УДК 004.02

КЛАСИФІКАЦІЯ МЕРЕЖНОГО ТРАФІКУ МЕТОДАМИ МАШИННОГО НАВЧАННЯ

Л.С. Глоба, А.А. Астраханцев, С.О. Цуканов
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»

Abstract *– The growth of traffic sources and their diversity leads to increased traffic volumes. This makes existing traffic classification methods less effective. In addition, the expansion of the range of services provided leads to the emergence of new threats and vulnerabilities in the network. The task of detecting threats at an early stage is very important, as losses from threats have increased significantly worldwide in recent years, and early detection will help minimize possible risks. At the same time, implementing artificial intelligence software into all network elements, as part of the 5G/6G concept, allows part of the attack detection procedures to be transferred to the network edge, primarily to base stations. The use of intelligent traffic classification methods will help to increase the efficiency of information processing, as well as detect anomalous traffic blocks and block their sources. The paper is devoted to the urgent task of analyzing the efficiency (accuracy, speed) of traffic classification methods with subsequent detection of malicious traffic. According to the results, the best methods for accuracy and speed are Decision Tree (DT) and Random Forest (RF). The optimal sets of hyper-parameters have been determined for all the analyzed methods. The next most efficient are multilayer perceptron neural networks and methods based on rules and fuzzy sets, but both algorithms require much longer training time than all others. The scientific novelty of the work is due to the analysis of the possibilities of applying classification methods based on rules and fuzzy sets and a comprehensive analysis of the performance of the studied methods on a real dataset. These traffic classification and anomaly detection methods should be implemented at base stations to increase the security and resilience of mobile networks.*

Анотація *– Зростання кількості джерел трафіку і збільшення їх різноманітності призводить до зростання обсягів трафіка. Це робить існуючі методи класифікації трафіка менш ефективними. Крім того, розширення спектра послуг, що надаються призводить до появи нових загроз і вразливостей у мережі. Задача виявлення загроз на ранніх стадіях є дуже важливою, оскільки останніми роками по всьому світу суттєво зросли збитки від реалізації загроз і раннє виявлення дозволить мінімізувати можливі ризики. При цьому впровадження програмних засобів зі штучним інтелектом у всі елементи мережі, закладене в концепції 5G/6G, дозволяє частину процедур по виявленню атак перенести саме на межу мережі, в першу чергу на базові станції. Застосування інтелектуальних методів класифікації трафіка сприятиме підвищенню ефективності обробки інформації, а також дозволить виявити аномальні блоки трафіка і заблокувати їх джерела. Робота присвячена актуальній задачі аналізу ефективності (точності, швидкодії) методів класифікації трафіка. Згідно з отриманими результатами, найкращими за критеріями точності та швидкодії є методи дерева рішень та випадкового лісу. Для всіх досліджуваних методів визначено оптимальні набори гіперпараметрів. Наступними по ефективності є нейронні мережі на основі башатошарового персептрону та засновані на використанні правил та нечітких множин, але обидва алгоритми потребують набагато більшого часу тренування серед всіх інших. Наукова новизна роботи обумовлена аналізом можливостей застосування методів класифікації на основі правил та нечітких множин, а також комплексним аналізом швидкодії досліджуваних методів на реальному датасеті. Вказані методи класифікації трафіка і виявлення аномалій можуть бути реалізовані на базових станціях і дозволять підвищити захищеність і стійкість до атак мобільної мережі.*

Вступ

Об’єм даних у повсякденному житті людства постійно зростає: з’являються нові сховища інформації, ресурси на яких інформація розміщується та способи як її передачі, так і обробки. Прикладом можуть бути сучасні транспортні засоби, які з’єднуватимуться гетерогенними технологіями радіодоступу та зможуть обмінюватися великою кількістю інформації з навколишнім середовищем. Це потребує значного розширення масштабу мережі та реалізації обробки інформації у реальному часі. Традиційні транспортні спеціальні мережі (VANET) еволюціонують в Інтернет транспортних засобів (IoV). Крім того, розвиток різноманітних smart-виробництв, а

відповідно збільшення сенсорів та датчиків призводить до перевантаження та зниження якості обслуговування як в мережі радіодоступу, так і в ядрі мережі. Трафік даних мобільної мережі збільшився на 36 відсотків між 1 кварталом 2022 року та 1 кварталом 2023 року. Квартальне зростання трафіку даних мобільної мережі між 4 кварталом 2022 року та 1 кварталом 2023 року становило близько 7 відсотків. Довгострокове зростання трафіка зумовлене як зростанням кількості підписок на смартфонах, так і збільшенням середнього обсягу даних на одну підписку, головним чином через збільшення кількості переглядів відеоконтенту. Також у 2023 році кількість підписок на мобільний зв'язок 5G досягне 1,5 мільярда [1].

Водночас проблемою є не лише швидкість зростання обсягів трафіка, а й величезна кількість атак, які можуть бути реалізовані через мережу, наприклад атаки на відмову в обслуговуванні або спроби вторгнення. Мережа має мати механізм розпізнавання зловмисних пакетів, оскільки ефективно виконана атака в мережах 5G може призвести до небажаних подій та серйозних наслідків.

Допомогти у вирішенні цих проблем можуть інтелектуальні засоби обробки та класифікації трафіка, які відповідно до концепції мереж 5G/6G можуть бути розгорнуті як на граничних серверах, так і безпосередньо на базовій станції. Таким чином актуальною є задача, пов'язана з підвищенням ефективності обробки, аналізу трафіка, а також виявленням потенційних атак у мобільних мережах 5G/6G.

Одним із підходів до вирішення цієї задачі, особливо виявлення атак зловмисників, є застосування методів машинного навчання, які дозволяють досить точно визначати аномальний трафік. Але застосування методів машинного навчання характеризується значними часовими затримками, тому потребує визначення метрик класифікації та оптимальних параметрів нейронної мережі для пришвидшення процесу класифікації. В зв'язку з цим метою даної роботи є аналіз ефективності алгоритмів машинного навчання для вирішення задачі класифікації трафіка на предмет потенційних загроз у мобільних мережах 5G за критеріями точності, влучності, повноти, та швидкодії під час проведення обчислень. За підсумками аналізу мають бути сформовані рекомендації по застосуванню алгоритмів машинного навчання і визначені оптимальні параметри, які мають значний вплив на процес класифікації.

I. Сучасні підходи щодо класифікації трафіку в комунікаційній мережі

Задачу класифікації трафіку в комунікаційній мережі розглянуто в ряді робіт. Зокрема в роботі [2] пропонується підхід для розпізнавання втручань за допомогою згорткової нейронної мережі (CNN). Причина використання даного підходу – велика розмірність вхідних даних. В роботі експерименти проводилися на наборі даних CICIDS2017 [3]. Дані представлено у вигляді таблиць, які мають 79 властивостей. В результаті вдалося досягти 99,87% точності та f-міри – 99,87% для 100000 записів.

В роботі [4] описано підхід, який використовує алгоритм нечіткої класифікації (FCM) для кластеризації даних, потім для кожного кластеру відбувається тренування

окремої нейронної мережі. На етапі прогнозування кожна з цих мереж видає свій прогноз і далі відбувається агрегація всіх результатів. Експерименти проводились на наборі даних KDDCUP1999. Результати роботи показують, що даний підхід для атак класу DoS показав найкращі результати з поміж інших алгоритмів, а саме дерево рішень, штучна нейронна мережа, та наївний баєсів класифікатор. Такий самий результат було отримано для R2L та U2L атак. Проте для деяких класів атак даний підхід поступається іншим, наприклад, для класу атак PRB алгоритми дерево рішень, штучна нейронна мережа та наївний баєсів класифікатор мають більше значень метрик класифікації.

Робота [5] порівнює два підходи для попередньої обробки даних – вибір ознак та вилучення ознак. Підхід вибору ознак – полягає у відборі найкращих ознак даних і відкиданні найгірших на основі експертних оцінок, статистик, аналізу даних. У результаті розмірність вхідних даних зменшується, що спрощує подальший аналіз, проте супроводжується втратою інформації, більш того, існує вірогідність помилково відкинути інформативні ознаки. Це може бути критичним при подальшому прогнозуванні. Натомість підхід вилучення ознак полягає в тому, щоб знижувати розмірність вхідних даних за допомогою методу головних компонент (Principal Component Analysis, PCA) або з використанням автоенкодерів. Відмінність цього підходу полягає в тому, що виконується відображення даних у простір меншої розмірності, це дозволяє втрачати менше інформації ніж в попередньому підході, але саме перетворення вхідної інформації потребує обчислювальних ресурсів, особливо у випадку автоенкодера.

Робота [6] пропонує відбір XGBOOST моделей за допомогою генетичного алгоритму. Важливим тут є вибір правильної функції, оскільки XGBOOST моделі можуть перенавчатися. Результати 10-кратного тесту перехресної перевірки показали, що запропонована модель перевершує найсучасніші моделі та інші класифікатори ансамблевого навчання з високими показниками виявлення 98,2%, 92,9%, 98,9% і 99,5% для flooding, scheduling, grayhole, blackhole атак відповідно, та до 99,9% для звичайного трафіку.

Робота [7] описує підхід, основна ідея якого – це прогнозування за допомогою LSTM мережі. Це робиться для того, щоб виконувати прогнозування для послідовності пакетів, а не для одного конкретного пакету, як виконується в роботах [3-5] та [6]. Такий підхід дозволяє точніше виявляти довгострокові атаки, наприклад DDoS або інші атаки.

Виходячи з проведеного аналізу, для проведення досліджень обрано наступні алгоритми: KNN, Softmax regression, Decision tree, Random Forest, MLP. Також пропонується порівняння цих моделей з підходом на нечіткій логіці TSK.

II. Постановка задачі дослідження

Багатокласові алгоритми класифікації спрямовані на призначення мітки класу для кожного вхідного екземпляра.

В якості вхідних даних використаємо навчальний набір даних у формі (x_i, y_i), де $y_i \in \{1,...,\mathrm{K}\}$ – *i*-та мітка класу. Необхідно знайти модель навчання *H* таку, щоб $\mathrm{H}(x_i) = y_i$ для нових непередбачених екземплярів [8]. Для порівняння ефективності класифікації обраними алгоритмами машинного навчання для вирішення задач класифікації трафіка використаний датасет LUFlow [9]. LUFlow – це набір даних виявлення вторгнень на основі потоку. LUFlow містить телеметрію, що містить нові вектори атак через композицію honeypots в адресному просторі Ланкастерського університету.

Розглянемо більш детально алгоритми машинного навчання, використані в даній роботі.

1. KNN. Алгоритм k-найближчих сусідів – це контрольований алгоритм машинного навчання. Основна ідея алгоритму k-найближчих сусідів полягає в тому, щоб знайти k найближчих точок даних або сусідів до заданої точки даних, а потім передбачити мітку або значення даної точки даних на основі міток або значень її k найближчих сусідів. Параметр k може бути будь-яким додатним цілим числом, але на практиці k часто невелике, наприклад 3 або 5. Параметр k у *k*-найближчих сусідах означає кількість елементів, які використовує алгоритм, щоб зробити прогноз.

2. Softmax regression (SR). Алгоритм "Софтмакс регресія" – це алгоритм класифікації, що являє собою узагальнення логістичної регресії для випадків коли кількість класів у задачі класифікації більше двох.

У налаштуванні софтмакс регресії можлива багатокласова класифікація, тому мітка $y\left(y \in \{1,...,\mathrm{K}\}\right)$ може приймати K різних значень. Маючи тестовий вхід x, потрібно, щоб модель оцінила ймовірність того, що мітка класу прийме кожне з K різних можливих значень. Таким чином, модель виведе K-вимірний вектор, сума елементів якого дорівнює 1, значення кожного елемента вектора показує ступінь приналежності конкретного спостереження до класу.

3. Decision tree (DT). Алгоритм "Дерево рішень" є частиною сімейства алгоритмів керованого навчання і відрізняється тим, що його можна використовувати як для задач класифікації, так і для задач регресії. Метою дерева рішень є створення моделі, яка здатна прогнозувати клас або значення цільової змінної, вивчаючи прості правила прийняття рішень з навчальних даних.

Процес прогнозування мітки класу для конкретного запису розпочинається з кореневого вузла дерева. Алгоритм порівнює значення кореневого атрибута з атрибутом запису і рухається по гілці, відповідній цьому значенню, до досягнення листового вузла.

Переваги дерев рішень включають їхню простоту для розуміння та інтерпретації, малу необхідність підготовки даних, вартість використання, що логарифмічно залежить

від кількості точок даних, та можливість вирішення проблем з декількома виходами. Вони також можуть працювати як з числовими, так і з категоріальними даними.

Недоліки дерев рішень включають можливість переобладнання, нестабільність відносно варіацій в даних. Додатково, дерева рішень можуть слабо прогнозувати, якщо деякі класи домінують у наборі даних, тому рекомендується збалансувати дані перед їхнім використанням.

4. Random Forest (RF). Алгоритм "Випадковий ліс" базується на використанні дерев рішень і може служити для вирішення як завдань класифікації, так і завдань регресії в машинному навчанні. Дерева рішень у машинному навчанні використовуються для створення моделей прогнозування і відомі як дерева прийняття рішень. У цих деревах передбачення подається через послідовні "якщо... тоді..." гілки рішень, що визначаються поділом на різні гілки. Кожна гілка визначається як функція в машинному навчанні, приймаючи рішення до досягнення кінцевого листа, що представляє прогнозований клас або значення.

В алгоритмі випадкового лісу прогнози формуються шляхом комбінування результатів з численних окремих дерев рішень. Для налаштування процесу навчання алгоритму використовують гіперпараметри, такі як кількість дерев, які об'єднуються, максимальна глибина дерев, максимальна кількість ознак при кожному розбитті, а також, чи використовується паралельне або послідовне навчання класифікаторів.

Основні переваги алгоритму випадкового лісу включають його відносну швидкість і потужність у завданнях класифікації та регресії. Цей алгоритм може ефективно обробляти багато задач, включаючи невеликі набори даних, оскільки розрахунки можуть бути паралелізовані, а вихідні дані надають імовірності прогнозування.

Серед недоліків випадкового лісу – схильність до перенавчання, також алгоритм може погано прогнозувати менш представлені класи в незбалансованих наборах даних.

5. MLP. Багатошаровий перцептрон (MLP) є найвідомішим і найчастіше використовуваним типом нейронної мережі. У більшості випадків сигнали передаються в мережі в одному напрямку: від входу до виходу. Немає циклу, вихід кожного нейрона не впливає на сам нейрон. Ця архітектура називається прямою передачею [10].

Процес тренування багатошарового перцептрона складається з двох фаз. У прямій фазі синаптичні ваги мережі фіксуються, і вхідний сигнал поширюється через мережу, шар за шаром, доки не досягне вихідного сигналу. Таким чином, на цій фазі зміни обмежуються потенціалами активації та виходами нейронів у мережі. У зворотній фазі сигнал про помилку створюється шляхом порівняння виходу мережі з бажаною відповіддю. Результуючий сигнал помилки поширюється по мережі знову шар за шаром, але цього разу поширення виконується у зворотному напрямку. На цій другій фазі вносяться послідовні коригування синаптичних ваг мережі. Розрахунок коригувань для вихідного шару простий, але набагато складніший для прихованих шарів [11].

6. TSK. Нечітка система Такагі-Сугено-Канга (TSK) може бути представлена як нейронна мережа з п'ятьма шарами, вона також відома як нечітка нейронна мережа Такагі-Сугено-Канга. Параметри нечіткої нейронної мережі Такагі-Сугено-Канга налаштовуються за допомогою еволюційних алгоритмів, або градієнтного спуску, або гібридних

алгоритмів. Еволюційні алгоритми вимагають підтримки великої популяції, можуть сходитися до різних рішень і, відповідно, потребують великих обчислювальних витрат. Градієнтний спуск вимагає обчислення градієнта для всього набору даних з метою повторного оновлення параметрів. Це може відбуватися досить повільно, якщо ми маємо велику кількість параметрів і велику кількість даних. Традиційні підходи до вирішення проблеми занадто великої кількості параметрів включають зменшення кількості функцій і коригування кількості правил [12]. Нечіткі моделі, попри те, що не є найбільш популярними, показують непогані результати. Для набору даних [13], який призначений для бінарної класифікації захворювання на COVID, мережі вдалося досягти 98,56% точності, за допомогою агрегування 5 таких мереж, які були натренеровані з різними параметрами.

III. Підходи до оцінки ефективності алгоритмів

Для оцінки ефективності алгоритмів класифікації необхідно обрати метрики класифікації для мультикласової класифікації, які будуть відображати якість результатів, а також виміряти час тренування та прогнозування. Для оцінки якості побудованих моделей у роботі використовуються наступні метрики (табл. 1).

Таблиця 1. Метрики класифікації для оцінки якості побудованих моделей

Метрика	Формула	Метрика	Формула
$Accuracy$	$\frac{\sum_{i=1}^{l}\frac{tp_i+tn_i}{tp_i+fn_i+fp_i+tn_i}}{l}$	$precision_{\text{M}}$	$\frac{\sum_{i=1}^{l}\frac{tp_i+tn_i}{tp_i+fp_i}}{l}$
$precision_{\mu}$	$\frac{\sum_{i=1}^{l}tp_i}{\sum_{i=1}^{l}(tp_i+fp_i)}$	$recall_{\text{M}}$	$\frac{\sum_{i=1}^{l}\frac{tp_i}{tp_i+fn_i}}{l}$
$recall_{\mu}$	$\frac{\sum_{i=1}^{l}tp_i}{\sum_{i=1}^{l}(tp_i+fn_i)}$	$Fscore_{\text{M}}$	$\frac{(\beta^2+1)precision_{\text{M}}recall_{\text{M}}}{\beta^2 precision_{\text{M}}recall_{\text{M}}}$

У табл. 1 наведено показники багатокласової класифікації. Для окремого класу C_i визначається tp_i; fn_i; tn_i; fp_i, де tp_i – кількість екземплярів, які вірно були визначені як ті, що відносяться до класу i; fn_i – кількість екземплярів, які помилково були визначені як ті, що не відносяться до класу i; tn_i – кількість екземплярів, які вірно були визначені як ті, що не відносяться до класу i; fp_i – кількість екземплярів, які помилково були визначені як ті, що не відносяться до класу i.

Якість загальної класифікації зазвичай оцінюють двома способами: мірою є середнє значення показників, розрахованих для C_i (макроусереднення, показане з індексом M), або за допомогою мікроусереднення [14].

IV. Підготовка експерименту та результати моделювання

Для проведення експерименту було взято частину датасету LUFlow. Усі маніпуляції з даними виконувались за допомогою мови програмування Python з використанням бібліотеки Pandas. Для побудови моделей дерева рішень, випадкового лісу, k-найближчих сусідів, софтмакс регресії, багатошарового перцептрона було використано бібліотеку scikit-learn, для моделі Такагі-Сугено-Канга було використано рішення з [15].

Перед проведенням експерименту, необхідно підготувати дані для подальшої роботи з ними. Датасет має 1068376 рядків та 16 стовпців. Перший етап у підготовці даних – видалення дублікатів. Це необхідно, оскільки наявність великої кількості дублікатів може уповільнити подальшу обробку даних, навчання моделей, а головне через наявність дублікатів неможливо точно порахувати статистики. В результаті було виявлено і видалено 4284 дублікатів. Наступний крок – перетворення значень цільової змінної “label” у числові: значення “outlier” було закодовано 0, “malicious” – 1, “benign” – 2. Це потрібно зробити, оскільки алгоритми машинного навчання працюють з числовими значеннями.

Далі проводиться нормалізація даних. Мета нормалізації полягає в тому, щоб трансформувати об’єкти в подібний масштаб. Це покращує продуктивність і стабільність тренування моделі. Нормалізацію було виконано за допомогою Z-оцінки. Результатом нормалізації Z-оцінки є те, що ознаки будуть перемасштабовані таким чином, щоб вони мали властивості стандартного нормального розподілу з $\mu = 0$ і $\sigma = 1$, де μ – середнє, а σ – стандартне відхилення від середнього. Стандартні оцінки зразків розраховуються за наступним рівнянням [16]:

$$Z = \frac{x - \mu}{\sigma}. \qquad (1)$$

Після цього дані були випадковим чином розділені на тренувальну та тестову вибірку. Для тестової вибірки було виділено 20% даних, інша частина даних використовувалася для тренування моделей.

Наступним кроком проведено тренування моделей на тренувальній вибірці та виміряно метрики класифікації. Результати наведено в табл. 2.

Таблиця 2. Ефективність класифікації для побудованих моделей

Назва методу	*accuracy*	$precision_{M}$	$precision_{\mu}$	$recall_{M}$	$recall_{\mu}$	$Fscore_{M}$	$Fscore_{\mu}$
DT	0,98	0,97	0,98	0,94	0,98	0,95	0,98
RF	0,98	0,97	0,98	0,94	0,98	0,95	0,98
KNN	0,97	0,93	0,97	0,93	0,97	0,93	0,97
SR	0,86	0,83	0,86	0,76	0,86	0,78	0,85
MLP	0,95	0,91	0,95	0,85	0,95	0,88	0,95
TSK	0,96	0,92	0,96	0,88	0,96	0,90	0,96

З табл. 2 випливає, що найкраще для даної задачі себе показали алгоритми дерева рішень і випадкового лісу. Значення для усіх метрик класифікації для цих методів більше ніж у інших.

Також було проведено вимір часу у секундах, за який модель тренується, прогнозує дані для тестової вибірки, та час, за який модель прогнозує одне спостереження з тестової вибірки. Результати експерименту наведені в таблиці 3. При цьому в табл. 3 як T1 позначений час тренування, T2 відповідає часу прогнозування для тестової вибірки, а T3 – час прогнозу для тестової вибірки для одного спостереження.

Таблиця 3. Аналіз швидкодії для побудованих моделей

Назва методу	T1, с	T2, с	T3, с
DT	3,28	0,013	0,00029
RF	26,99	0,012	0,0003
KNN	3,89	167,6	0,00786
SR	30,80	0,184	0,01888
MLP	593,4	0,507	0,00046
TSK	877,3	0,007	0,00116

Результати порівняльного аналізу показали, що обидва алгоритми дерева рішень та випадкового лісу показали найкращі результати не тільки за метриками (точність 0,98), а й по часу тренування та прогнозування (~0,012 с). Далі за значеннями метрик класифікації йде алгоритм k-найближчих сусідів. Серед інших переваг – швидке тренування, однак алгоритм погано справляється з одночасним прогнозуванням великої кількості даних.

Наступними по ефективності є багатошаровий перцептрон та нечітка нейронна мережа Такагі-Сугено-Канга. Обидва алгоритми потребують набагато більшого часу тренування серед всіх інших. Нечітка нейронна мережа Такагі-Сугено-Канга показала трохи кращі результати за метриками класифікації, ніж багатошаровий перцептрон, але час його тренування суттєво більший. Він краще прогнозує велику кількість даних, але поступається при прогнозі одного спостереження. Загалом можна зробити висновок, що дані методи мають потенціал для вирішення подібних задач.

Гірше всіх показав себе алгоритм софтмакс регресії за всіма метриками класифікації. Це пов'язано з тим, що лінійна модель не здатна вирішити конкретно цю задачу з прийнятною точністю. Разом з тим дана модель має швидкий час тренування та задовільний час прогнозу.

Таким чином, експериментальні дані показують, що найбільш прийнятними є моделі дерева рішень та випадкового лісу. Для всіх досліджуваних методів були визначені оптимальні значення гіперпараметрів (табл. 4), які дозволяють отримати характеристики, наведені в табл. 2.

Таблиця 4. Оптимальні значення гіперпараметрів для побудованих моделей

Назва методу	Оптимальні значення гіперпараметрів моделі
DT	*min_samples_split*: **3**, *min_samples_leaf*: **4**, *max_depth*: **10**, *criterion*: **entropy**
RF	*n_estimators*: **20**, *min_samples_split*: **5**, *min_samples_leaf*: **5**, *max_features*: **1**, *max_depth*: **20**, *criterion*: **gini**
KNN	*n_neighbors*: **5**
SR	*penalty*: **none**, *max_iter*: **100**
MLP	*learning_rate*: **invscaling**, *hidden_layer_sizes*: (**40, 20**), *alpha*: **0,006**, *activation*: **relu**
TSK	*n_rule*: **50**, *order*: **1**, *lr*: **0,01**

Згідно з табл. 4 для найкращих методів рекомендовано застосовувати наступні параметри: для випадкового лісу кількість дерев 20 та максимальну глибину дерева 20, а для дерева рішень – максимальну глибину дерева 10.

В результаті проведених досліджень можна констатувати, що застосування інтелектуальних методів класифікації трафіка випадкового лісу та дерева рішень підвищує ефективність класифікації, а також дозволяє виявити аномальні блоки трафіка і в подальшому заблокувати їх джерела.

Висновки

Поява нових сервісів і нових типів та джерел трафіку ускладнюють роботу існуючих методів класифікації й обробки трафіка. Додатково зловмисники вносять свій вплив, намагаючись отримати доступ до приватних даних або унеможливити роботу мережі взагалі.

В даній роботі запропоновано рішення задачі підвищення ефективності процесів обробки, аналізу та класифікації трафіка в мобільній мережі за критеріями точності та швидкодії під час проведення обчислень, виявлення потенційних загроз у мобільних мережах 5G. Для проведення експериментальних досліджень створено програмний додаток на мові Python.

Найкращі результати за вказаними критеріями забезпечують методи дерева рішень та випадкового лісу. Для всіх досліджуваних методів визначено налаштування

гіперпараметрів, які забезпечують найкращі значення за наведеними вище критеріями. Результати, отримані для цих методів, перевершують найближчий до них метод k-найближчих сусідів за показником точності (accuracy) на ~1,03%.

Досліджено можливість використання методів на нечітких правилах для класифікації трафіка. Запропонований підхід може бути застосований у вузлі мобільної мережі. Результати показали, що застосування таких методів дає високу точність класифікації (0,96), але вимагає значно більшого часу навчання моделі (877 с, в 32 рази довше ніж випадковий ліс), що може бути неприйнятним для on-fly підтримки під час обробки трафіку та постійного самонавчання в реальній мобільній мережі.

Запропоновані рішення можуть бути першим етапом комплексної обробки пакетів в мобільній мережі. Разом із кластеризацією, слайсингом та розподіленою обробкою даних запропонований підхід сприятиме підвищенню ефективності системи мобільного зв'язку загалом.

Перелік скорочень

CNN – Convolutional Neural Network
DDoS – Distributed denial-of-service attack
DoS – Denial-of-service attack
DT – Decision Tree
FCM – Fuzzy C-Means
IoV – Internet of Vehicles
KDDCUP – Knowledge Discovery and Data Mining Tools Competition
KNN – k-Nearest Neighbors
MLP – Multilayer perceptron
PCA – Principal Component Analysis
PRB – Probing / surveillance attack
R2L – Remote-to-Local attack
RF – Random Forest
SR – Softmax regression
TSK – Takagi-Sugeno-Kang model
U2L – User-to-Local attack
VANET – Vehicular Ad Hoc Network
XGBOOST – Extreme Gradient Boost

Список літератури

1. Ericsson Mobility Report (2023), **URL**: https://www.ericsson.com/en/reports-and-papers/mobility-report/reports/june-2023.

2. *Mohammadpour, L., Ling, T.C., Liew, C.S., Aryanfar, A.* (2020), "A Mean Convolutional Layer for Intrusion Detection System", Security and Communication Networks, No. 1, P. 1–16. **DOI**: https://doi.org/10.1155/2020/8891185.

3. CICIDS2017 Intrusion Detection Evaluation Dataset, **URL**: https://www.kaggle.com/datasets/cicdataset/cicids2017.

4. *Wang, G., Hao, J,. Ma, J., Huang, L.* (2010), "A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering", Expert Systems with Applications, No. 37(9), P. 6225–6232. **DOI**: https://doi.org/10.1016/j.eswa.2010.02.102.

5. *Ngo, V. D., Vuong, T. C., Van Luong, T.* (2023), "Machine learning-based intrusion detection: feature selection versus feature extraction", Cluster Computing, No. 27, P. 2365–2379, **DOI**: https://doi.org/10.1007/s10586-023-04089-5.

7. *Sun, P., Liu, P., Li, Q., Liu, C., Lu, X., Hao, R., Chen, J.* (2020), "DL-IDS: Extracting Features Using CNN-LSTM Hybrid Network for Intrusion Detection System", Security and Communication Networks, No. 1, P. 1–11. **DOI**: https://doi.org/10.1155/2020/8890306.

6. *Alqahtani, M., Gumaei, A., Mathkour, H., Maher Ben Ismail, M.* (2019), "A Genetic-Based Extreme Gradient Boosting Model for Detecting Intrusions in Wireless Sensor Networks", Sensors, No. 19(20), 4383. **DOI**: https://doi.org/10.3390/s19204383.

8. *Mehra, N., Gupta, S.* (2013), "Survey on multiclass classification methods", International Journal of Computer Science and Information Technologies, No. 4, P. 572–576.

9. LUFlow Network Intrusion Detection Data Set, **URL**: https://www.kaggle.com/datasets/mryanm/luflow-network-intrusion-detection-data-set.

10. *Popescu, M., Balas, V. E., Perescu-Popescu, L., Mastorakis, N. E.* (2009), "Multilayer perceptron and neural networks", WSEAS Transactions on Circuits and Systems, No. 8(7), P. 579–588.

11. *Haykin, S.* (2009), Neural Networks and Learning Machines, 3rd edition, Pearson, 938 p.

12. *Shapoval, N.* (2022), "TSK Fuzzy Neural Network Use for COVID-19 Classification", Electronics and Control Systems, No. 1(71), P. 50–54. **DOI**: https://doi.org/10.18372/1990-5548.71.16825.

13. Symptoms and COVID Presence (May 2020 data), **URL**: https://www.kaggle.com/datasets/hemanthhari/symptoms-and-covid-presence.

14. *Sokolova, M., Lapalme, G.* (2009), "A systematic analysis of performance measures for classification tasks", Information Processing and Management, No. 45, P. 427–437. **DOI**: https://doi.org/10.1016/j.ipm.2009.03.002.

15. *Cui, Y.*, PyTSK's documentation, **URL**: https://pytskdocs.readthedocs.io/en/latest/.

16. *Patro, S., Gopal, K., Sahu, K. K.* (2015), "Normalization: A Preprocessing Stage". *IARJSET*, P. 20–22. **DOI**: https://doi.org/10.17148/iarjset.2015.2305.