МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

КУЛИК Степан Андрійович

Виявлення мережевих атак у трафіку великих даних на основі еволюційних обчислень / Detection of Network Attacks in Big Data Traffic Based on Evolutionary Computations

Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

Кваліфікаційна робота

Виконав студент групи КН-43
С.А. Кулик

Науковий керівник:
к.е.н., доцент, О. Ю. Ніпіаліді

Кваліфікаційну роботу допущено до захисту
«___»_____ 2025 р.

В.о. завідувача кафедри
_____ Н.М. Васильків

Тернопіль – 2025

ЗМІСТ

ВСТУП

Завдяки стрімкому розвитку хмарних обчислень і віртуалізації стало можливим ефективно обробляти великі обсяги даних, які надходять від інтелектуальних сенсорних пристроїв. Це дозволяє як збирати дані з багатьох джерел (вхідний трафік), так і приймати обґрунтовані рішення на їх основі (вихідний контроль) [1].

На відміну від традиційних комп'ютерних мереж, середовище великих даних відзначається різноманітністю джерел інформації, різною природою цих даних, а також великою кількістю типів пристроїв, які їх генерують. Обмін даними у таких мережах відбувається майже безперервно, в режимі реального часу, з використанням складних технологій передачі. У зв'язку з цим зростають вимоги до потужності пристроїв, швидкості реагування системи та точності передачі інформації [2].

Через такі особливості трафік у великих даних стає вразливим до різного роду кіберзагроз. Це створює серйозні ризики для безпеки усієї інформаційної системи. Зловмисники можуть скористатися вразливими місцями для викрадення комерційних даних, блокування доступу до послуг, знищення або викривлення інформації, а також для шантажу й виведення з ладу критичних компонентів систем [2]. Такі атаки можуть не лише завдати значної шкоди промисловому виробництву, але й нести загрозу національній безпеці. Наприклад, масштабні DDoS-атаки (розподілені атаки на відмову в обслуговуванні), яка відбуваються в Україні під час війни, спричинили серйозні збої в роботі систем.

Точне й своєчасне виявлення мережевих атак дозволяє виявляти потенційні загрози безпеці ще на ранніх етапах та вживати відповідних заходів захисту. Саме тому тема виявлення атак у трафіку великих даних стає дедалі актуальнішою [3].

Метою кваліфікаційної роботи є розробка методу виявлення мережевих атак у трафіку великих даних на основі поєднання еволюційних обчислень та

методів машинного навчання з метою підвищення точності, ефективності та швидкодії в умовах високої розмірності та багатовимірності даних.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

– провести аналіз сучасних методів виявлення мережевих атак;

– виявити основні проблеми, пов’язані з обробкою високовимірного трафіку;

– розробити ефективну процедуру попередньої обробки даних;

– запропонувати ієрархічний підхід до вибору ознак для двокласової та багатокласової класифікації;

– реалізувати двошарову модель класифікації мережевого трафіку з використанням LightGBM;

– провести експериментальні дослідження з використанням відкритих наборів даних;

– порівняти отримані результати з існуючими підходами.

Об'єктом дослідження є процеси виявлення мережевих атак у високонавантажених інформаційних системах, що функціонують у середовищі великих даних.

Предметом дослідження є методи та алгоритми виявлення мережевих атак на основі еволюційних обчислень і машинного навчання, зокрема із застосуванням ієрархічної класифікації трафіку та вибору ознак у багатовимірних даних.

Результати кваліфікаційної роботи апробовані та опубліковані у матеріалах студентської науково-практичної конференції “Інтелектуальні інформаційні технології в прикладних дослідженнях” (IITAR – 2025), м. Тернопіль, Україна, 27-29 травня 2025 р.

Кваліфікаційна робота складається із вступу, трьох розділів, висновків, списку використаних джерел та додатків.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ВИЯВЛЕННЯ МЕРЕЖЕВИХ АТАК

1.1 Опис предметної області

Серед найбільш поширених типів мережевих атак, що загрожують інформаційним системам у середовищі великих даних, можна виокремити такі [2, 3]:

1. DoS-атаки (Denial of Service) – атаки, спрямовані на виведення з ладу системи шляхом перевантаження її запитами.
2. DDoS-атаки (Distributed Denial of Service) – розподілені DoS-атаки, які здійснюються з великої кількості скомпрометованих пристроїв (ботнетів) і є особливо складними для виявлення та нейтралізації.
3. Фішинг (Phishing) – метод соціальної інженерії, за якого зловмисники намагаються виманити конфіденційні дані шляхом імітації легітимних ресурсів.
4. SQL-ін’єкції (SQL Injection) – атаки на веб-додатки з метою несанкціонованого доступу до баз даних через маніпуляції з SQL-запитами.
5. Сканування портів (Port Scanning) – розвідка мережі для виявлення відкритих портів і потенційних векторів атак.
6. Brute Force (груба сила) – підбір паролів або ключів доступу шляхом перебору можливих варіантів.
7. ARP-spoofing та DNS-spoofing – атаки, що пов’язані з підміною адрес або маршрутів трафіку з метою перехоплення даних або їх підробки.
8. Атаки типу "людина посередині" (Man-in-the-Middle, MitM) – атаки, за яких зловмисник перехоплює та потенційно змінює комунікації між двома сторонами.

Розпізнавання цих типів атак вимагає гнучких, масштабованих та адаптивних методів аналізу, здатних ефективно працювати з великими, стрімкими й різнорідними потоками трафіку.

У сучасних умовах цифровізації та зростання кількості підключених до мережі пристроїв проблема забезпечення кібербезпеки стає надзвичайно актуальною. Одним з найважливіших завдань у цій сфері є виявлення мережевих

атак, яке дозволяє запобігти несанкціонованому доступу до ресурсів, втраті конфіденційних даних, порушенню стабільної роботи інформаційних систем. Традиційні методи виявлення атак, такі як статистичний аналіз, аналіз поведінки користувачів та глибоке інспектування пакетів (Deep Packet Inspection, DPI), вимагають постійного моніторингу та аналізу стану всіх компонентів мережі. Хоча ці методи продемонстрували свою ефективність у локальних та середніх за масштабом мережах, вони виявляються малопридатними у середовищі великих даних, де обсяги трафіку надзвичайно великі, кількість пристроїв стрімко зростає, а взаємодії між ними стають все складнішими. У таких умовах традиційні системи спостереження та контролю потребують значних обчислювальних ресурсів і втрачають свою продуктивність, що знижує їхню ефективність у режимі реального часу [4].

У зв’язку з цим зростає інтерес до інтелектуальних методів, які базуються на машинному та глибокому навчанні [5, 6]. Ці підходи здатні автоматично аналізувати поведінкові характеристики мережевого трафіку, виявляти нетипові шаблони або аномалії, які можуть свідчити про спробу вторгнення чи іншу загрозу безпеці. Завдяки цьому стає можливим виявлення атак у режимі реального часу на основі аналізу поведінки, що є надзвичайно перспективним напрямом у галузі еволюційних обчислень. Суть такого підходу полягає у використанні моделей, які навчаються на історичних даних і згодом здатні розпізнавати нові, раніше невідомі види атак.

Особливу роль у цьому процесі відіграють глибокі нейронні мережі, зокрема згорткові (Convolutional Neural Networks, CNN) та рекурентні нейронні мережі (Recurrent Neural Networks, RNN), які широко використовуються в суміжних галузях – обробці природної мови, розпізнаванні образів, комп’ютерному зорі [7, 8]. Такі моделі здатні відображати складні нелінійні залежності між ознаками, які часто залишаються непоміченими для традиційних алгоритмів. У контексті мережевого трафіку це означає можливість точного виявлення прихованих ознак атак, їх динаміки та варіативності.

Проте, незважаючи на високу ефективність, глибоке навчання має низку обмежень при використанні у великих обсягах даних. Зокрема, йдеться про

високе обчислювальне навантаження, тривалий час навчання та значні вимоги до ресурсів пам’яті, що унеможливлює його ефективне застосування в реальному часі у багатьох прикладних системах [9]. Для вирішення цих проблем дослідники почали вивчати потенціал менш ресурсоємних моделей машинного навчання, таких як дерева рішень, метод опорних векторів або наївні байєсівські класифікатори. Ці моделі швидко навчаються та легко реалізуються, однак мають низьку здатність до узагальнення, погано пристосовані до змін у середовищі та часто втрачають точність при обробці нових або складних типів атак [10].

З огляду на зазначене, актуальним напрямом досліджень є поєднання еволюційних обчислювальних підходів із методами машинного навчання. Це дозволяє досягти компромісу між швидкістю, точністю та масштабованістю моделей. Розробка таких гібридних систем, які адаптивно навчаються та ефективно виявляють атаки в умовах інтенсивного та різнорідного трафіку, відкриває нові можливості для створення сучасних систем кіберзахисту, здатних діяти в умовах великих даних.

На рисунку 1.1 подано схематичне зображення мережевих атак у середовищі великих даних.

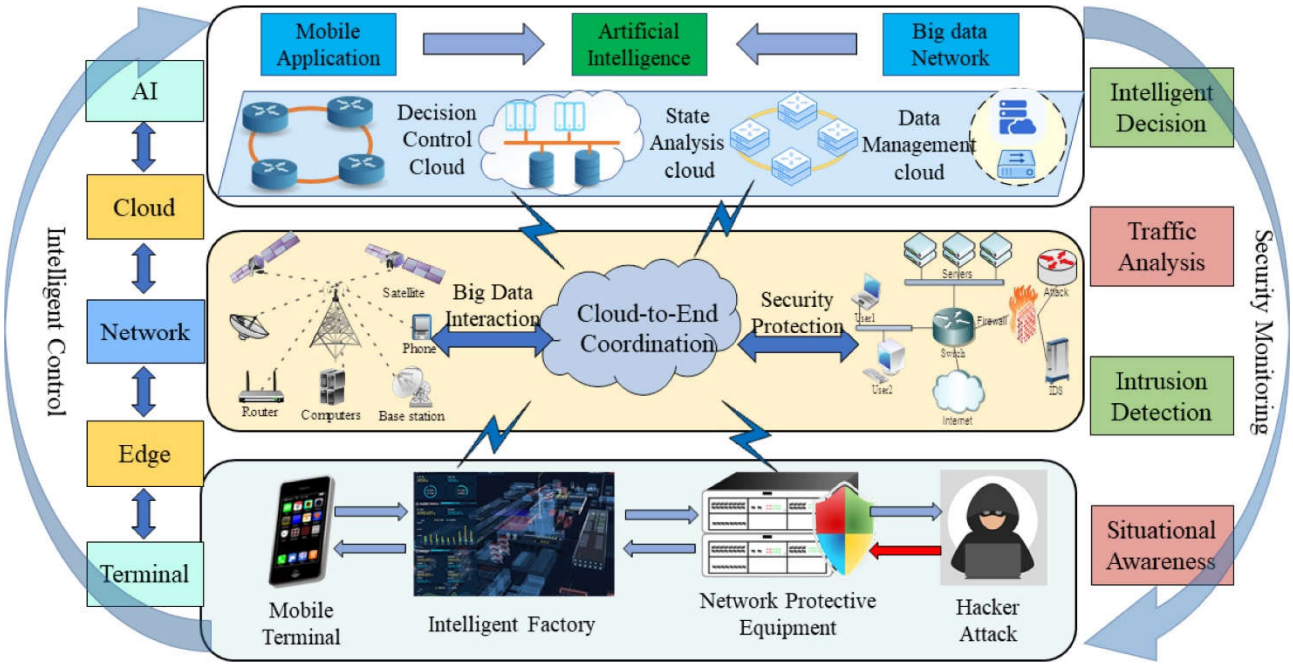

Рисунок 1.1 – Атаки у середовищі великих даних [11]

Як видно з рисунку, обробка й виявлення атак у такому складному середовищі пов’язана з низкою серйозних проблем, які досі залишаються невирішеними.

1.2 Аналіз відомих рішень

Виявлення мережевих атак у середовищі великих даних має враховувати обмеженість обчислювальних ресурсів і пропускної здатності, характерну для таких сценаріїв. Важливо, щоб методи виявлення не заважали нормальній роботі великої кількості пристроїв у мережі. Оскільки методи виявлення, орієнтовані на поведінку, стають дедалі популярнішими, темпи досліджень у цій сфері стрімко зростають. Залежно від використаних підходів, наявні роботи можна умовно поділити на три основні групи: методи на основі простого машинного навчання, методи на основі ансамблевого навчання та методи глибокого навчання.

Прості методи машинного навчання привертають усе більше уваги, особливо в контексті виявлення атак у великих даних. Проте результати попередніх досліджень показують, що методи, як-от дерева рішень, наївні байєсівські класифікатори та метод опорних векторів (SVM), мають серйозні обмеження. Зокрема, вони не демонструють належного рівня узагальнення на нових даних і схильні до перенавчання, що призводить до низької точності виявлення атак.

Щоб подолати ці обмеження, Saba та співавт. [12] поєднали генетичний алгоритм із методом опорних векторів. Генетичний алгоритм використовувався для вибору релевантних ознак, які потім подавалися на вхід до класифікатора SVM. Це дозволило суттєво підвищити точність виявлення. Cong та ін. [13] поєднали модель градієнтного бустингу з байєсівською оптимізацією для об’єднання результатів двох класифікаторів, що покращило здатність моделі до узагальнення.

Li та співавт. [14] запропонували метод стекування класифікаційних моделей, який поєднує чотири алгоритми: дерева рішень (DT), випадковий ліс (RF), надлишкові дерева (ET) і XGBoost. Об’єднання декількох моделей дало

змогу підвищити точність, однак такий підхід потребує значно більше часу на навчання, особливо при обробці даних з великою кількістю ознак. Це знижує загальну ефективність виявлення в реальному часі.

Інші дослідники також досліджували можливості зменшення розмірності даних для підвищення ефективності моделей. Pantelis та ін. [15] застосували поєднання вибору ознак із генетичним алгоритмом для оцінки ризиків, що дозволило зменшити складність моделі. Li та співавт. [16] використали алгоритм головних компонент (PCA) для покращення ваг ознак, зменшення зашумленості даних і зниження їхньої розмірності. У результаті було досягнуто покращення точності класифікації деревом рішень.

Zhang та ін. [17] поєднали метод головних компонент із лінійним дискримінантним аналізом для зменшення розмірності простору ознак, що дало змогу компенсувати нестачу ознак для менш представлених класів і суттєво покращити точність їх виявлення. Reddy та ін. [18] розробили новий алгоритм пошуку ознак у поєднанні з рекурентною нейронною мережею для ефективного виявлення атак у хмарному середовищі. Mata та співавт. [19] використали методику зменшення ознак на основі теорії нечітких множин (rough set), що дозволило покращити вхідні дані для дерева рішень, зменшити кількість помилкових спрацювань і підвищити точність виявлення.

На основі аналізу можна зробити висновок, що ефективні методи вибору ознак здатні не лише зменшити розмірність даних і скоротити час навчання моделей, але й покращити загальну ефективність виявлення атак. Крім того, релевантні ознаки мають тісний зв’язок із результатами класифікації, що позитивно впливає на точність розпізнавання різних класів даних і дозволяє зменшити кількість хибних спрацювань.

Ансамблеве навчання є одним із сучасних підходів у машинному навчанні. Його основна ідея полягає у поєднанні кількох слабких класифікаторів для досягнення кращого кінцевого результату шляхом голосування між ними. Такий підхід дозволяє підвищити точність розпізнавання і здатність до узагальнення у порівнянні з використанням лише однієї моделі [20].

Наприклад, Zhang та ін. [21] запропонували алгоритм виявлення вторгнень, що базується на ансамблі з трьох гілок випадкового лісу (Random Forest), який був побудований з використанням стратегії Bagging для формування кількох шарів класифікаторів. Це дало змогу підвищити точність виявлення атак.

Chang та співавт. [22] представили ненаглядний метод виявлення вторгнень, який поєднує моделі випадкового лісу та методу опорних векторів. Хоча вони внесли певні структурні вдосконалення до моделі Random Forest, точність виявлення все ще залишалася недостатньою, особливо через високу розмірність вхідних даних.

Ahmad та ін. [23] використали метод опорних векторів як базову модель, яку поєднали з методом інтегрованого навчання Adaboost (адаптивного бустингу). Вони ітеративно вдосконалювали базовий класифікатор, що дозволило підвищити точність у порівнянні з використанням одного класифікатора. Проте об’єднання кількох моделей ускладнило процес навчання, зробило його менш ефективним і менш придатним для обробки великих обсягів даних з високою розмірністю.

Hua та співавт. [24] запропонували метод ефективної класифікації мережевого трафіку, що поєднує вбудований підхід до вибору ознак з алгоритмом LightGBM. LightGBM — це легковаговий метод ансамблевого навчання, який здатен уникати перенавчання або недонавчання моделі. До того ж, він споживає небагато пам’яті й часу, що робить його зручним для використання в умовах великих даних. У поєднанні з ефективною попередньою обробкою даних і методами балансування класів LightGBM демонструє високу здатність виявляти атаки, які зустрічаються рідко.

Глибоке навчання, у свою чергу, застосовує різні типи глибоких нейронних мереж, які здатні моделювати складні нелінійні залежності. Це робить глибоке навчання потужним інструментом для виявлення прихованих просторово-часових ознак кібератак.

Наприклад, Kuang та ін. [25] застосували довгострокову короткочасну пам’ять (LSTM) для аналізу часових ознак у послідовних потіках даних. Це

дозволило краще виявляти часові закономірності, характерні для шкідливих дій у мережі, та підвищити точність виявлення зловмисного програмного забезпечення.

Проте традиційні LSTM-мережі мають обмеження: вони обробляють послідовності лише в одному напрямку й не враховують залежності між майбутніми подіями. Щоб вирішити цю проблему, Osama [26] і Fadwa [27] запропонували використання двонапрямленої LSTM-архітектури (BiLSTM), яка дозволяє аналізувати часові зв'язки з обох боків — як у минулому, так і в майбутньому відносно кожної точки в даних.

Zhang та співавт. [28] пішли далі й застосували багатомасштабну згорткову нейронну мережу (CNN) для видобування просторових ознак, а потім використали LSTM для аналізу часових зв'язків. Такий підхід дозволив спочатку виявити просторові характеристики атаки, а вже потім — проаналізувати часову динаміку.

Незважаючи на високу ефективність глибокого навчання у виявленні складних і прихованих ознак атак, слід зазначити, що такі моделі потребують значних обчислювальних ресурсів і часу на навчання. Особливо це стосується глибоких багатошарових архітектур, які можуть бути занадто витратними для практичного застосування у середовищах великих даних.

Крім того, в останні роки також з'явилися дослідження, які стосуються використання підкріплювального навчання для виявлення атак [29]. Однак, незважаючи на новизну підходу, його результати у реальному застосуванні виявилися менш ефективними, ніж очікувалося.

1.3 Постановка задачі дослідження

В умовах стрімкого зростання обсягів мережевого трафіку, яке є наслідком масового розгортання інтелектуальних пристроїв, Інтернету речей та хмарних обчислень, постає критично важлива задача — забезпечення ефективного і точного виявлення мережевих атак у середовищі великих даних. Однак вирішення цього завдання супроводжується низкою суттєвих труднощів.

Насамперед варто відзначити проблему високої розмірності даних, або так зване «прокляття розмірності». У такому середовищі генерується величезна кількість трафіку, який має складну багатовимірну структуру. Така кількість ознак ускладнює процес очищення даних і значно ускладнює процес навчання моделей машинного навчання, які не завжди здатні ефективно працювати з такими обсягами різнорідної інформації. Ще однією важливою проблемою є складність у виявленні інформативних ознак у даних великої розмірності. Велика кількість параметрів, багато з яких можуть бути слабо релевантними або надмірними, перешкоджає моделі знаходити приховані залежності між ознаками. Більше того, у традиційних підходах складні ознаки залишаються необробленими, тобто модель працює з даними поверхнево, що знижує її здатність до точного виявлення атак. Третьою важливою проблемою є неефективність багатьох існуючих систем виявлення, які, хоч і здатні досягати високої точності завдяки об'єднанню кількох моделей, проте потребують багато часу на обробку та мають затримки у прийнятті рішень. Такий компроміс між точністю і швидкодією є неприйнятним у сценаріях великих даних, де критичним є забезпечення оперативного реагування на загрози.

Для вирішення вищезазначених проблем у даній роботі запропоновано використати підхід до виявлення мережевих атак, який поєднує еволюційні обчислення та методи машинного навчання. Основна ідея полягає в тому, щоб поетапно обробити великі обсяги трафіку, видобути з них найінформативніші ознаки та класифікувати загрози з високою точністю та мінімальними затримками. Для попередньої обробки даних використано підхід, який включає наступні етапи: очищення інформації, вирівнювання класів за допомогою алгоритму RandomSample-SMOTE та уникнення перенавчання або недонавчання моделі на нерівномірно представлених вибірках. Далі потрібно реалізувати ієрархічний підхід до видобування ознак, у якому буде враховуватися, що двокласова (атака/не атака) і багатокласова (типи атак) класифікація мають різну природу. Для кожного з цих завдань застосовано окрему стратегію відбору ознак, що дозволяє зменшити розмірність і підвищити точність.

В рамках даної кваліфікаційної роботи буде розроблено багаторівневу модель класифікації трафіку, яка спочатку виконуватиме грубу двокласову класифікацію з метою швидкої фільтрації потенційно небезпечного трафіку, а потім – більш детальну багатокласову класифікацію для визначення конкретного типу атаки. Це дозволить значно підвищити точність розпізнавання та зменшити кількість хибнопозитивних спрацювань.

Також, потрібно розробити модель еволюційного навчання на основі ансамбля моделей LightGBM, яка поєднує у собі ефективні алгоритми вибору ознак і стратегії розділення класів. Перша модель у цій структурі відповідає за базову двокласову класифікацію, тоді як друга –за деталізовану багатокласову. Такий підхід дозволить не лише досягти високої точності виявлення атак, а й мінімізувати обчислювальні витрати та затримки при прийнятті рішень, що є особливо важливим для роботи з великими потоками мережевого трафіку в режимі реального часу.

Метою кваліфікаційної роботи є розробка методу виявлення мережевих атак у трафіку великих даних на основі поєднання еволюційних обчислень та методів машинного навчання з метою підвищення точності, ефективності та швидкодії в умовах високої розмірності та багатовимірності даних.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

– провести аналіз сучасних методів виявлення мережевих атак;

– виявити основні проблеми, пов’язані з обробкою високовимірного трафіку;

– розробити ефективну процедуру попередньої обробки даних;

– запропонувати ієрархічний підхід до вибору ознак для двокласової та багатокласової класифікації;

– реалізувати двошарову модель класифікації мережевого трафіку з використанням LightGBM;

– провести експериментальні дослідження з використанням відкритих наборів даних;

– порівняти отримані результати з існуючими підходами.

2 МЕТОД ВИЯВЛЕННЯ МЕРЕЖЕВИХ АТАК У ТРАФІКУ ВЕЛИКИХ ДАНИХ НА ОСНОВІ МАШИННОГО НАВЧАННЯ ТА ЕВОЛЮЦІЙНИХ ОБЧИСЛЕНЬ

2.1 Концепція методу виявлення атак у трафіку великих даних

У сучасних інформаційних системах, які функціонують у середовищі великих даних, зростає потреба у високоефективних методах виявлення атак, здатних обробляти великі обсяги мережевого трафіку у режимі реального часу. Для цього необхідно поєднати точність і масштабованість аналітичних моделей із достатньою швидкістю прийняття рішень. У межах цієї роботи запропоновано концепцію методу виявлення атак, яка базується на поєднанні машинного навчання з еволюційними обчисленнями. Такий підхід дозволяє ефективно працювати із багатовимірними, гетерогенними та високошвидкісними потоками трафіку, забезпечуючи адаптивне навчання моделі, виділення релевантних ознак та точну класифікацію загроз.

Запропонована модель має ієрархічну архітектуру, яка складається з п’яти послідовних рівнів (компонентів), кожен з яких виконує спеціалізовану функцію в рамках загального процесу виявлення атак:

1. Рівень попередньої обробки трафіку великих даних відповідає за приведення вхідного потоку даних до уніфікованого вигляду. На цьому етапі здійснюється очищення від шуму, заповнення пропущених значень, видалення неінформативних ознак, нормалізація числових параметрів та балансування класів шляхом синтетичного надсемплінгу (наприклад, із використанням методу RandomSample-SMOTE). Це забезпечує високу якість вхідних даних для подальшого аналізу.

2. Рівень вибору ознак для двокласової класифікації реалізує автоматизовану процедуру відбору найбільш інформативних параметрів для класифікації трафіку на «атака» / «не атака». Вибір ознак ґрунтується на градієнтному аналізі важливості в моделі LightGBM, що дозволяє значно зменшити розмірність простору ознак і підвищити швидкість обробки.

3. Рівень навчання моделі для двокласової класифікації формує першу модель виявлення атак, яка виконує грубе сортування трафіку за двома класами: шкідливий та безпечний. На цьому рівні реалізується базова логіка фільтрації трафіку, де використовується стратегія нульової суми — кожен зразок належить або до класу атак, або до класу нормального трафіку. Модель тренується з використанням регуляризації (L2) та методів крос-валідації, що запобігає перенавчанню та забезпечує стабільність роботи.

4. Рівень вибору ознак для багатокласової класифікації застосовується до підмножини трафіку, який був класифікований як атакувальний. На цьому етапі повторно виконується вибір ознак, однак уже з урахуванням завдань розрізнення типів атак. Враховуючи іншу природу класифікації, набір ознак формується окремо, з використанням аналогічного механізму градієнтного аналізу в LightGBM, що дозволяє виділити ключові характеристики для кожного класу атак.

5. Рівень навчання моделі для багатокласової класифікації завершує структуру моделі. Тут будується спеціалізований класифікатор, здатний визначити конкретний тип атаки (наприклад, DoS, фішинг, SQL-ін’єкція тощо) серед тих, що були виявлені на попередньому рівні. Модель також реалізується на основі LightGBM, що дозволяє поєднувати високу точність із мінімальними затримками обробки.

6. Таким чином, концепція методу передбачає розділення процесу виявлення атак на кілька послідовних етапів із вузькою спеціалізацією, що забезпечує адаптивність, масштабованість та ефективність. Особливістю підходу є те, що на кожному рівні відбувається цілеспрямоване зниження складності задачі – як за рахунок відбору ознак, так і за рахунок поетапної класифікації, що суттєво підвищує загальну якість роботи системи.

Загальна схема запропонованого підходу подана на рисунку 2.1, де показано взаємозв’язки між компонентами та послідовність обробки трафіку на кожному з рівнів.

Рисунок 2.1 – Загальна схема запропонованого підходу

2.2 Попередня обробка даних

У сценаріях роботи з великими даними попередня обробка даних має вирішальне значення. Її головна мета – перетворити сирі, неструктуровані або забруднені дані на такий формат, який може бути використаний для навчання моделей машинного навчання. Основні етапи попередньої обробки включають:

– обробку символів і відсутніх значень;

– балансування класів у вибірці**;**

– нормалізацію даних**;**

– розбиття набору даних на частини**.**

У наборі даних можуть бути наявні невпізнавані символи або значення, які не піддаються обробці. Наприклад, символи "NaN" (Not a Number) чи "Inf" (нескінченність) можуть спричинити помилки при аналізі. Такі значення необхідно знайти і замінити. Зазвичай їх замінюють на середнє значення відповідного стовпця, яке розраховується за формулою:

$$\bar{x} = \frac{1}{m}\sum_{i=1}^{m} C_i, \tag{2.1}$$

де $\bar{x}$ – середнє значення в стовпці,

m – кількість валідних значень у поточному стовпці,

C_i – значення i -го елемента у відповідному стовпці.

Якщо стовпець містить лише порожні значення або лише нулі, такий стовпець вважається неінформативним і підлягає видаленню.

Для видалення неінформативних стовпців проводиться статистичний аналіз кожного стовпця у наборі даних. У процесі аналізу виявляються стовпці, які:

– складаються лише з нулів,

– повністю порожні,

– є дублікатами інших стовпців.

Такі стовпці видаляються, оскільки вони не несуть жодної корисної інформації для моделі і лише ускладнюють навчання.

Однією з важливих проблем у роботі з великими наборами даних є дисбаланс між класами – коли кількість прикладів одного класу значно перевищує кількість прикладів іншого. Це може призводити до того, що модель ігнорує менш представлений клас і дає упереджені результати.

Щоб вирішити цю проблему, проводиться кілька послідовних дій.

Статистичний аналіз кожного зразка. Якщо виявляються зразки, що належать до надто рідкісних класів (наприклад, одиничні випадки), їх видаляють як "шумові" дані, що можуть зашкодити навчанню.

Випадкове зменшення кількості зразків більшості (random down sampling): з класу, який має занадто багато зразків, випадковим чином відбирається обмежена кількість прикладів. Це дозволяє зменшити перекіс у бік більшості. Формула випадкового вибіркового відбору подана у рівнянні:

$$P_{new} = \sum_{i}^{Number} F.sample(m), \tag{2.2}$$

де $Number$ – означає кількість відібраних зразків,

F – випадково вибрана множина з m елементів з загальної вибірки.

Синтетичне збільшення вибірки меншості за допомогою методу SMOTE (Synthetic Minority Over-sampling Technique). Цей метод створює нові синтетичні зразки на основі існуючих, щоб збільшити кількість прикладів рідкісного класу.

Формула SMOTE наведена в наступному рівнянні:

$$F_n new = F_i + \xi * \{F_i(n) - F_i\}, \tag{2.3}$$

де $F_n new$ – це синтезований новий запис трафіку, який буде додано до класу меншості;

F_i – будь-який наявний зразок з класу меншості;

$F_i(n)$ – сусідній зразок до F_i, обраний з числа найближчих сусідів;

ξ – випадкове число в діапазоні від 0 до 1, яке визначає, де саме між F_i та $F_i(n)$ буде створено новий зразок.

Таким чином, новий зразок створюється як точка на відрізку між двома близькими зразками меншості, злегка зміщена випадковим чином. Це дозволяє розширити вибірку меншості новими, подібними, але не ідентичними зразками, що покращує баланс між класами і загальну здатність моделі до класифікації.

Після обробки класів проводиться перегрупування міток. На основі початкових категорій та потреб класифікаційної моделі, мітки класів об’єднуються у дві множини:

BL – мітки для двокласової класифікації (наприклад, "атака" або "нормальний трафік");

ML – мітки для багатокласової класифікації (наприклад, різні типи атак: DoS, Probe, R2L тощо).

У результаті маємо структуру: $Label = \{BL, ML\}$.

На етапі попередньої обробки всі дані ознак (атрибутів) проходять процес нормалізації, щоб зробити їх придатними для обробки алгоритмами машинного навчання. Це важливо, оскільки різні ознаки можуть мати різні одиниці вимірювання або масштаб значень, що ускладнює навчання моделей.

Спочатку всі категоріальні ознаки (наприклад, тип протоколу, стан з'єднання) кодуються за допомогою методу "one-hot" кодування. Це означає, що кожна категорія перетворюється на окремий стовпець із двійковими значеннями: 1, якщо категорія присутня, і 0 – якщо відсутня.

Потім усі числові ознаки нормалізуються до діапазону від 0 до 1 за допомогою методу мін-макс нормалізації. Це забезпечує рівнозначний внесок кожної ознаки у модель. Формула нормалізації наступна:

$$C_j new = \frac{C_j - C_j min}{C_j max - C_j min}, \quad (2.4)$$

де C_j – початкове значення ознаки;

$C_j min$ – мінімальне значення цієї ознаки в усій вибірці;

$C_j max$ – максимальне значення цієї ознаки;

$C_j max$ – нормалізоване значення ознаки.

Після завершення всіх етапів обробки, набір даних розподіляється на три частини відповідно до потреб моделі:

– навчальна вибірка (training set) – для навчання моделі;

– валідаційна вибірка (validation set) — для налаштування параметрів і вибору найкращої моделі;

– тестова вибірка (test set) — для перевірки ефективності навченої моделі на нових даних.

Параметри експериментального середовища представлені в таблиці 2.1.

Таблиця 2.1 – Параметри експериментального середовища

Платформа	Параметр	Конфігурація
Апаратне забезпечення	Процесор (CPU)	AMD EPYC 7T83
	Жорсткий диск (SSD)	500 ГБ
	Оперативна пам’ять	32 ГБ
Програмне забезпечення	Операційна система	Ubuntu 22.04
	Python	Версія 3.9.13
	Scikit-learn	Версія 1.1.1
	LightGBM	Версія 3.2.1

2.3 Вибір ознак на основі LightGBM

У даній роботі для вибору найважливіших ознак використовується модель LightGBM – одна з найефективніших реалізацій градієнтного бустингу. Процес вибору ознак виконується на двох рівнях:

– вибір ознак для двокласової класифікації (наприклад, "атака" або "нормальний трафік");

– вибір ознак для багатокласової класифікації (для різних типів атак).

На відміну від інших моделей градієнтного бустингу, таких як XGBoost чи CatBoost, LightGBM використовує спеціальний алгоритм під назвою GOSS (Gradient-based One-Side Sampling) [30], який дозволяє прискорити вибір ознак, зменшуючи обсяг оброблюваних даних без втрати точності.

Як працює вибір ознак у LightGBM?

1. Для кожної ознаки модель обчислює градієнти похибок для кожного зразка у навчальному наборі. Ці градієнти показують, наскільки сильно кожен зразок впливає на помилку моделі.

2. Усі дані впорядковуються за модулем градієнта у порядку спадання.

3. З цього впорядкованого списку вибирається:

P – верхні $a \times 100\%$ прикладів із найбільшими градієнтами (найінформативніші дані);

Q – випадкові $b \times 100\%$ прикладів із решти менш інформативних зразків.

Потім множини P і Q об'єднуються, і з них обчислюється приріст інформації (gain) – метрика, яка показує, наскільки ефективно ця ознака ділить дані.

Формули приросту інформації (gain):

Загальний приріст для ознаки f у точці поділу i обчислюється як:

$$G_f(i) = \frac{1}{S}\left(G_f^1(i) + G_f^2(i)\right) \tag{2.5}$$

де $G_f^1(i)$ – внесок від лівої підмножини (після поділу),

$G_f^2(i)$ – внесок від правої підмножини,

S – загальна кількість вибраних зразків.

Обчислення приросту для лівої частини:

$$G_f^1(i) = \frac{1}{S_l(i)}\left(\sum_{X_k \in P_l} g_k + \frac{1-a}{b}\sum_{X_k \in Q_l} g_k\right)^2 \tag{2.6}$$

Аналогічно для правої частини:

$$G_f^2(i) = \frac{1}{S_h(i)}\left(\sum_{X_k \in P_h} g_k + \frac{1-a}{b}\sum_{X_k \in Q_h} g_k\right)^2 \tag{2.7}$$

де P_l і Q_l – підмножини P і Q, що потрапили до лівої частини після поділу;

P_h і Q_h – підмножини P і Q, що потрапили до правої частини;

g_k – градієнт похибки для зразка X_k;

$S_l(i)$, $S_h(i)$ – кількість зразків у відповідній підмножині після поділу в точці i;

a і b – коефіцієнти, які визначають розмір підмножин P і Q.

Ці формули дозволяють ефективно оцінити значущість кожної ознаки і відібрати лише ті, які найбільше впливають на розділення класів. Завдяки цьому зменшується розмірність даних і підвищується продуктивність моделі.

Таким чином, LightGBM фокусується на слабо навчених зразках (які мають великі градієнти) і ігнорує зразки, які вже добре вивчені (з малими градієнтами).

Це дозволяє моделі точніше оцінювати "gain" (виграш) навіть при меншій кількості навчальних прикладів, що підвищує ефективність обробки великих обсягів даних.

Для подальшого підвищення ефективності, LightGBM використовує:

– гістограмний підхід для пошуку найкращих точок поділу (див. рисунок 3);

– алгоритм зростання за листками (leaf-wise growth), який є ефективнішим за рівне зростання по рівнях (level-wise);

– обмеження глибини дерева, що допомагає уникнути перенавчання і зменшує витрати пам’яті.

Як працює гістограмний підхід?

Кожну одновимірну ознаку розбивають на кілька діапазонів значень – "бінів" (від англ. *bins*).

Для кожного біна формується гістограма, де зберігається:

– кількість зразків у цьому біні;

– сума градієнтів цих зразків.

Для багатовимірних наборів даних LightGBM створює окремі гістограми для кожної ознаки.

Потім відбувається пошук найкращої точки поділу для вузлів дерева саме через аналіз гістограм, що прискорює процес побудови дерева.

Після побудови дерева, LightGBM дозволяє оцінити важливість кожної ознаки. Для цього обчислюється внесок кожної ознаки в процес класифікації для кожного класу трафіку.

Усі ознаки сортуються за спаданням важливості, і з них по черзі відбираються топ-k найважливіших, які й подаються на вхід базовій моделі LightGBM для повторного навчання.

Поступово збільшуючи кількість k, дослідники обирають таке значення, при якому модель досягає максимальної точності класифікації. У підсумку ці k ознак використовуються для навчання моделей на відповідному шарі.

Алгоритм вибору ознак для кожного шару описано на рисунку 2.2.

```
Вхід:
    D – навчальна вибірка
    a, b – параметри GOSS-алгоритму
    Kmax – максимальна кількість ознак для перевірки
    Модель – базова модель LightGBM

Вихід:
    k₁ – оптимальна кількість ознак для двокласової класифікації
    k₂ – оптимальна кількість ознак для багатокласової класифікації
    F₁ – вибрані ознаки для двокласової класифікації
    F₂ – вибрані ознаки для багатокласової класифікації

1. // --- Вибір ознак для двокласової класифікації ---
2. Обчислити важливість ознак за допомогою LightGBM на основі GOSS
3. Відсортувати всі ознаки за спаданням важливості → список F_sorted
4. Для k від 1 до Kmax:
       a. Вибрати топ-k ознак з F_sorted → F_tmp
       b. Навчити модель на F_tmp
       c. Обчислити точність моделі
5. Обрати значення k = k₁, при якому точність максимальна
6. Зберегти відповідний набір ознак: F₁ = топ-k₁ ознак
7. Побудувати модель двокласової класифікації на основі F₁
8. Отримати підмножину атакувального трафіку D_attack з результатів класифікації

9. // --- Вибір ознак для багатокласової класифікації ---
10. Повторити кроки 2–6 для D_attack
11. Зберегти результат: F₂ = топ-k₂ ознак
12. Побудувати модель багатокласової класифікації на основі F₂

Кінець алгоритму
```

Рисунок 2.2 – Алгоритм вибору ознак на основі моделі LightGBM для двох рівнів класифікації (двокласової та багатокласової)

Спочатку навчальна вибірка подається у шар вибору ознак для двокласової класифікації, де відбирається k_1 найкращих ознак. Потім модель виконує двокласову класифікацію трафіку. Трафік, який було правильно класифіковано як атакувальний, передається далі у шар вибору ознак для багатокласової

класифікації, де обирається k_2 ознак. Ці ознаки потім використовуються для побудови моделі, яка розпізнає конкретні типи атак.

Побудова гістограми та дерева в LightGBM показано на рисунку 2.3.

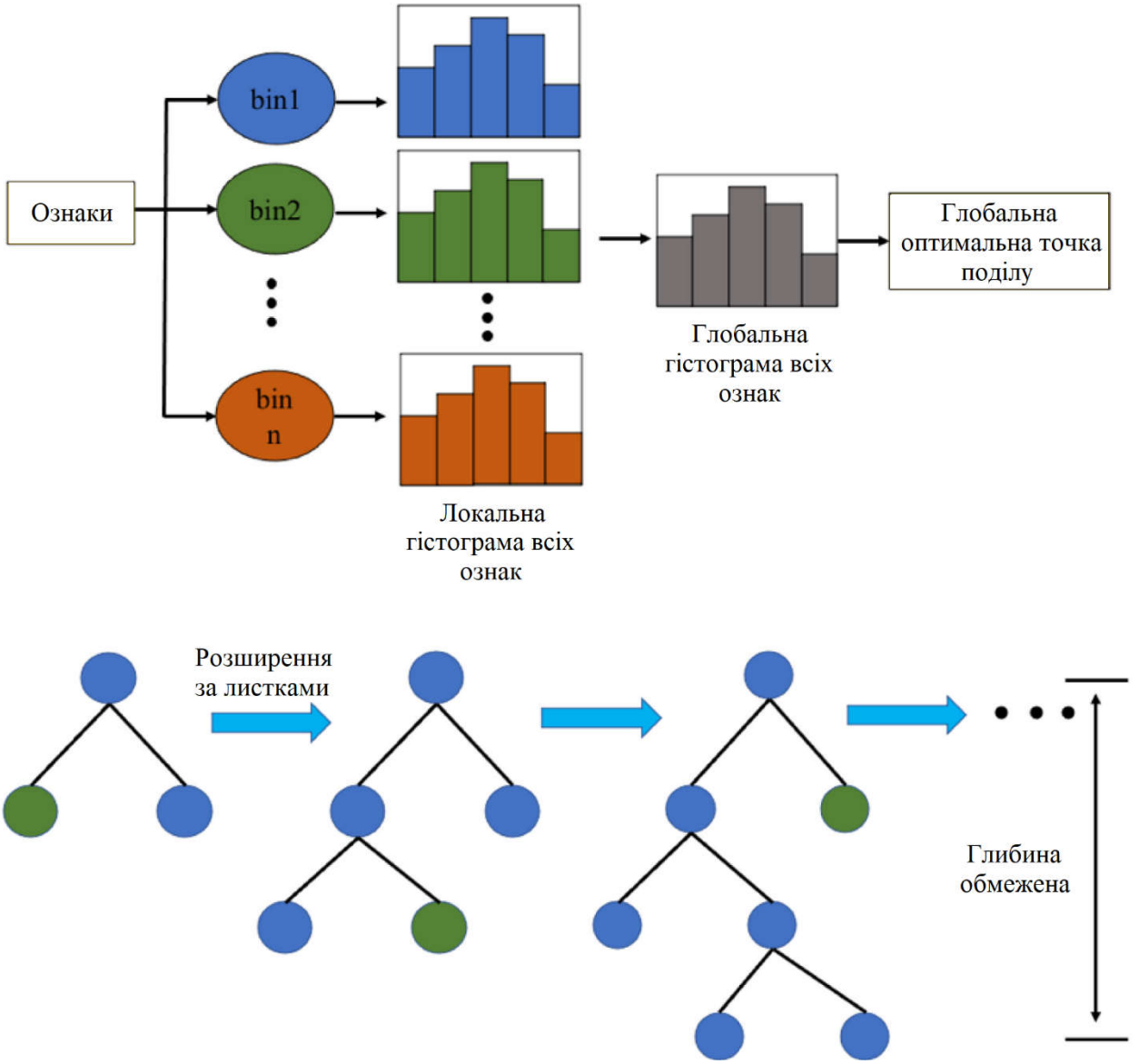

Рисунок 2.3 – Побудова гістограми та дерева в LightGBM

На рисунку 2.4 подано розподіл міток і пов’язаних із ними ознак у наборі даних.

На рисунку 2.4 представлено розподіл міток у наборі даних та їх зв’язок із відповідними ознаками, що використовуються для класифікації мережевого трафіку. Зображення ілюструє, як після етапу попередньої обробки дані поділяються на дві основні категорії: безпечний трафік (benign) та атакувальний

трафік (attack). Для кожної з категорій у процесі відбору ознак визначаються релевантні характеристики, які найбільше впливають на рішення моделі.

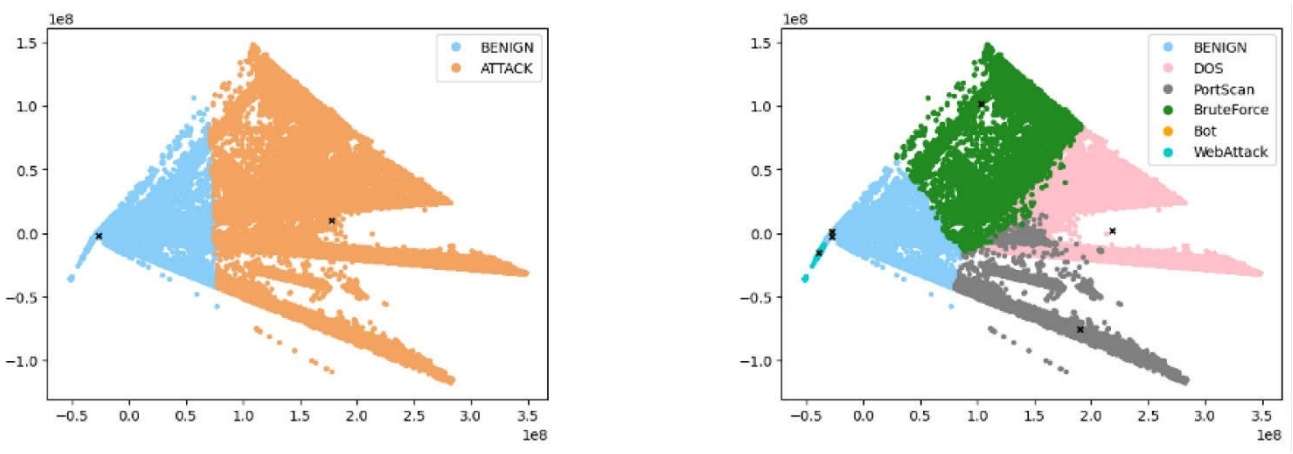

Рисунок 2.4 – Розподіл міток і пов’язаних із ними ознак у наборі даних

Атакувальний трафік, у свою чергу, класифікується на кілька підтипів, зокрема: DoS-атаки, бот-атаки, сканування портів, веб-атаки та атаки типу brute force. Для кожного типу атак побудовано окремий вектор ознак, що враховує специфіку характерної активності у трафіку.

Таке групування ознак дозволяє моделі не лише виявити факт атаки, а й точніше ідентифікувати її тип. Крім того, завдяки розділенню процесу класифікації на дві стадії – спочатку грубу двокласову, а потім деталізовану багатокласову – підвищується загальна точність і стійкість системи до хибних спрацьовувань.

2.4 Дворівнева модель класифікації атак

У задачах виявлення атак у трафіку великих даних дуже важливо досягти максимально швидкої та точної реакції. Оскільки обсяг трафіку надзвичайно великий, ефективність обробки моделі напряму впливає на здатність вчасно реагувати на загрози.

У запропонованій моделі використано алгоритм LightGBM, який реалізує стратегію бустингу – тобто поєднання кількох слабких моделей (слабих

класифікаторів) у єдину потужну модель. Такий підхід дозволяє суттєво підвищити здатність моделі до узагальнення (тобто її здатність працювати на нових, раніше невідомих даних) і знизити помилки класифікації.

Окрім цього, LightGBM використовує стратегію зростання дерева за листками (leaf-wise growth), що дозволяє зменшити похибку моделі та покращити точність розпізнавання.

Таким чином, у роботі застосовується двошарова модель виявлення трафіку з використанням LightGBM:

На першому рівні LightGBM виконує класифікацію трафіку на два класи:

– атака;

– безпечний (нормальний) трафік.

У цьому шарі реалізується стратегія нульової суми, відповідно до якої кожен фрагмент трафіку повинен бути однозначно класифікований як або безпечний, або шкідливий. Такий підхід дозволяє краще виявляти нові, раніше невідомі типи атак, оскільки навіть не схожі на навчальні дані зразки можуть бути віднесені до класу "атака" у разі незначної схожості.

Далі трафік, який був класифікований як атакувальний, передається у другий шар класифікації, де LightGBM використовується для розподілу між конкретними типами атак. Наприклад, це можуть бути такі класи: DoS-атака, фішинг, сканування портів, проникнення в систему тощо.

Таким чином, модель виконує спочатку грубе фільтрування на рівні "атака/не атака", а потім – детальну класифікацію типу атаки, що дозволяє досягти високої точності та швидкості.

Щоб уникнути перенавчання (overfitting), у кожному шарі використовуються:

– L2-регуляризація, яка допомагає зменшити складність моделі та забезпечити кращу узагальнюваність;

– стратегія ранньої зупинки – якщо після кількох ітерацій якість моделі не покращується, процес навчання автоматично зупиняється.

Для налаштування гіперпараметрів (наприклад, кількість дерев, глибина дерева, коефіцієнт регуляризації) використовується метод TuneGridSearchCV –

сітковий пошук з крос-валідацією. Під час тренування модель перевіряється за допомогою 5-кратної крос-валідації, що дозволяє переконатися у стабільності та надійності моделі на різних частинах вибірки.

3 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ МЕТОДУ ВИЯВЛЕННЯ МЕРЕЖЕВИХ АТАК

3.1 Методологія проведення експериментальних досліджень

3.1.1 Попередня обробка даних

У якості основи для моделювання та тестування було обрано набір даних CICIDS2017, що є одним із найвідоміших відкритих датасетів для аналізу мережевих атак. Цей набір містить реальні мережеві дані, охоплює всі основні протоколи, імітує взаємодії між хостами в умовах реальної інфраструктури та включає приклади як безпечного, так і атакувального трафіку. У CICIDS2017 представлені такі типи атак, як DoS, brute force, web-атаки, сканування портів тощо, що повністю відповідає вимогам до експериментального середовища в умовах великих даних [31].

На етапі попередньої обробки дані було очищено відповідно до логіки, закладеної у шар попередньої обробки (див. параграф 2.2). Зокрема:

1. Заміна недопустимих символів (NaN, Inf) на середнє значення стовпця:
   – у стовпці *Flow Bytes/s* виявлено 1358 значень NaN та 1509 – Inf;
   – у стовпці *Flow Packets/s* знайдено 2867 значень Inf.
2. Видалено непотрібні або дубльовані стовпці: зокрема, стовпці 31, 33, 56–61, що містили лише нулі або повторювали інші.
3. Об’єднано та очищено категорії: категорії *Infiltration*, *Web Attack SQL Injection* та *Heartbleed* вилучено як шумові (занадто малі вибірки).
4. Дані об'єднано у форматі CSV.

Для візуального аналізу було застосовано метод головних компонент (PCA) з метою перевірки розподілу міток і відповідних ознак. Результати, представлені на рисунку 2.4, показали чітке відокремлення безпечного та атакувального трафіку, хоча деякі типи атак (наприклад, *Web Attack*) мають схожі ознаки із звичайним трафіком, а *Bot*-атаки мають надто малу кількість прикладів для якісної візуалізації.

Далі мітки були перегруповані:

– для двокласової класифікації (BL): усі атаки об’єднано в клас Attack, а звичайний трафік у клас Benign.

– для багатокласової класифікації (ML): збережено окремі мітки для типів атак (DoS, BruteForce, WebAttack, PortScan, Bot).

Було проведено балансування вибірки:

– для класів більшості застосовано випадкову вибірку (down-sampling);

– для класів меншості – синтетичне доповнення даних методом SMOTE.

Кількість зразків після обробки наведено в таблиці 3.1.

Таблиця 3.1 – Кількість зразків після балансування вибірки

Початкова мітка	Початкова кількість	BL мітка	К-сть (BL)	ML мітка	К-сть (ML)
BENIGN	2 273 097	BENIGN	150 000	Benign	150 000
DDoS	128 027	Attack	345 000	DoS	150 000
DoS GoldenEye	10 293				
DoS Slowloris	5 796				
DoS Slowhttptest	5 499				
DoS Hulk	231 073				
PortScan	158 930			PortScan	150 000
FTP-Patator	7 938			BruteForce	15 000
SSH-Patator	5 897				
Web Attack Brute Force	1 507			WebAttack	15 000
Web Attack XSS	36				
Bot	1 966			Bot	15 000

Після завершення обробки дані були закодовані за допомогою one-hot кодування, нормалізовані та розділені на навчальну, валідаційну та тестову вибірки. Навчальна вибірка використовувалася для побудови моделі, а валідаційна – для пошуку оптимальних гіперпараметрів за допомогою методу

TuneGridSearchCV. Завдяки цьому гіперпараметри моделі, що використовуються на тестовому етапі, відповідають найкращим знайденим значенням. Після завершення навчання модель була протестована на тестовій вибірці, а результати оцінено за кількома метриками ефективності.

3.1.2 Вибір ознак

У середовищі великих даних ефективний вибір ознак відіграє ключову роль у боротьбі з так званим «прокляттям розмірності», коли надмірна кількість ознак не лише ускладнює навчання моделей, але й знижує точність виявлення. У запропонованому підході вибір ознак реалізовано відповідно до ієрархічної структури моделі, що передбачає окреме вилучення інформативних параметрів для задачі двокласової та багатокласової класифікації.

1. Вибір ознак для двокласової класифікації.

На першому етапі навчальну вибірку передано до шару вибору ознак для двокласової класифікації, де відбувається визначення значущості кожної ознаки на основі базової моделі LightGBM. Лише обмежена кількість ознак справді впливає на результат класифікації, тоді як більшість інших можуть знижувати ефективність моделі.

Згідно з алгоритмом вибору ознак, модель по черзі тестувалася на наборах зростаючої кількості найважливіших ознак. Максимальна точність була досягнута при значенні $k = 16$, після чого темп покращення або знижувався, або залишався сталим. Відповідно, 16 найважливіших ознак було відібрано для побудови моделі двокласової класифікації. Їх перелік подано в таблиці 3.2.

2. Вибір ознак для багатокласової класифікації.

Зразки атакувального трафіку, правильно класифіковані попередньою моделлю, передаються до шару вибору ознак для багатокласової класифікації. Аналогічно до попереднього етапу, модель LightGBM формує рейтинг ознак на основі їх впливу на класифікацію за типами атак. Результати подано на рисунку 3.1, а вплив кількості ознак на точність багатокласової моделі – на рисунку 3.2.

Таблиця 3.2 – Обрані 16 ознак для двокласової класифікації

№	Назва ознаки	№	Назва ознаки
1	Flow Duration	9	Total Length of Fwd Packets
2	Destination Port	10	Fwd IAT Min
3	Flow IAT Min	11	PSH Flag Count
4	Bwd Packet Length Min	12	Flow IAT Mean
5	Flow Bytes/s	13	Packet Length Std
6	Flow IAT Max	14	Flow Packets/s
7	Init_Win_bytes_backward	15	Fwd Header Length
8	Init_Win_bytes_forward	16	Average Packet Size

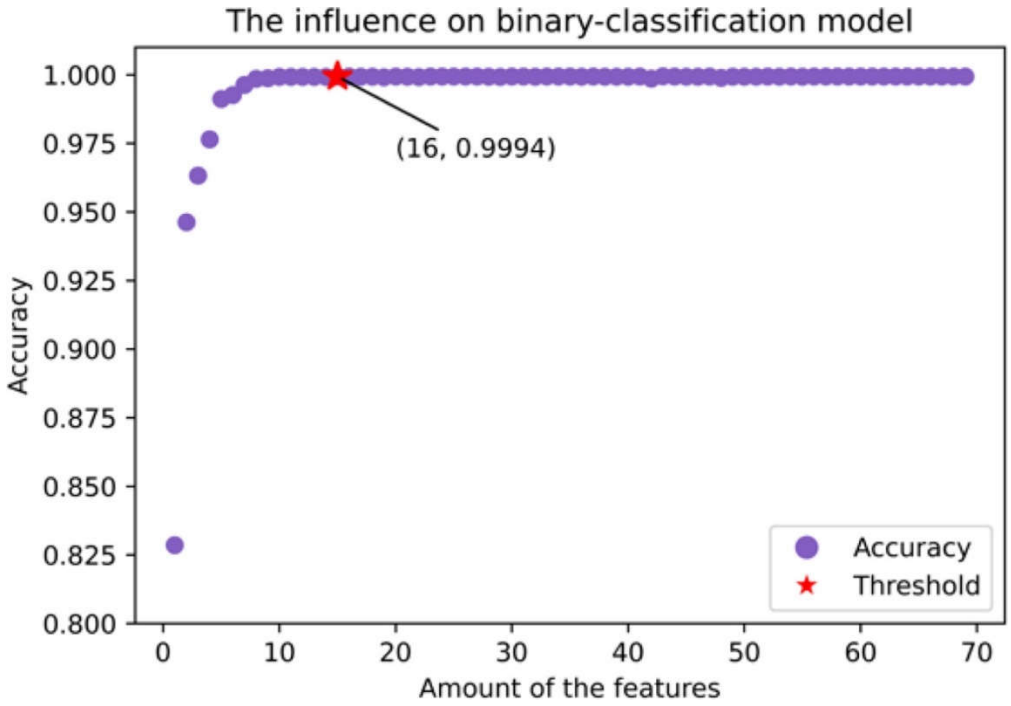

Рисунок 3.1 – Ранжування важливих ознак для багатокласової класифікації

Як видно з рисунку, оптимальний результат також досягнуто при 16 ознаках, що свідчить про збалансованість методології.

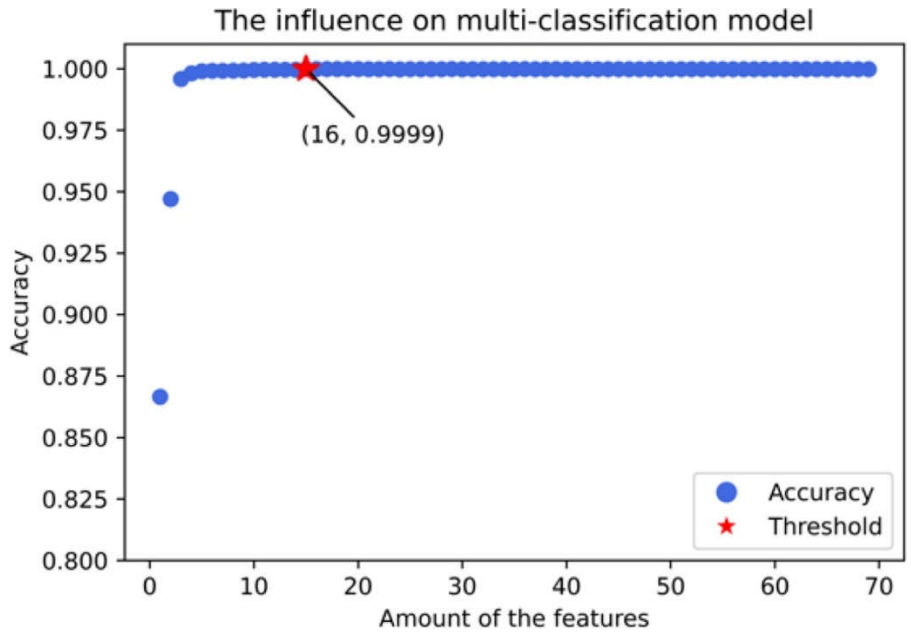

Рисунок 3.2 – Вплив кількості ознак на модель багатокласової класифікації

Порівняння таблиць 3.2 і 3.3 показує, що сім ознак є спільними для обох рівнів класифікації, що вказує на їхню високу дискримінативну здатність. Повний список відібраних ознак для багатокласової класифікації подано в таблиці 3.3. Повторювані ознаки виділено жирним шрифтом.

Таблиця 3.3 – Обрані 16 ознак для багатокласової класифікації

№	Назва ознаки	№	Назва ознаки
1	**Flow Duration**	9	Max Packet Length
2	**Destination Port**	10	Average Packet Size Fwd
3	**Flow IAT Min**	11	Flow IAT Variance
4	**Flow Bytes/s**	12	Bwd IAT Max
5	**Init_Win_bytes_forward**	13	Packet Length Min
6	**Average Packet Size**	14	Packet Length Mean
7	Total Backward Packets	15	**ACK Flag Count**
8	Flow IAT Std	16	Down/Up Ratio

На основі проведеного аналізу можна зробити висновок, що застосування ієрархічного підходу до вибору ознак є ефективним для обох типів класифікаційних задач. Це дозволяє зменшити розмірність, підвищити точність

та зменшити навантаження на обчислювальні ресурси без втрати інформативності вхідних даних.

3.1.3 Навчання моделі

Після завершення етапів попередньої обробки та вибору ознак, навчальні та валідаційні вибірки були очищені від другорядних ознак, після чого до моделі LightGBM передавались лише релевантні значення ознак. На першому етапі було здійснено навчання двокласової моделі LightGBM. Для забезпечення найкращих результатів класифікації гіперпараметри моделі було оптимізовано за допомогою методу сіткового пошуку. Для перевірки стабільності отриманих результатів використовувався метод п’ятиразової крос-валідації.

На рисунку 3.3 зображено значення логарифмічної функції втрат (logloss) протягом 200 ітерацій моделі двокласової класифікації.

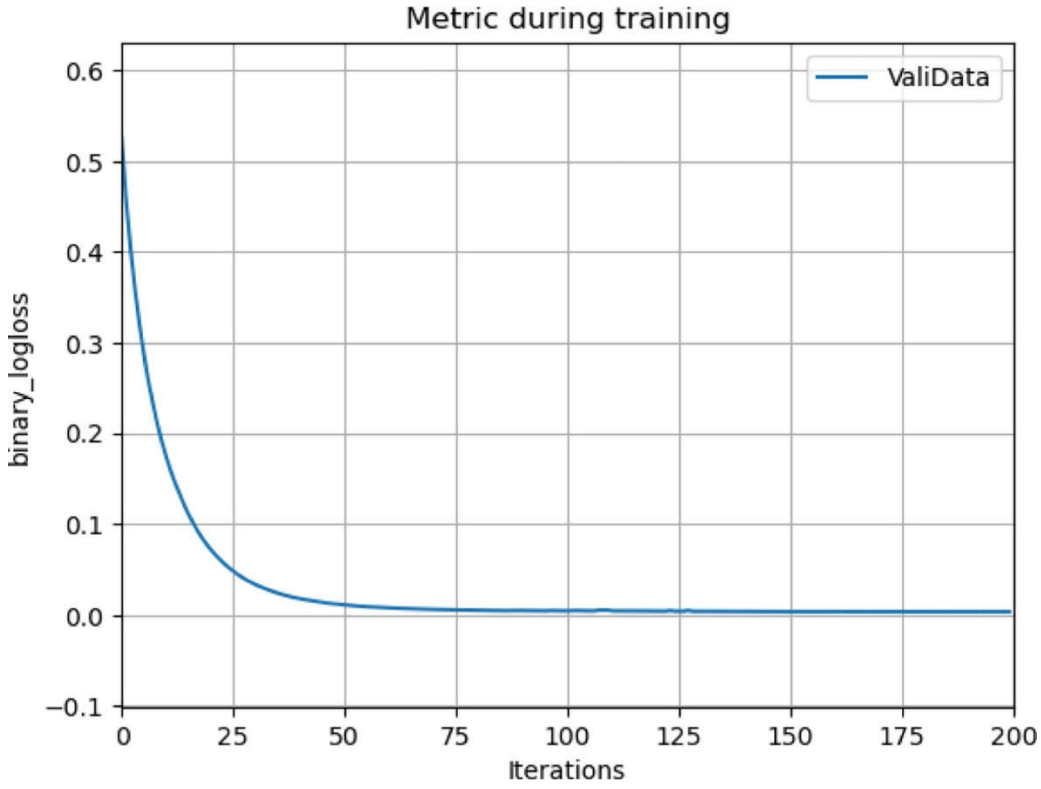

Рисунок 3.3 – Значення функції втрат протягом 200 ітерацій для моделі двокласової класифікації

З графіка видно, що модель швидко досягає збіжності, а значення втрат стабілізується на низькому рівні без значних коливань. Це свідчить про ефективне навчання та гарну узгодженість моделі з навчальними даними.

Далі, 241 402 зразки атакувального трафіку, які були правильно ідентифіковані двокласовою моделлю, було передано до шару вибору ознак багатокласової класифікації, після чого обрані ознаки використовувалися для навчання багатокласової моделі LightGBM.

На рисунку 3.4 наведено графік зміни logloss для багатокласової моделі протягом 200 ітерацій.

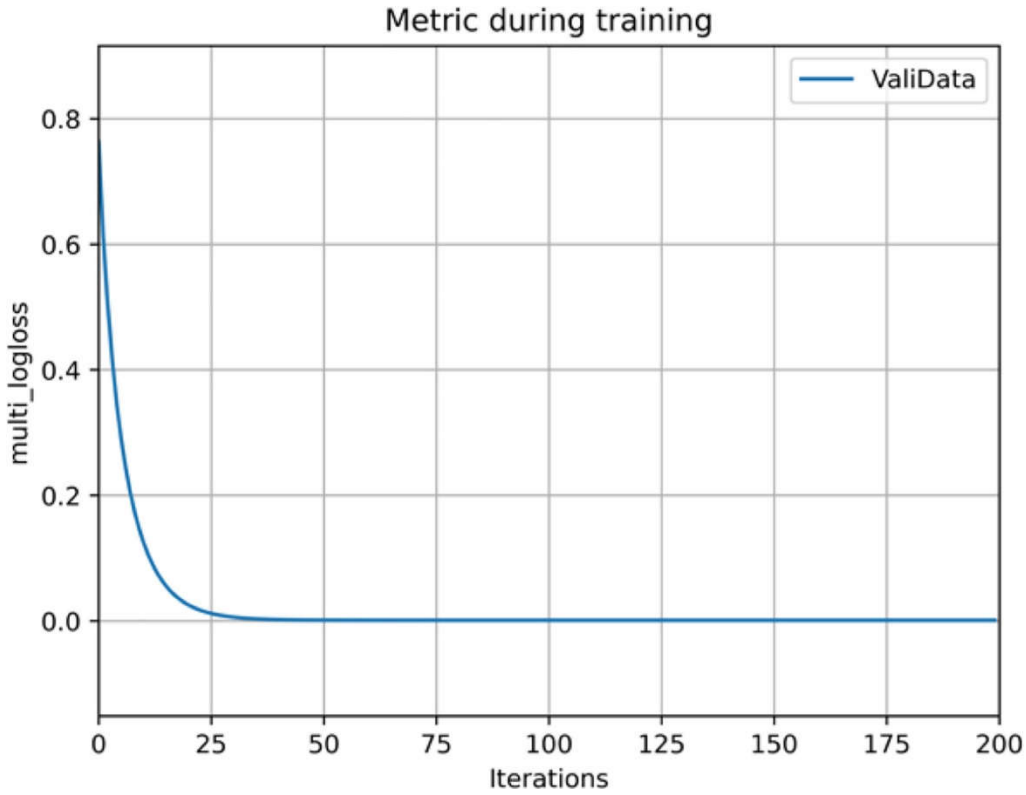

Рисунок 3.4 – Значення функції втрат протягом 200 ітерацій для моделі багатокласової класифікації

У порівнянні з рисунком 3.3, можна помітити, що багатокласова модель збігається швидше, що пояснюється двома факторами:

– по-перше, із навчального набору було вилучено трафік, що не є атакувальним, завдяки чому обсяг даних зменшився і прискорилося навчання;

– по-друге, ознаки різних типів атак мають більш виражені відмінності, що полегшує процес класифікації.

Це видно як за початковими значеннями втрат, так і за характером зниження logloss.

На рисунку 3.5 представлено приклад дерева рішень, згенерованого в багатокласовій моделі LightGBM. Як видно, глибина дерева є невеликою, а поділ вузлів добре збалансований, що підтверджує ефективність запропонованої моделі.

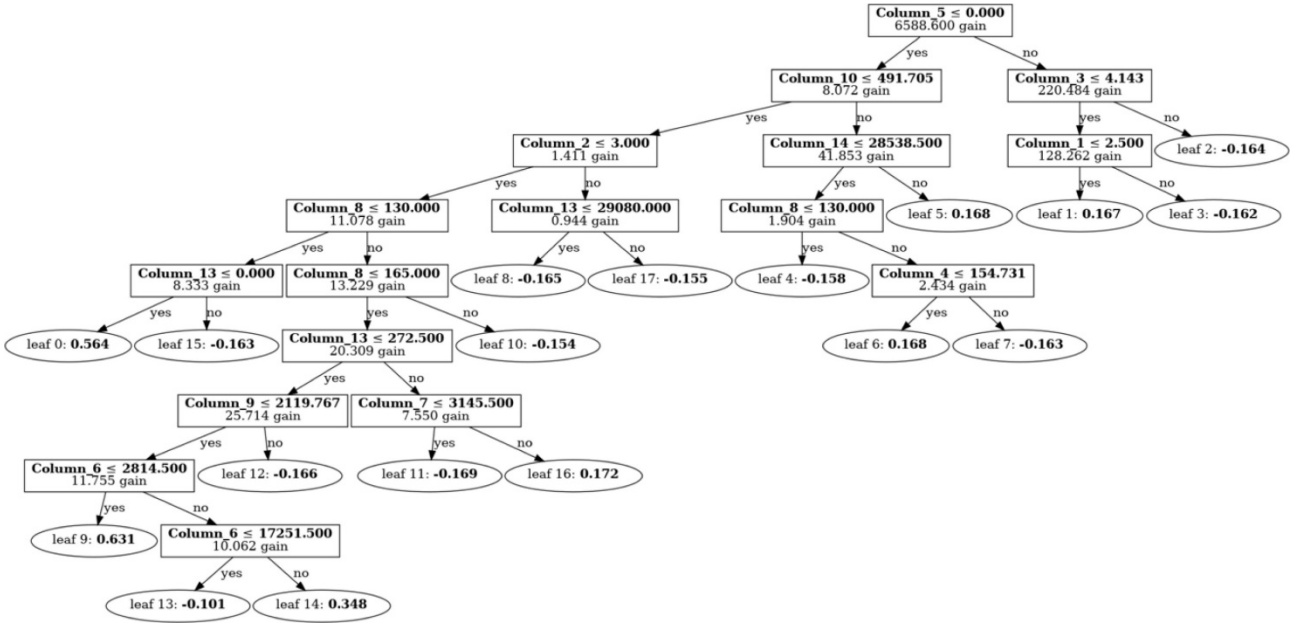

Рисунок 3.5 – Діаграма дерева рішень у моделі багатокласової класифікації

Узагальнюючи результати, можна стверджувати, що запропонований підхід демонструє високу ефективність у сценаріях великих даних, дозволяючи скоротити час навчання, знизити витрати обчислювальних ресурсів та забезпечити точне виявлення атак із мінімальними втратами інформації.

3.2 Аналіз результатів експериментальних досліджень

Для перевірки ефективності запропонованого методу було проведено тестування навченої моделі на тестовій вибірці, яка включає 30161 зразок атак типу DoS, 29922 атак типу PortScan, 3029 Web-атак, 3022 атак типу Bot, 2983 атак типу BruteForce, а також 29883 зразки безпечного (benign) трафіку.

Для порівняння результатів було сформовано контрольні групи моделей машинного та глибокого навчання:

– до моделей машинного навчання віднесено: дерево рішень (Decision Tree, DT), випадковий ліс (Random Forest, RF) та базову модель градієнтного бустингу (lightgbm);

– до моделей глибокого навчання було включено поширені архітектури згорткових нейронних мереж (CNN) та рекурентних нейронних мереж типу LSTM.

Однак, оскільки CNN краще працює з просторовими залежностями (наприклад, зображення), а LSTM – із послідовними даними (наприклад, часові ряди), то окреме використання цих моделей не дозволило досягти задовільних результатів: моделі не змогли належно адаптуватися до складної природи мережевих атак і продемонстрували низьку якість розпізнавання (через ефект недонавчання). З огляду на це, у роботі для порівняння також застосовано комбіновану модель CNN-LSTM, що дозволяє одночасно враховувати просторові та часові характеристики мережевого трафіку, і є популярною у сучасних дослідженнях.

Для оцінки якості роботи моделей використано стандартні метрики класифікації, які широко застосовуються у сфері машинного навчання [14]. У цьому дослідженні вони лише коротко охарактеризовані:

– Accuracy (точність) – загальна частка правильно класифікованих зразків (атаки + benign);

– Precision (прецизійність) – здатність моделі правильно розпізнавати негативні приклади (тобто, не помилково визначати benign як атаку);

– Recall (повнота) – здатність моделі правильно виявляти всі наявні зразки атак (істинно позитивні класи);

– F1-score – зважене гармонійне середнє між Precision і Recall, що враховує як точність, так і повноту класифікації;

– Inference time (час інференсу) – час, витрачений моделлю на класифікацію всієї тестової вибірки.

У таблиці 3.4 наведено порівняльні результати експериментів для задачі двокласової класифікації, де трафік поділяється на два класи: безпечний та атакувальний. Всі моделі навчались на однаково підготовленому датасеті, який

було очищено та збалансовано шляхом обробки класового дисбалансу. Це дозволяє здійснити коректне порівняння ефективності моделей за ключовими метриками: точність (Accuracy), прецизійність (Precision), повнота (Recall), F1-міра та час інференсу (Time).

Таблиця 3.4 – Порівняння результатів експериментів із двокласової класифікації

Метод	Клас	Precision	Recall	F1-score	Accuracy	Час (с)
Запропонована модель	BENIGN	0.9987	0.9965	0.9976	0.9985	0.0189
	ATTACK	0.9985	0.9994	0.9989		
DT	BENIGN	0.9987	0.9992	0.9989	0.9985	0.0264
	ATTACK	0.9981	0.9971	0.9976		
RF	BENIGN	0.9982	0.9981	0.9982	0.9989	0.1274
	ATTACK	0.9992	0.9992	0.9992		
LightGBM	BENIGN	0.9997	0.9982	0.9989	0.9993	0.0464
	ATTACK	0.9992	0.9998	0.9995		
CNN-LSTM	BENIGN	0.9157	0.7978	0.8527	0.9168	0.2080
	ATTACK	0.9171	0.9682	0.9420		

Всі моделі машинного навчання досягли високих результатів – понад 99.8% точності, що підтверджує ефективність реалізованого в роботі методу попередньої обробки. Водночас запропонована модель демонструє кращий компроміс між точністю та швидкістю роботи.

Модель LightGBM, побудована без ієрархічного підходу, досягла трохи вищої точності (99.93%), проте її час інференсу (0.0464 с) майже в 2,5 рази перевищує час, необхідний для виконання тієї ж задачі запропонованою моделлю (0.0189 с). Модель дерева рішень (DT) показала гарну точність, однак через просту структуру має нижчу здатність до узагальнення. Випадковий ліс (RF) забезпечив високу точність, але значно поступається у швидкодії.

Найнижчі результати продемонструвала глибока модель CNN-LSTM, яка виявилась не лише найповільнішою (0.2080 с – понад у 10 разів довше, ніж у

запропонованої моделі), але й мала найгіршу точність та F1-міру, особливо для класу BENIGN (F1 = 0.8527). Це свідчить про недонавчання на звичайному трафіку та перенавантаження ресурсів. Також варто враховувати, що моделі глибокого навчання потребують значно більших обсягів навчальних даних, чутливі до гіперпараметрів та потребують складної архітектури для забезпечення високої ефективності.

Таким чином, можна стверджувати, що запропонований у роботі підхід перевершує інші моделі за сукупністю метрик, забезпечуючи високу точність, стабільність класифікації та мінімальний час реагування, що особливо важливо для систем виявлення атак у реальному часі.

У таблиці додатку А представлено порівняльні результати багатокласової класифікації атакувального трафіку для різних моделей. Аналіз показує, що запропонований у цій роботі метод демонструє перевагу за всіма показниками точності, повноти та F1-міри для більшості класів атак, зокрема PortScan, BruteForce та Bot. Зокрема, для класу *Bot* модель досягла абсолютної точності (1.0000) за всіма метриками, що свідчить про надійність та узгодженість її роботи навіть при обмеженій кількості зразків.

Для DoS-атак точність виявлення є порівняною з іншими моделями, а відносно Web-атак результат запропонованої моделі дещо поступається моделі Random Forest (RF) лише за точністю (Precision), але при цьому перевершує її за часом виконання, витрачаючи менше 5% часу, необхідного для роботи RF. Це критично важливо для систем виявлення атак у режимі реального часу, де час реакції є не менш важливим, ніж точність.

Щодо моделі CNN-LSTM, то, попри прийнятний рівень точності для окремих класів атак, загальна ефективність та швидкість її роботи є найнижчими. Її час обробки (0.1340 с) у 10 разів перевищує час, необхідний запропонованій моделі (0.0128 с), що підтверджує її непридатність для практичного використання у сценаріях великих даних з високою швидкістю генерації трафіку.

На рисунку 3.6 представлено теплову карту, яка наочно демонструє точність класифікації запропонованою моделлю для кожного типу атак. Як видно, всі класи атак, включно з малочисельними (*Bot*, *WebAttack*), були

розпізнані з високою точністю, що ще раз підтверджує високу узагальнюючу здатність запропонованого підходу навіть у складних умовах розподілу класів.

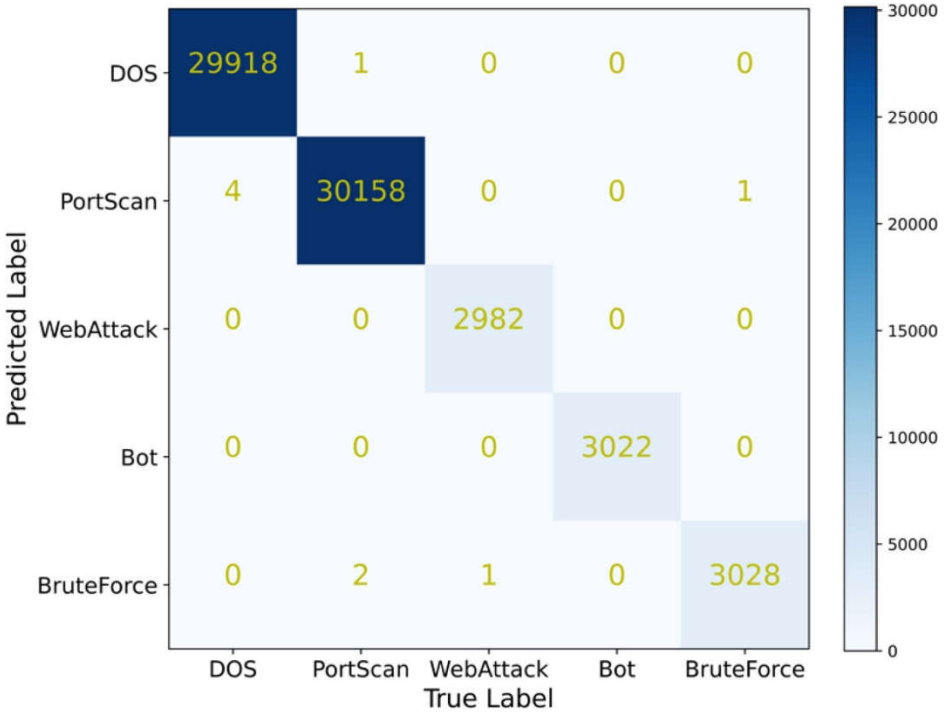

Рисунок 3.6 – Теплова карта ефективності виявлення атак моделлю

Таким чином, за результатами багатокласових експериментів, можна стверджувати, що запропонований метод є найбільш збалансованим за показниками точності, продуктивності та обчислювальної ефективності порівняно з іншими протестованими моделями.

ВИСНОВКИ

1. У сучасному цифровому середовищі мережеві атаки набувають дедалі складніших форм і активно використовують як технічні вразливості систем, так і соціальні маніпуляції. Серед найнебезпечніших загроз виокремлюють DoS/DDoS-атаки, фішинг, SQL-ін’єкції, сканування портів і атаки типу "людина посередині", кожна з яких вимагає специфічних підходів до виявлення. Це формує потребу у впровадженні адаптивних методів аналізу мережевого трафіку, здатних працювати у реальному часі в умовах великих обсягів даних і високої варіативності атак.

2. Аналіз існуючих підходів засвідчив ефективність використання методів машинного навчання для виявлення атак, однак прості моделі мають обмежену здатність до узагальнення та не завжди витримують обчислювальні навантаження. Глибокі нейронні мережі показують високі результати у видобуванні прихованих ознак, але потребують значних ресурсів і часу. У цьому контексті поєднання еволюційних обчислень із легкими моделями, такими як LightGBM, відкриває нові можливості для побудови масштабованих, швидкодійних та точних систем виявлення мережевих атак у середовищі великих даних.

3. Запропоновано метод виявлення мережевих атак, який є багаторівневим та включає послідовні етапи обробки даних, вибору релевантних ознак і поетапної класифікації. Такий підхід дозволяє не лише підвищити точність виявлення, а й зменшити час обробки трафіку за рахунок цілеспрямованої фільтрації та спеціалізації моделей. Розділення задачі на двокласову і багатокласову класифікацію забезпечує адаптивність і масштабованість методу в умовах високої складності та обсягів даних.

4. Попередня обробка даних у запропонованому методі реалізується через очищення, нормалізацію, балансування вибірки та перекодування ознак. Використання техніки RandomSample-SMOTE дозволило усунути дисбаланс між класами, що є критичним для ефективного навчання моделі. Ретельна підготовка

даних сприяє підвищенню загальної якості класифікації та запобігає упередженості моделі до переважаючих класів.

5. Алгоритм вибору ознак на основі моделі LightGBM забезпечує автоматизоване виявлення найбільш значущих параметрів для обох рівнів класифікації. Завдяки градієнтному аналізу і гістограмному представленню даних модель досягає високої продуктивності навіть при великій кількості вхідних ознак. У поєднанні з двошаровою структурою класифікації, це дозволяє швидко та точно розпізнавати як наявність атак, так і їх конкретні типи.

6. Результати експериментальних досліджень підтвердили ефективність запропонованої моделі в умовах великих даних. Застосування ієрархічного підходу до вибору ознак і дворівневої структури класифікації дозволило досягти високої точності як для двокласової, так і багатокласової класифікації. Крім того, оптимізація гіперпараметрів та попереднє балансування вибірки сприяли зниженню похибок класифікації та забезпечили стабільну роботу моделі на різних етапах навчання.

7. Порівняльний аналіз з іншими моделями машинного та глибокого навчання засвідчив перевагу запропонованого підходу за такими критеріями, як точність, F1-міра та час інференсу. Модель забезпечила найкращий баланс між якістю класифікації та обчислювальною ефективністю, що робить її придатною для практичного застосування в режимі реального часу. Особливо варто відзначити її здатність точно виявляти навіть малочисельні класи атак, що є критично важливим для сучасних систем кіберзахисту.