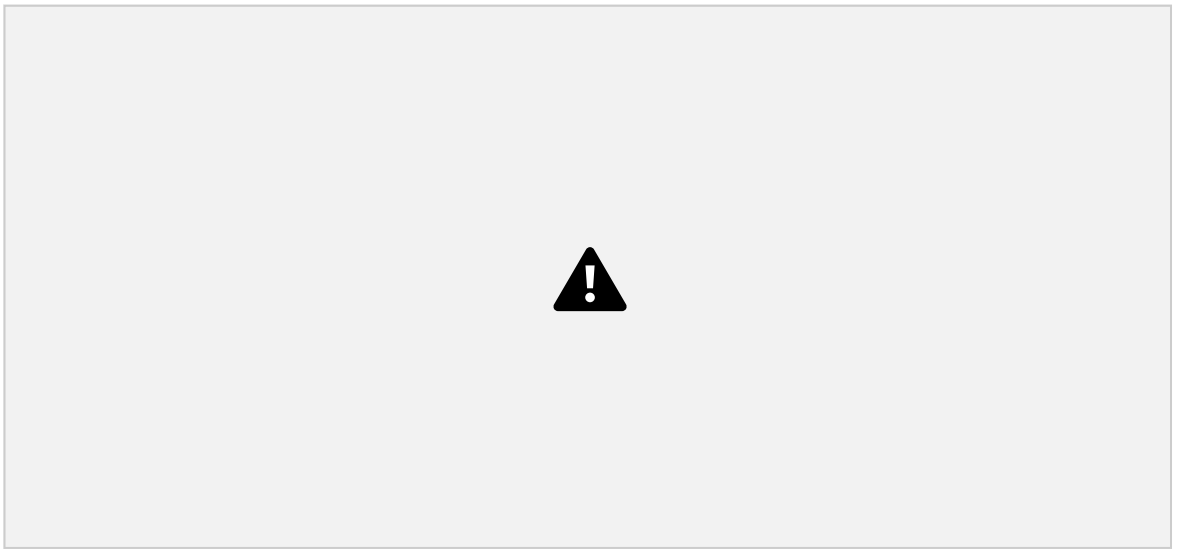

Рисунок 3.1 Навчання та втрати валідації: a) Втрати OtherNet, b) Втрати IoTNet

Зокрема, аналіз рис. 3.1a показує, що лише моделі OtherNet Sampler та OtherNet CELoss демонструють значне зменшення втрат при навчанні. Однак

втрати на валідаційній вибірці зменшуються не так швидко, що може свідчити про перенавчання на навчальних даних. Модель CBLoss, з іншого боку, демонструє значне зменшення втрат порівняно з іншими моделями, що можна

пояснити меншою кількістю категорій, які використовуються для класифікації атак (Рис 3.9 та 3.10).

На рисунку 3.1b показано більшу варіацію втрат між моделями IoTNet порівняно з моделями OtherNet. Це також підтверджується результатами валідації втрат, досягнутих кожною моделлю під час навчання, як показано в таблиці 3.1.

Таблиця 3.1 - Найкращі втрати валідації під час навчання

Модель	Втрата валідації
OtherNet	1.4712
OtherNet Sampler	1.4920
OtherNet CELoss	1.4744
OtherNet CBLoss	0.6882

Модель	Втрата валідації
IoTNet	0.0347
IoTNet Sampler	0.0882
IoTNet CELoss	0.0623
IoTNet CBLoss	0.0100

На рисунку 3.2 показано динаміку точності валідації для всіх моделей. Видно, що точність є більш мінливою порівняно з втратами під час навчання та валідації: як архітектура OtherNet, так і архітектура IoTNet досягли пікової точності близько 99%. Як і у випадку з втратами, основне покращення точності відбувається протягом перших 20 епох навчання.

У моделі OtherNet використання повторної вибірки знизило точність порівняно з іншими підходами, тоді як в IoTNet точність залишилася порівнянною з іншими ефективними підходами. Використання функції втрати балансу класів дало інші результати: в той час як OtherNet мав нестабільну точність, IoTNet показав стабільну точність валідації з варіацією лише 0,01%.

Примітно, що 182 282 (97,49%) з 186 976 зразків належать до класу безпеки. Це означає, що класифікатори, які мають тенденцію до надмірного наближення до домінуючого класу, можуть досягти вищої загальної точності. Таким чином,

для незбалансованих наборів даних точність не обов'язково є відповідним

критерієм для оцінки якості моделі.

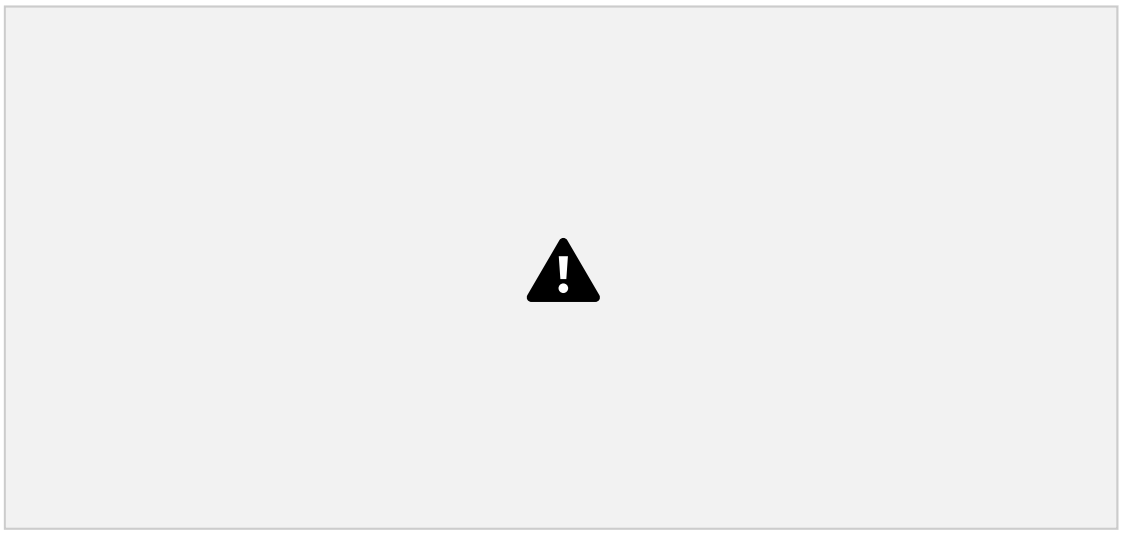

Рисунок 3.2 Точність валідації: a) Точність OtherNet, b) Точність IoTNet

Як показано в таблиці 3.2, час навчання для моделі OtherNet був коротшим, ніж для моделі IoTNet, що є очікуваним результатом, оскільки IoTNet має більше шарів згортки. Це був очікуваний результат, оскільки IoTNet має більшу кількість шарів згортки. Використання зважених функцій втрат CELoss або CBLoss не вплинуло на час навчання для жодної з архітектур. І навпаки, використання передискретизації призвело до збільшення часу через передискретизацію та недодискретизацію.

Таблиця 3.2 - Середній час навчання на епоху

Модель	Час , с
OtherNet (CELoss/CBLoss)	125.96
OtherNet Sampler	143.88
IoTNet (CELoss/CBLoss)	247.35
IoTNet Sampler	251.72

Загалом було підготовлено вісім різних моделей, загальний час навчання

для всіх моделей склав приблизно 42 години. Оскільки чотири моделі могли навчатися одночасно, кожен цикл навчання тривав трохи більше 10 годин.

3.2 Вплив методів обробки дисбалансу на ефективність класифікації аномалій алгоритму

Для перевірки класифікаційної здатності моделей було обрано моделі з найменшою валідаційною втратою, які було використано для класифікації вибірок з тестового набору. Результати класифікації для кожної моделі представлені та оцінені за допомогою матриці розбіжностей.

Багатокласова матриця розбіжностей містить наступні значення для кожного класу α:

- Істинно позитивні (TP): елементи, де фактичний клас α правильно передбачено як α.

- Істинно негативні (TN): елементи, де фактичний клас β правильно передбачено як β.

- Хибно позитивні (FP): елементи, де фактичний клас β помилково передбачено як α.

- Хибно негативні (FN): елементи, де фактичний клас α помилково передбачено як β.

Значення з матриці потім використовуються для оцінки найкращої моделі для кожної нейронної мережі та обчислення загальних метрик, які вимірюють ефективність виявлення аномалій для кожного класу. Точність (Accuracy) - це відношення правильних прогнозів до загальної кількості зразків (формула 3.1), повнота (Precision) - це відношення істинних позитивних результатів до загальної кількості позитивних результатів (формула 3.2), повторюваність (Recall) - це відношення правильних прогнозів серед усіх релевантних зразків (формула 3.3), а оцінка F1 (Score) - це середнє гармонійне значення точності та повноти (формула 3.4).

(3.1)

(3.2)

□ ○ (3.3)

□□□□□□o□□□ ○ □□□□□□ (3.4)

На рисунку 3.3 показано матрицю невідповідностей OtherNet, навчену за допомогою стандартної функції втрати перехресної ентропії. Видно, що всі зразки були класифіковані на три великі категорії, що є очікуваним результатом при навчанні на незбалансованому наборі даних.

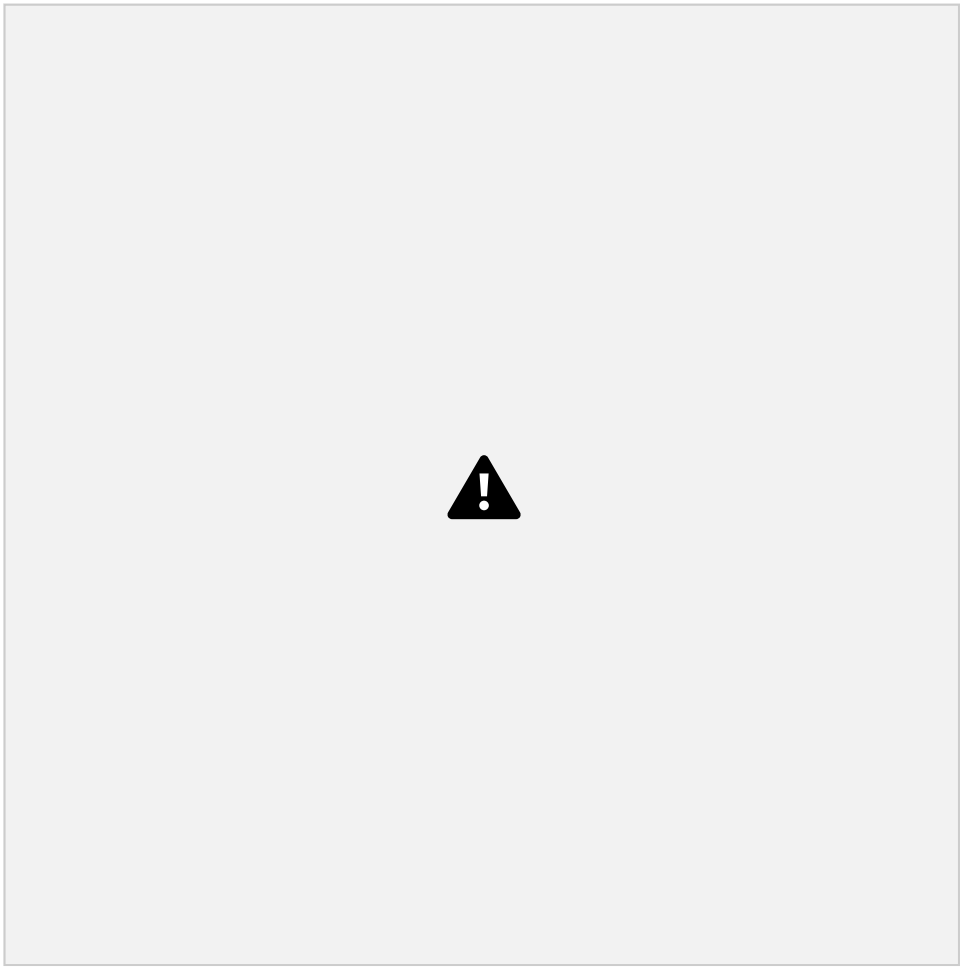

Рисунок 3.3 - Матриця невідповідностей для OtherNet з функцією втрат перехресної ентропії

Значна кількість шаблонів атак була класифікована як безпечні. Зокрема,

аналізи, які були помилково класифіковані як безпечні. Модель досягла високої повноти для класів, які вона класифікувала, з найнижчою оцінкою в класі експлойтів - 88,9%.

На рисунку 3.4 показано, що IoTNet, як і OtherNet, класифікував зразки лише як найбільший клас, але, на відміну від OtherNet, було менше зразків атак, неправильно класифікованих як безпечні; Повнота була такою ж або нижчою, ніж у OtherNet, причому відсоток фазерів зменшився найбільше , який зменшився з 91,60% до 79,84%.

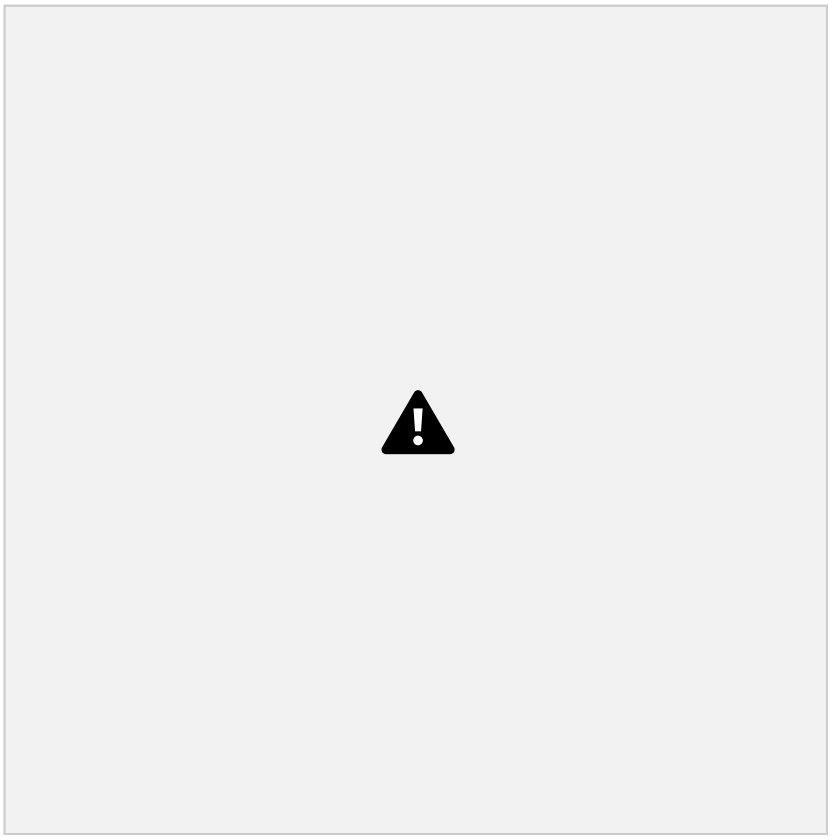

Рисунок 3.4 - Матриця невідповідностей для IoTNet з функцією втрат перехресної ентропії

На рисунку 3.5 показано ефект повторної вибірки за допомогою зваженого випадкового семплера в OtheNet; Worms був найменшим класом і мав найбільшу кількість зважених вибірок, що дало найкращі результати, з повнотою 100% DoS атаки були найскладнішими для класифікації і були класифіковані правильно. Лише 37,08% зразків було класифіковано.

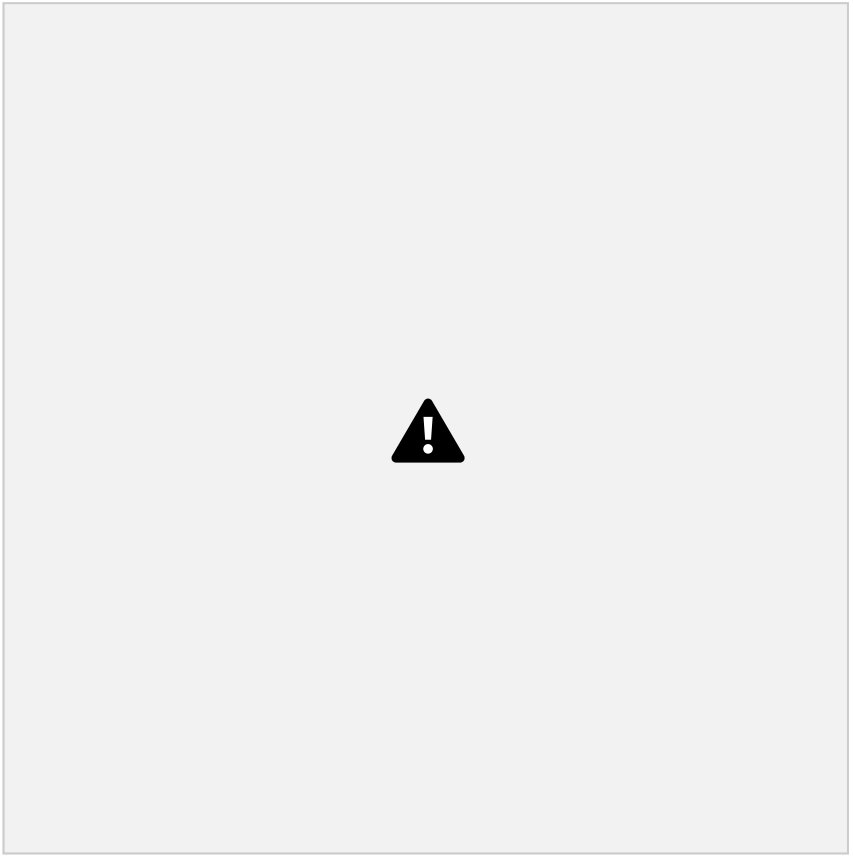

Рисунок 3.5 - Матриця невідповідностей для OtherNet зі Зваженим випадковим семплером

Аналіз рисунку 3.6 показує, що як IoTNet, так і OtherNet мають схожі проблеми з коректною класифікацією DoS-зразків. У порівнянні з OtherNet з передискретизацією, повнота класифікації шеллкоду в IoTNet значно вища і становить 91,23%. В обох мережах кількість зразків, помилково класифікованих як безпечні, була меншою, коли використовувалася передискретизація, ніж коли передискретизація не використовувалася. Загалом, класифікації стали більш рівномірно розподілені по всіх 10 категоріях, що вказує на те, що цей метод може частково вирішити проблему дисбалансу класів у наборі даних.

Зважена функція втрати перехресної ентропії (Рис. 3.7) показує вплив використання зваженої функції втрати перехресної ентропії на продуктивність OtherNet. З цього рисунку видно, що OtherNet не може виявити додаткові класи, що призводить до незначного зменшення кількості атак, класифікованих як безпечні, і збільшення кількості безпечних зразків, помилково класифікованих як атаки.

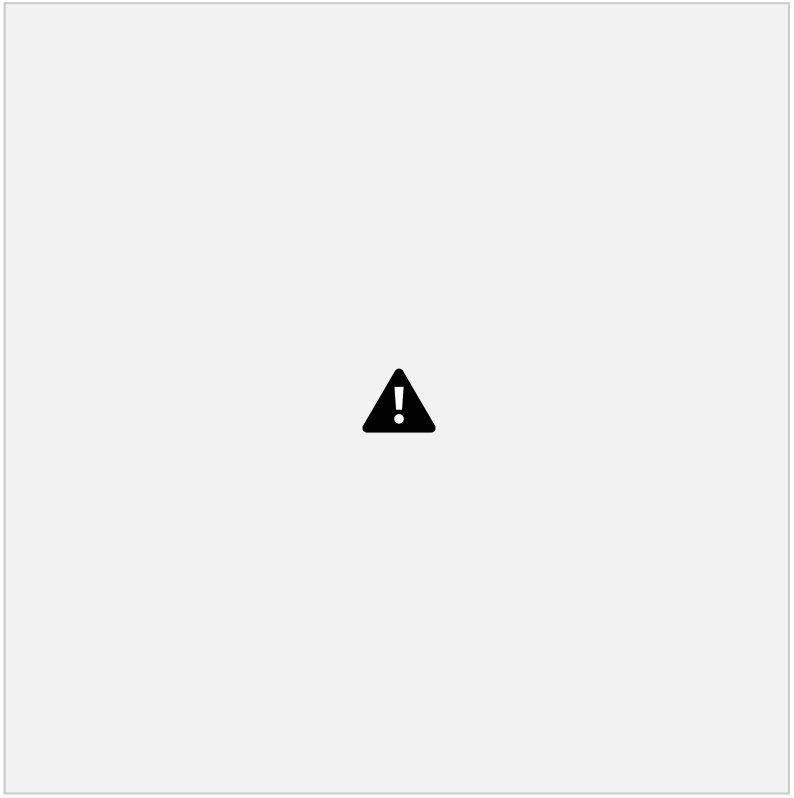

Рисунок 3.6 - Матриця невідповідностей для IoTNet із Зваженим випадковим семплером

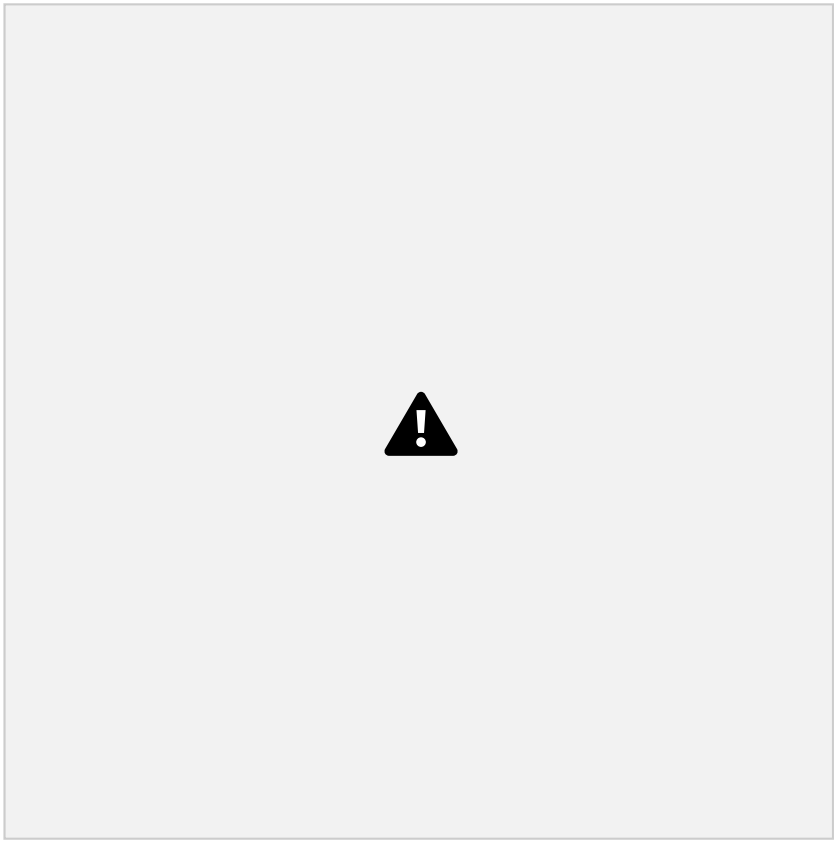

Рисунок 3.7 - Матриця невідповідностей для OtherNet із Зваженою крос ентропійною функцією втрат

На рисунку 3.8 показано результати IoTNet при використанні зваженої

функції втрати перехресної ентропії. З нього видно, що кількість класів, використаних для класифікації, збільшилася, і деякі зразки були класифіковані як дослідницькі.

Однак жоден із зразків, класифікованих як дослідницькі, не був класифікований правильно: як і в OtherNet, було менше хибнопозитивних спрацьовувань у безпечних класах і більше хибнонегативних спрацьовувань у небезпечних класах; на відміну від OtherNet, повнота фазерів збільшилася, а повнота експлойтів зменшилася.

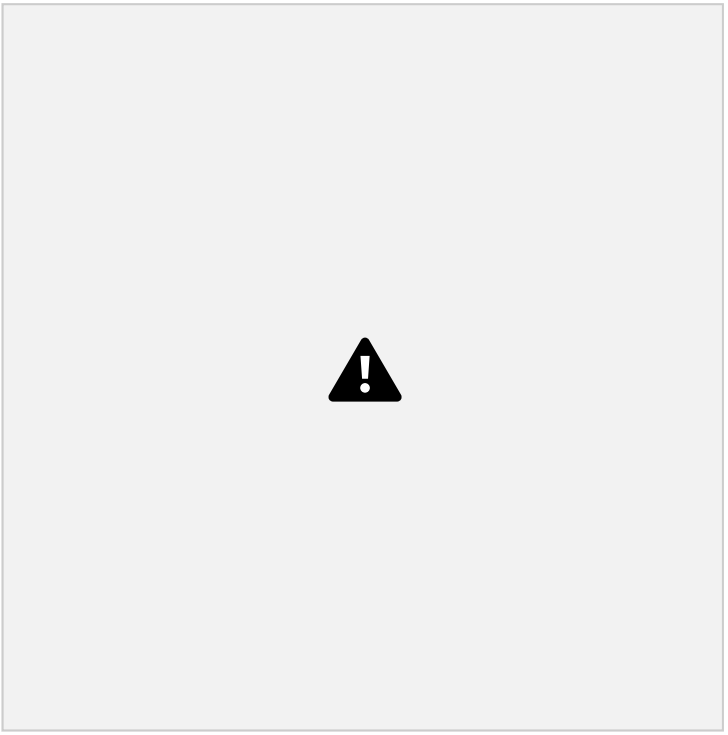

Рисунок 3.8 - Матриця невідповідностей для IoTNet зі Зваженою крос ентропійною функцією втрат

Аналіз на рисунку 3.9 показує, що при використанні функції втрати балансу класів OtherNet зразки класифікуються лише на дві великі категорії, з великою кількістю хибних спрацьовувань для безпечних класів. Це гірше, ніж без методу покращення навчання на незбалансованих наборах даних, як показано на рисунку 3.3.

На рисунку 3.10 показано результати з використанням функції втрати балансу класів IoTNet, де всі приклади класифіковано як атаки з використанням

бекдорів, що призвело до найгіршої продуктивності моделі.

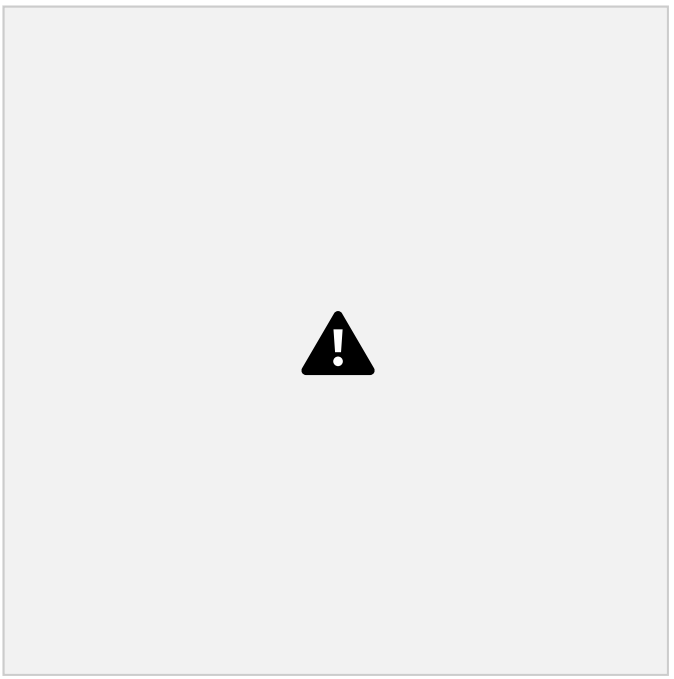

Рисунок 3.9 - Матриця невідповідностей для OtherNet зі збалансованою за класами функцією втрат

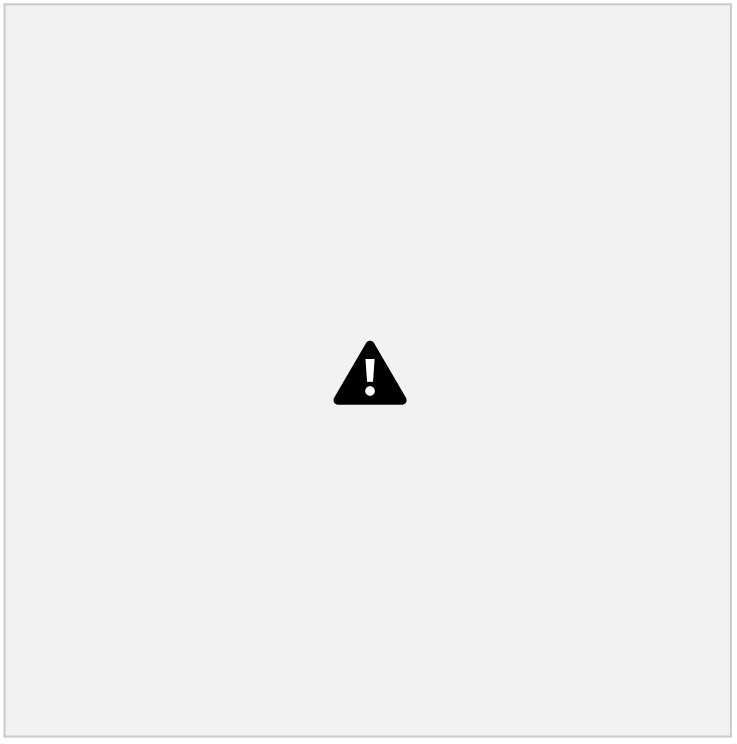

Рисунок 3.10 - Матриця невідповідностей для IoTNet зі збалансованою за класами функцією втрат

На основі даних схем маю певні негативні та позитивні підсумки. Таблиця 3.3 показує результати повторної вибірки за допомогою зваженого випадкового генератора для кожного класу OtherNet. Цифри в дужках показують різницю з

результатами Sarhan et. al [32]: OtherNet демонструє вищу повноту в 6 з 10 класів. Найбільша перевага спостерігається в аналізі з коефіцієнтом повноти 0,93, що на 62,44% вище.

Показники F1 для всіх класів, окрім бекдорів, є нижчими для OtherNet. Клас з найвищою загальною ефективністю - це SECURE. Надзвичайно низька точність пояснюється помилковою класифікацією, поширеною серед більшості класів, як показано на рисунку 3.5.

Таблиця 3.3 - Результати багатокласової класифікації OtherNet

Клас	Accuracy	Precision	Recall (DR)	*F*1 Score
Аналіз	99.84%	0.088	0.93 (+62.44%)	0.16 (−0.01)
Бекдор	99.92%	0.11	0.67 (+26, 37%)	0.19 (+0.01)
Безпечний	97.51%	1.0	0.97 (−02.40%)	0.99 (−0.01)
DoS	99.72%	0.22	0.37 (+07.51%)	0.28 (−0.08)
Експлойти	97.56%	0.29	0.67 (−13.27%)	0.41 (−0.43)
Фазери	99.66%	0.70	0.85 (+04.16%)	0.77 (−0.08)
Загальний	99.76%	0.30	0.70 (−15.44%)	0.42 (−0.48)
Розвідка	99.87%	0.69	0.95 (+14.98%)	0.80 (−0.03)
Шел-код	99.88%	0.15	0.60 (−28.02%)	0.23 (−0.46)
Хробаки	99.97%	0.19	1.00 (+14.02%)	0.31 (−0.38)

Результати для IoTNet, отримані шляхом повторної вибірки за допомогою зваженого випадкового пробовідбірника, представлені в таблиці 3.4 для кожного класу в тому ж форматі, що і в таблиці 3.3 IoTNet демонструє вищу варіабельність з точки зору точності та повноти порівняно з OtherNet, при цьому загальна та факторна класифікація показує гірші результати.

Таблиця 3.4 - Результати багатокласової класифікації IoTNet

Клас	Accuracy	Precision	Recall (DR)	*F*1 Score

Аналіз	99.81%	0.073	0.93 (+62.44%)	0.14 (−0.03)
Бекдор	99.84%	0.057	0.67 (+26, 37%)	0.11 (−0.07)
Безпечний	99.07%	1.0	0.99 (−00, 80%)	1.0
DoS	99.73%	0.12	0.14 (−15, 71%)	0.13 (−0.23)
Експлойти	99.27%	0.78	0.57 (−23, 50%)	0.66 (−0.18)
Фазери	99.52%	0.63	0.65 (−15, 51%)	0.64 (−0.21)
Загальний	99.52%	0.12	0.44 (−41, 17%)	0.19 (−0.71)
Розвідка	99.87%	0.7	0.96 (+15, 75%)	0.81 (−0.02)
Шел-код	99.61%	0.067	0.91 (+03, 56%)	0.12 (−0.57)
Хробаки	99.96%	0.12	0.91 (+04, 93%)	0.22 (−0.47)

Цей результат відповідає результатам, представленим Sarhan et al. Ми змогли досягти більшої повноти у всіх перерахованих вище класах, використовуючи OtherNet з передискретизацією. Наша модель показала гірші результати при класифікації експлойтів, дженериків та шеллкоду [32].

Таблиця 3.5 - Час висновування тестового набору

Модель	Час , с
OtherNet (CELoss/CBLoss)	50
OtherNet Sampler	52
IoTNet (CELoss/CBLoss)	52
IoTNet Sampler	54

У таблиці 3.5 показано час виведення для кожної моделі на тестовому наборі. Максимальна різниця у часі виведення, яку ми спостерігали, становила 4 с між OtherNet з навчанням на основі цінності та IoTNet зі зваженими випадковими вибірками. Не було ніякої різниці між двома методами навчання на

основі цінності, причому останній був найефективнішим, а останній - найменш

ефективним. Середній час виведення склав 51 секунду. Середній час виведення на вибірку становить 0,27 мс, оскільки вимірюється час виведення для всього тестового набору.

3.3 Висновок до розділу

Експериментальні результати показують, що навіть невеликі згорткові нейронні мережі можна ефективно використовувати для класифікації мережевих атак на основі набору даних NF-UNSW-NB15-v2, якщо застосувати методи передискретизації. Порівняно з еталонним дослідженням, вищі показники повноти були досягнуті щонайменше для половини всіх класів як в моделі OtherNet, так і в моделі IoTNet. Водночас спостерігалося загальне зниження метрики F1, що вказує на те, що помилки між класами були рівномірно розподілені і не зміщені в бік домінуючого класу. Цю тенденцію підтверджують результати, наведені на рисунках 3.5 та 3.6.

Аналіз моделі показав, що вона має обмежені можливості узагальнення і тому не може бути ефективно використана для виявлення атак нульового дня. Тим не менш, гібридний підхід, що поєднує передискретизацію та навчання з урахуванням вартості, може покращити якість узагальнення та підвищити ефективність виявлення атак з рідкісних класів. Ми не проводили повного дослідження гіперпараметрів, зосередившись лише на найбільш впливових з них - швидкості навчання та розмірі пакетів; кількість блоків і груп у моделі IoTNet та рівень регуляризації (відсіву) не коригувалися через значний час, витрачений на навчання.

На етапі відбору навчальних даних спочатку розглядався більший набір даних, але через ще вищий рівень дисбалансу та ризик того, що певні класи будуть недостатньо представлені, було вирішено працювати з меншим, але більш структурованим набором даних. Це виявилося найкращим компромісом між

представництвом усіх класів та економією часу. Однак це рішення обмежило

можливість узагальнення моделі.

В цілому, аналіз показав, що досягнутий в експериментах рівень класифікації був недостатнім для практичного використання моделі як повноцінної системи виявлення вторгнень. Навчені моделі показали низьку продуктивність порівняно з еталонами. Найкращі результати були отримані при використанні ітеративних семплерів, але навіть тоді точність для певних класів залишалася низькою. Тому перспективним напрямком подальших досліджень є розробка гібридних підходів, які поєднують повторну вибірку та навчання з урахуванням вартості. Це дозволило б моделям зосередитися на класах, які важко класифікувати, надавши їм додаткову вагу у функції втрат.

Середній час навчання як для досліджуваних моделей OtherNet, так і для IoTNet був занадто великим, щоб реалізувати процес навчання безпосередньо на пристроях IoT. Такий сценарій не був передбачений, і моделі довелося повністю навчати в зовнішньому середовищі, зокрема в хмарній інфраструктурі, перш ніж розгортати навчену версію безпосередньо на пристрої IoT для виконання висновків на основі поточних вхідних мережевих пакетів. Теоретично, можна виконати обмежене додаткове навчання локально на пристрої, орієнтуючись на нові шаблони атак, які щойно були виявлені в потоці трафіку.

Фактор часу навчання виявився вирішальним для побудови ефективної IDS, оскільки швидкість оновлення моделі визначає її адаптивність у разі виникнення нових загроз. Тому бажано мати можливість повністю навчити модель за обмежений час, наприклад, вночі. Під час експериментів IoTNet постійно показував довший час навчання, ніж OtherNet, але результати класифікації залишалися низькими. Середній час класифікації для всього тестового набору склав 51 с, що відповідає приблизно 0,27 мс на зразок.

Така швидкість не гарантує ефективної роботи в реальному часі при високому навантаженні мережі. Як правило, це може призвести до накопичення недіагностованих зразків і затримок у процесі реагування. Однак навіть така швидкість класифікації може бути прийнятною для обладнання з обмеженою

пропускною здатністю мережі: різниця в часі висновку між OtherNet і IoTNet становить менше трьох секунд, що свідчить про те, що продуктивність на етапі

прогнозування майже ідентична.

Основною метою було створення класифікатора, який можна було б ефективно узагальнити, особливо для виявлення атак нульового дня. Жодна з протестованих моделей не продемонструвала достатньої узагальнювальної здатності і характеризувалася зниженою точністю при класифікації нових або погано представлених зразків. Одним із способів покращення узагальнення може бути введення вагових коефіцієнтів у функцію втрат, вибірку або і те, і інше. Іншим підходом до покращення якості узагальнення є збільшення розміру моделі, або за рахунок збільшення коефіцієнта ширини IoTNet, або за рахунок використання більших зображень.

Однією з головних труднощів у процесі навчання моделі була незбалансованість використовуваних наборів даних. Навіть за допомогою методів повторної вибірки та навчання на основі вартості не вдалося досягти задовільного рівня точності класифікації для всіх класів. Поточний дисбаланс у кількості вибірок між більшістю та меншістю класів має значний вплив на узагальнюючі можливості моделі, що особливо помітно в задачі виявлення рідкісних типів атак.

На етапі попередньої обробки даних доцільно провести розширений аналіз ознак з метою оцінки їх інформативності та значущості. Це дозволяє диференційовано розподілити ознаки при формуванні зображення і надати більше місця параметрам з більшою класифікаційною вагою. Крім того, повторення ознак або включення перетворень (наприклад, обертання, масштабування, шумування) для створення більших зображень може покращити вхідне представлення шару згортки та зменшити ймовірність надмірної підгонки.

ВИСНОВКИ

В цій кваліфікаційній роботі було проведено повний цикл дослідження, проектування та програмної реалізації системи виявлення аномального мережевого трафіку з використанням методів глибокого навчання. Основною метою даної роботи була побудова ефективної системи виявлення атак в мережах Інтернету речей (IoT), що характеризуються високим ступенем гетерогенності, обмеженими обчислювальними ресурсами пристроїв та складністю ідентифікації новітніх загроз. Ці фактори обґрунтували вибір підходу на основі згорткових нейронних мереж (CNN), які можуть обробляти структуровані дані NetFlow, перетворені в зображення.

На етапі проектування системи був проведений детальний аналіз методів машинного та глибокого навчання, що дозволило сформувати два модельних архітектурних рішення: OtherNet є базовою структурою ШНМ, в той час як IoTNet орієнтована на роботу в умовах обмежених ресурсів IoTNet - структура CNN, орієнтована на роботу в умовах обмежених ресурсів. Запропонований підхід характеризується перетворенням числових характеристик NetFlow у візуальне представлення, кольорове зображення 7×7, що дозволило повною мірою використати переваги CNN. Крім того, було проведено високоякісну обробку даних, таку як очищення, нормалізація та стратифіковане розбиття, щоб забезпечити рівномірне представлення класів у навчанні.

Значну увагу було приділено проблемі дисбалансу класів, яка є типовою для задач виявлення вторгнень. Було реалізовано та експериментально досліджено декілька підходів до вирішення цієї проблеми: зважена ресемплінг (ресемплінг за допомогою PyTorch's WeightedRandomSampler), класово-зважені функції втрат (Weighted Cross-Entropy Loss), кожна з яких збалансовані за класами втрати (Class-Balanced Loss), що враховують кількість ефективних вибірок у кожному класі. Експериментальні результати показують, що поєднання зваженої вибірки та ефективних архітектур (особливо OtherNet) дозволяє досягти високої повноти (до 100%) для рідкісних класів атак, таких як Worms та Reconnaissance.

Програмна реалізація дозволила протестувати і порівняти вісім моделей з

різними архітектурами і різними способами усунення дисбалансу. Аналіз навчання та валідації моделей показав швидку збіжність та стабілізацію значень втрат у перші 20 епох. Найкращі результати були досягнуті з моделями, що використовують зважені функції втрат або повторну вибірку, але були відмінності в здатності моделей узагальнювати дані. Наприклад, IoTNet показала вищу точність у певних класах і меншу кількість хибних спрацьовувань, що свідчить про її корисність для вбудованих рішень IoT.

Фінальна оцінка класифікаційних можливостей моделей з використанням матриці розбіжностей між класами та метрик точності, відтворення та оцінки F1 підтверджує, що запропонована система може ефективно виявляти як поширені, так і рідкісні атаки. Модель OtherNet, навчена за допомогою зваженого семплера, виявилася найефективнішою у виявленні міноритарних класів, перевершивши результати, отримані в попередніх дослідженнях. Водночас IoTNet демонструє менший час обробки на тестовому наборі, що є значною перевагою для використання в обмежених умовах.

В рамках цієї статті було протестовано кілька підходів до підвищення точності моделей глибокого навчання при роботі з незбалансованими наборами даних. Зокрема, було досліджено ефективність передискретизації та навчання з урахуванням вартості, щоб оцінити вплив цих підходів на класифікаційні характеристики моделей. Результати експериментів визначили перспективні напрямки для подальшого вдосконалення підходу до виявлення аномалій мережевого трафіку.

Одним з важливих напрямків подальших досліджень є можливість реалізації гібридного підходу, який поєднує оптимізацію вагових коефіцієнтів з передискретизацією та навчанням з урахуванням вартості. Така комбінація може дозволити моделям більш точно адаптуватися до рідкісних класів і значно зменшити вплив дисбалансу в даних. Крім того, для збагачення варіабельності навчальної вибірки можна використовувати інші стратегії вибірки або методи передискретизації.

Особливу увагу слід приділити методам попередньої обробки, які дозволяють формувати зображення без використання заповнення, оскільки

надмірне додавання неінформативних значень може негативно вплинути на якість навчання. Оскільки згорткові нейронні мережі використовують фільтри для обробки локальних областей зображення, бажано дослідити підходи, які збільшують розмір зображення за рахунок реплікації ознак, що може підвищити локальну кореляцію між ознаками.

Крім того, існує потенціал для подальшого вдосконалення архітектури IoTNet та оптимізації гіперпараметрів. Зокрема, швидкість навчання і розмір пакетів можна адаптивно регулювати за допомогою сучасних методів, таких як планування швидкості навчання. Оскільки в цьому дослідженні використовувався виключно Adam, використання альтернативних оптимізаторів для навчання моделі також є перспективним. Розширення цього підходу може покращити якість фактичного узагальнення та продуктивність моделі.

Таким чином, результати цього дослідження мають як теоретичне, так і практичне значення. З одного боку, вони доводять ефективність застосування CNN до даних NetFlow у форматі зображень, а з іншого боку, реалізована система може бути інтегрована в реальні мережеві інфраструктури, особливо в контексті IoT-пристроїв та периферійних вузлів. Подальші дослідження можуть бути спрямовані на вдосконалення архітектури моделі, поєднання CNN з рекурентними мережами та трансформаторами, а також на автоматичний вибір ознак за допомогою методів AutoML.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Що таке DDoS-атака та як їй запобігти? [Електронний ресурс]. –

Hostinger, 2023. – Режим доступу: https://support.hostinger.com/uk/articles/5634639-що-таке-ddos-атака-та-як-їй-запобігти (дата звернення 18.02.2025).

2. Поширені атаки на IoT та захист від них [Електронний ресурс]. – 2023. – Режим доступу: https://corewin.ua/blog/attacks-on-iot-how protect/?utm_source=chatgpt.com (дата звернення 18.02.2025).

3. DDoS-атака [Електронний ресурс]. – ESET, 2022. – Режим доступу: https://www.eset.com/ua/support/information/entsiklopediya-ugroz/distributed-denial of-service/ (дата звернення 18.02.2025).

4. 10 ознак того, що ваша мережа під кібератакою [Електронний ресурс]. – 10Guards, 30.09.2022. – Режим доступу: https://10guards.com/ua/blog/2022/09/30/10-signs-your-network-is-under-a-cyber attack/?utm_source=chatgpt.com (дата звернення 18.02.2025).

5. Все, що треба знати про DDoS-атаки [Електронний ресурс]. – NordVPN, 2023. – Режим доступу: https://nordvpn.com/uk/blog/shcho-take-ddos-ataka/ (дата звернення 18.02.2025).

6. Що таке експлойт (exploit) простими словами. Види, приклади та методи захисту [Електронний ресурс]. – Mignews, 2023. – Режим доступу: https://mignews.com.ua/tehnologii/shho-take-eksplojt-exploit-prostymy-slovamy vydy-pryklady-ta-metody-zahystu.html (дата звернення 18.02.2025).

7. Експлойт: Що це таке, приклади та захист [Електронний ресурс]. – CyberSet, 2024. – Режим доступу: https://cyberset.com.ua/cybersecurity/cybersecurity-basics/exploit/ (дата звернення 18.02.2025).

8. Що таке вразливість нульового дня: zero day – IT Education Blog [Електронний ресурс]. – IT Education, 2024. – Режим доступу: https://itedu.center/ua/blog/articles/zero day/?srsltid=AfmBOoqrUs0PX0Bqn_asKPHKfyue9py4dUKT8pyhtbSM4xjWGKfIP MZs (дата звернення 18.02.2025).

9. WannaCry [Електронний ресурс]. – Wikipedia, 2024. – Режим доступу: https://uk.wikipedia.org/wiki/WannaCry (дата звернення 18.02.2025). 10.

Вразливість Heartbleed [Електронний ресурс]. – Bakotech, 2023. – Режим доступу: https://bakotech.ua/ua/news/heartbleed-uyazvimost-v-openssl-kak zashchititsya/ (дата звернення 18.02.2025).

11. Що таке Meltdown – Терміни та визначення у сфері кібербезпеки [Електронний ресурс]. – VPN Unlimited, 2024. – Режим доступу: https://www.vpnunlimited.com/ua/help/cybersecurity/meltdown?srsltid=AfmBOoocV Jnrb_6bIXPFnsswEODWvmkHxedA-MvHm5JGMvKZ-W-obxn1 (дата звернення 20.02.2025).

12. Spectre (уразливість) [Електронний ресурс]. – Wikipedia, 2024. – Режим доступу: https://uk.wikipedia.org/wiki/Spectre_(%D1%83%D1%80%D0%B0%D0%B7%D0% BB%D0%B8%D0%B2%D1%96%D1%81%D1%82%D1%8C) (дата звернення 20.02.2025).

13. Що таке атака "Людина посередині" (MITM) та як себе захистити? | CyberCalm [Електронний ресурс]. – 2023. – Режим доступу: https://cybercalm.org/novyny/shho-take-ataka-man-in-the-middle-ta-yak-sebe zahystyty/ (дата звернення 20.02.2025).

14. Brute force атаки: Як вони працюють і як захистити дані [Електронний ресурс]. – Foxminded, 2023. – Режим доступу: https://foxminded.ua/brute-force/ (дата звернення 20.02.2025).

15. 3 Types of Anomalies in Anomaly Detection | HackerNoon [Електронний ресурс]. – HackerNoon, 2023. – Режим доступу: https://hackernoon.com/3-types-of anomalies-in-anomaly-detection?utm_source=chatgpt.com (дата звернення 20.02.2025).

16. Zhou Y., Zhang W., Zeng H., Zheng Z. Модель виявлення мережевих вторгнень на основі алгоритму кластеризації K-means та багатьох згорткових нейронних мереж. У: 4-та міжнародна конференція з розширених алгоритмів і нейронних мереж (AANN), 2024. Neurophotonics; том 13416: 134161M. 17. Xie S. Покращений алгоритм кластеризації K-means для виявлення вторгнень у бездротових мережах на основі федеративного навчання. Wireless Communications and Mobile Computing 2021; 2021: 9322368.

18. Hao S., Fu W., Chen X. та ін. Виявлення аномального трафіку в мережі за допомогою мультивиглядового злиття ознак. arXiv preprint 2024; arXiv:2409.08020.

19. Moretti D., Onofri E., Cristiani E. Виявлення аномального транспортного трафіку та збоїв датчиків за допомогою методів кластеризації. arXiv preprint 2025; arXiv:2504.00881.

20. Caville E., Lo W.W., Layeghy S., Portmann M. Anomal-E: система самонавчального виявлення вторгнень у мережу на основі графових нейронних мереж. arXiv preprint 2022; arXiv:2207.06819.

21. Duan X., Fu Y., Wang K. Метод виявлення аномалій у мережевому трафіку на основі багатомасштабного залишкового класифікатора. Computer Communications 2023; 206: 206–216.

22. Budiati H., Wijaya A.B.M., Zebua B.S.V., Jatmika, Sumihar Y.P. Застосування методу кластеризації K-means для виявлення аномалій мережевого трафіку. Jurnal Mantik 2022; 6(3): 3499–3504.

23. Zhang M., Subramaniya K. Аналіз систем виявлення мережевих вторгнень на основі алгоритму K-means. Advances in Science, Technology and Engineering Systems Journal 2025; 3(1): 60.

24. .Gadal S., Mokhtar R., Abdelhaq M., Alsaqour R., Ali E. S., Saeed R.. Machine learning-based anomaly detection using K-Mean array and sequential minimal optimization / S. Gadal [et al.] // Electronics. — 2022. — Vol. 11, no. 2158. DOI: https://doi.org/10.3390/electronics11142158.

25. Thankappan M., Rifà-Pous H., Garrigues C.. A signature-based wireless intrusion detection system framework for multi-channel man-in-the-middle attacks against protected Wi-Fi networks / M. Thankappan, H. Rifà-Pous, C. Garrigues // IEEE Access. — 2024. — Vol. 12. — P. 23096—23121. DOI: 10.1109/ACCESS.2024.3362803.

26. Корнієнко В., Герасіна О., Тимофєєв Д., Сафаров О., Ковальова Ю.. Ідентифікація та прогнозування самоподібного трафіку інформаційно комунікаційних мереж для систем виявлення атак / В. Корнієнко [та ін.] //

Information Technology: Computer Science, Software Engineering and Cyber Security. — 2022. — № 1. — С. 20—29. DOI: https://doi.org/10.32782/IT/2022-1-4.

27. Zhang K., Polycarpou M. M., Parisini T.. Enhanced anomaly detector for nonlinear cyber-physical systems against stealthy integrity attacks / K. Zhang, M. M. Polycarpou, T. Parisini // IFAC-PapersOnLine. — 2020. — Vol. 53, no. 2. — P. 13682—13687. DOI: https://doi.org/10.1016/j.ifacol.2020.12.870.

28. Волокита А., Меленчуков М.. Дослідження моделей виявлення атак на розподілені системи за допомогою згорткових нейронних мереж / А. Волокита, М. Меленчуков // Measuring and Computing Devices in Technological Processes. — 2024. — № 3. — С. 224—229. DOI: https://doi.org/10.31891/2219-9365-2024-79-29.

29. Sheng S., Wang X.. Network traffic anomaly detection method based on chaotic neural network / S. Sheng, X. Wang // Alexandria Engineering Journal. — 2023. — Vol. 77. — P. 567—579. DOI: https://doi.org/10.1016/j.aej.2023.07.019.(29)

30. Янко А., Прокудін А., Філь І., Крук О.. Виявлення атак типу LDDOS за допомогою SDN мереж з елементами машинного навчання / А. Янко [та ін.] // Measuring and Computing Devices in Technological Processes. — 2024. — № 4. — С. 287—296. DOI: https://doi.org/10.31891/2219-9365-2024-80-36.

31. Protic D., Stanković M., Antić V.. WK-FNN design for detection of anomalies in the computer network traffic / D. Protic, M. Stanković, V. Antić // Facta Universitatis - Series: Electronics and Energetics. — 2022. — Vol. 35, no. 2. — P. 269—282. DOI: 10.2298/FUEE2202269P.

32. M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, “Towards a standard feature set of NIDS datasets,” CoRR, vol. abs/2101.11315, 2021.

33. J. Kim, J. Kim, H. Kim, M. Shim, and E. Choi, “Cnn-based network intrusion detection against denial-of-service attacks,” Електроніка, том 9, № 6, 2020.

34. D. Vasan, M. Alazab, S. Wassan, H. Naeem, B. Safaei, and Q. Zheng, “Im cfn: Image-based malware classification using fine-tuned convolutional neural network architecture,” Computer Networks, vol. 171, p. 107138, 2020.

35. A. M. Zaza, S. K. Kharroub, and K. Abualsaud, “Lightweight iot malware de tection solution using cnn classification,” in 2020 IEEE 3rd 5G World Forum (5GWF), 2020, pp. 212–217.

36. Kaspersky. You shall not pass. Kaspersky checked 193 million passwords for resistance to various compromising techniques [Електронний ресурс] – 2024. – Режим доступу: https://www.kaspersky.com/about/press-releases/you-shall-not-pass kaspersky-checked-193-million-passwords-for-resistance-to-various-compromising techniques

37. Novatech. Password Security in 2024: A Deep Dive into Best Practices [Електронний ресурс] – 2024. – Режим доступу: https://novatech.net/blog/password security-in-2024-a-deep-dive-into-best-practices

38. Meyer L.A., Romero S., Bertoli G., Burt T., Weinert A., Lavista Ferres J. How effective is multifactor authentication at deterring cyberattacks? [Електронний ресурс] – 2023. – Режим доступу: https://arxiv.org/abs/2305.00945 (дата звернення 18.02.2025).

39. MetaCompliance. Password Policy Best Practices 2023 [Електронний ресурс] – 2023. – Режим доступу: https://www.metacompliance.com/blog/cyber security-awareness/password-policy-best-practices-2023

40. Di Tizio G., Armellini M., Massacci F. Software Updates Strategies: a Quantitative Evaluation against Advanced Persistent Threats [Електронний ресурс] – 2022. – Режим доступу: https://arxiv.org/abs/2205.07759