

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки**

До захисту допущено
Завідувач кафедри

Дмитро ЛАНДЕ

«12» червня 2024 р.

Дипломна робота

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Системи, технології та математичні
методи кібербезпеки»
спеціальності 125 «Кібербезпека»**

на тему: Виявлення мережевих атак методами пошуку аномалій на основі машинного навчання

Виконала: здобувачка вищої освіти **IV** курсу, групи ФБ-04

Подвисоцька Ольга Едуардівна

Керівник: к. т. н., доцент кафедри інформаційної безпеки

Коломицев Михайло Володимирович

Рецензент: асистент кафедри математичного моделювання та аналізу даних

Крохальов Іван Данилович

Засвідчую, що в цій дипломній роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Здобувачка вищої освіти

Київ — 2024 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки**

Рівень вищої освіти — перший (бакалаврський)
Спеціальність — 125 «Кібербезпека»
Освітньо-професійна програма «Системи, технології та математичні методи кібербезпеки»

ЗАТВЕРДЖУЮ
Завідувач кафедри

Дмитро ЛАНДЕ

«12» червня 2024 р.

**ЗАВДАННЯ
на дипломну роботу здобувачці вищої освіти**

Подвисоцькій Ользі Едуардівні

1. Тема роботи: Виявлення мережевих атак методами пошуку аномалій на основі машинного навчання,
керівник роботи к. т. н., доцент кафедри інформаційної безпеки Коломицев Михайло Володимирович,
затверджені наказом по університету від «31» травня 2024 р. №2251-с.
2. Термін подання здобувачкою вищої освіти роботи: 12 червня 2024 р.
3. Вихідні дані до роботи: наявні методи виявлення аномалій методами машинного навчання, дані про мережевий трафік і журнали системних подій CSE-CIC-IDS2018.
4. Зміст роботи: класифікація аномалій, огляд існуючих методів аналізу мережевого трафіку за допомогою машинного навчання, розробка та тестування комбінованого методу виявлення аномалій.
5. Перелік ілюстративного матеріалу: презентація.
6. Дата видачі завдання: 12.01.2024.

Календарний план

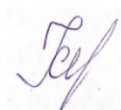
№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів дипломної роботи	Примітка
1	Визначення тематики й отримання завдання	12.01.2024	Виконано
2	Постановка мети та задач дипломної роботи	13.01.2024 – 30.01.2024	Виконано
3	Ознайомлення з літературними джерелами й аналіз наявних методів	31.01.2024 – 20.02.2024	Виконано
4	Формулювання й узгодження теми дипломної роботи	27.02.2024	Виконано
5	Підбір алгоритмів і пошук тестових даних	28.02.2024 – 19.03.2024	Виконано
6	Розробка комбінованого методу для виявлення аномалій мережевого трафіку	20.03.2024 – 14.04.2024	Виконано
7	Проходження переддипломної практики	15.04.2024 – 17.05.2024	Виконано
8	Тестування запропонованого методу	18.05.2024 – 02.06.2024	Виконано
9	Аналіз отриманих результатів	03.06.2024 – 11.06.2024	Виконано
10	Передзахист дипломної роботи	12.06.2024	Виконано
11	Захист дипломної роботи	19.06.2024	Виконано

Здобувачка вищої освіти



Ольга ПОДВИСОЦЬКА

Керівник роботи



Михайло КОЛОМИЦЕВ

РЕФЕРАТ

Обсяг дипломної роботи: 46 сторінок, 27 ілюстрацій, 1 таблиця, 1 додаток і 14 літературних джерел.

Мета дослідження: удосконалення існуючих способів виявлення мережових атак методом пошуку аномалій на основі машинного навчання.

Методи дослідження: аналіз теоретичних джерел і поширених алгоритмів машинного навчання для виявлення аномалій мережевого трафіку, вибір оптимальних методів і параметрів кластеризації та класифікації, пошук і попередня обробка даних про мережеві атаки, тестування та оцінка роботи комбінованого методу.

Результати роботи були представлені на XXII Всеукраїнській науково-практичній конференції студентів, аспірантів та молодих вчених «Теоретичні та прикладні проблеми фізики, математики та інформатики (13–17 травня 2024 р., м. Київ, Україна).

Ключові слова: машинне навчання (ML), навчання без учителя, навчання з учителем, кластеризація, класифікація, мережевий трафік, виявлення аномалій, аномалії мережевого трафіку.

ABSTRACT

The volume of the thesis is 46 pages, 27 illustrations, 1 table, 1 appendix and 14 literature sources.

Purpose of the research: improvement of existing methods of network attack detection using anomaly detection based on machine learning.

Methods used and results obtained: analysis of theoretical sources and common machine learning algorithms for detecting network traffic anomalies, selection of optimal methods and parameters of clustering and classification, search and preprocessing of data about network attacks, testing and evaluation of the combined method.

The results were presented at the XXII All-Ukrainian Scientific and Practical Conference of Students, Postgraduates and Young Scientists “Theoretical and Applied Problems of Physics, Mathematics and Informatics” (13–17 May 2024, Kyiv, Ukraine).

Keywords: machine learning (ML), unsupervised learning, supervised learning, clustering, classification, network traffic, anomaly detection, network traffic anomalies.

ЗМІСТ

ВСТУП.....	7
1 АНОМАЛІЇ МЕРЕЖЕВОГО ТРАФІКУ	9
1.1 Поняття аномалій мережевого трафіку та їх види	9
1.2 Види мережових атак, на які можуть вказувати аномалії	11
Висновки до розділу 1.....	13
2 ВИКОРИСТАННЯ МАШИННОГО НАВЧАННЯ ДЛЯ ВИЯВЛЕННЯ АНОМАЛІЙ МЕРЕЖЕВОГО ТРАФІКУ	14
2.1 Типи задач машинного навчання.....	14
2.2 Використання класифікації для аналізу мережевого трафіку.....	15
2.3 Пошук аномалій методами кластеризації	17
Висновки до розділу 2.....	21
3 КОМБІНОВАНИЙ МЕТОД ПОШУКУ АНОМАЛІЙ	22
3.1 Сутність комбінованого методу	22
3.2 Підбір алгоритмів і тестових даних.....	23
3.3 Аналіз результатів роботи методу.....	24
Висновки до розділу 3.....	32
ВИСНОВКИ	34
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	35
ДОДАТОК А	37

ВСТУП

На поточний момент існує чимало технологій для детектування мережевих атак, як-от системи виявлення та запобігання вторгненням (IDS, IPS), аналізатори поведінки мережі (NBA), системи управління подіями інформаційної безпеки (SIEM) тощо. Можливості таких інструментів покривають значну частину потреб їхніх користувачів. Проте зі зростанням частоти впровадження штучного інтелекту в процеси управління безпекою з'являються нові перспективи для розвитку підходів, пов'язаних з методами машинного навчання.

Актуальність роботи. Однією з можливостей застосування машинного навчання є виявлення аномалій трафіку з метою детектування мережевих атак. Серед наявних підходів виділяють навчання з учителем і без учителя. Цим способам аналізу властиві розмежування, зумовлені здебільшого наявністю чи відсутністю попередньо розміченої вибірки даних. Проте обидва підходи мають свої переваги, які має сенс об'єднати, створивши певний комбінований метод виявлення аномалій.

Мета дослідження. Удосконалення існуючих способів виявлення мережевих атак методом пошуку аномалій на основі машинного навчання.

Завдання дослідження. Проаналізувати існуючі методи виявлення аномалій; запропонувати покращений підхід до пошуку аномалій; протестувати отриманий метод на реальних даних мережевого трафіку.

Об'єкт дослідження. Дані про мережевий трафік і мережеві атаки.

Предмет дослідження. Аномалії мережевого трафіку та способи їх виявлення.

Методи дослідження. Аналіз теоретичних джерел і поширених алгоритмів машинного навчання для виявлення аномалій мережевого трафіку, вибір оптимальних методів і параметрів кластеризації та класифікації, пошук і попередня обробка даних про мережеві атаки, тестування й оцінка роботи комбінованого методу.

Наукова новизна одержаних результатів. Новий метод дає можливість застосовувати навчання з учителем для пошуку аномалій на нерозмічених даних мережевого трафіку.

Практичне значення одержаних результатів. Комбінований підхід зменшує об'єми використовуваних обчислювальних ресурсів для процесу виявлення мережових атак і поєднує переваги кластеризації та класифікації.

Апробація результатів роботи. Результати роботи були представлені на XXII Всеукраїнській науково-практичній конференції студентів, аспірантів та молодих вчених «Теоретичні і прикладні проблеми фізики, математики та інформатики (13–17 травня 2024 р., м. Київ, Україна).

Публікації. За результатами XXII Всеукраїнської науково-практичної конференції студентів доповідь було надруковано у збірник матеріалів.

1 АНОМАЛІЇ МЕРЕЖЕВОГО ТРАФІКУ

1.1 Поняття аномалій мережевого трафіку та їх види

Поняття аномалії в будь-якому контексті є схожими за своєю суттю та зводяться до наступного визначення. Аномалія — це будь-яке відхилення від нормальної або очікуваної поведінки. Розрізняють наступні їх типи:

- Точкові: найбільш прості та розповсюджені. Виникають, коли окремі події чи дані відрізняються від інших, нормальних, подій або даних. Часто пов'язані з мережевими атаками.

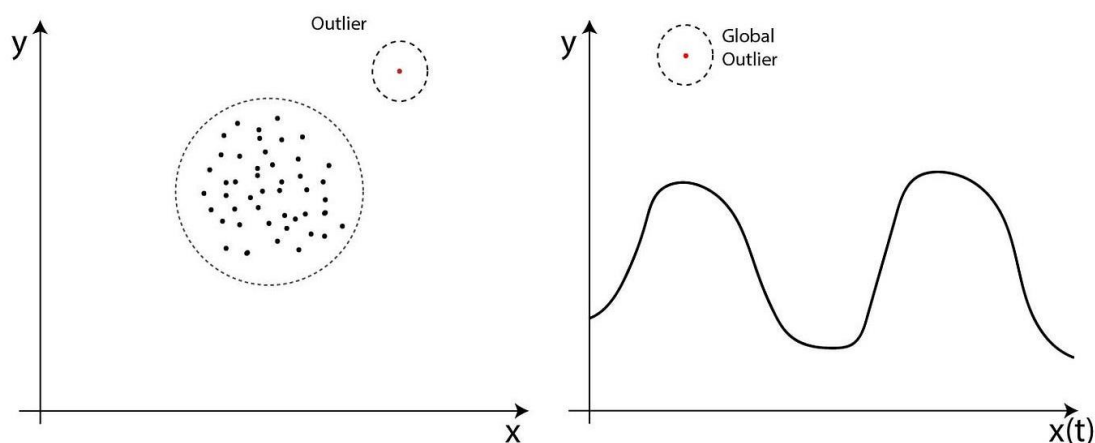


Рисунок 1.1 — Візуальне представлення точкових аномалій

- Контекстні: класифікація подій як нормальних чи аномальних може залежати від контексту, у якому їх розглядають.

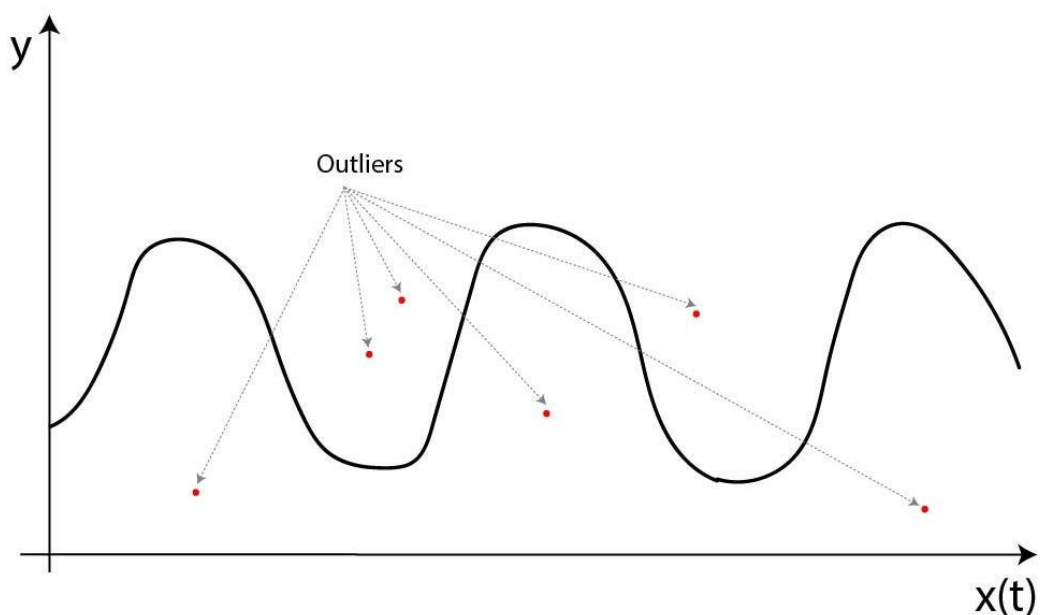


Рисунок 1.2 — Візуальне представлення контекстних аномалій

- Групові: з'являються за умови, що певна сукупність даних відрізняється від усіх інших, навіть якщо окремі сутності не є незвичайними. Можуть бути ознакам постійних загроз підвищеної складності (Advanced Persistent Threat, APT).

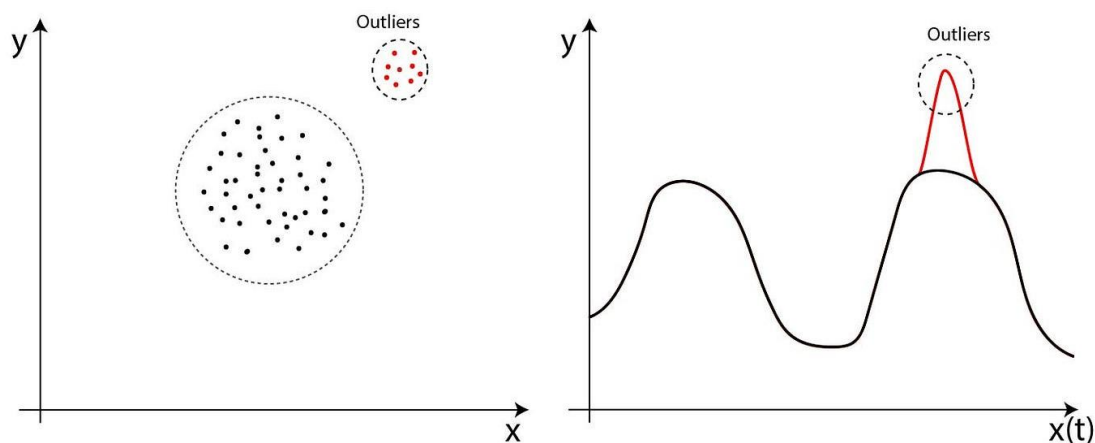


Рисунок 1.3 — Візуальне представлення групових аномалій

Окрім цього, виділяють аномалії за ознаками (Feature-Based) і часовими рядами (Time Series). Їх виявляють із застосуванням аналізу окремих характеристик даних або часових шаблонів трафіку.

Аномалією ж мережевого трафіку вважається незвичайна мережева активність, яка відрізняється від стандартних патернів. Вона може проявлятися в різних формах: різкі сплески об'ємів даних, нехарактерне використання мережевих

протоколів або портів, неочікувані зміни в часовій активності тощо. Такі відхилення можуть вказувати на збої в роботі обладнання, проблеми в конфігурації пристроїв, збільшення навантаження на мережу, зміни в поведінці користувачів, оновлення програмного забезпечення або спробу здійснення атаки.

Надалі в контексті цієї роботи під аномальним трафіком будемо мати на увазі той трафік, що не є нормальним, тобто безпечним.

1.2 Види мережевих атак, на які можуть вказувати аномалії

За одним із визначень мережевою атакою є дії, вжиті за допомогою використання комп'ютерних мереж, для порушення, відмови, погіршення якості чи знищення інформації, що знаходиться в комп'ютерах і комп'ютерних мережах, або самих комп'ютерів і мереж [4].

Наявність аномалій може вказувати зокрема на такі види атак:

- Distributed Denial of Service, DDoS — вичерпання ресурсів і відмова в обслуговуванні внаслідок надсилання великої кількості запитів до цільової системи.

Ознаки: суттєві сплески трафіку та багаторазові запити з різних IP-адрес.

- Сканування портів (Port Scanning) — пошук відкритих або вразливих портів на цільовому пристрої.

Ознаки: велика кількість спроб під'єднання до різних портів за короткий проміжок часу.

- Атака типу «людина посередині» (Man-in-the-Middle, MitM) — перехоплення зловмисником даних, що передаються між двома сторонами, без відома цих сторін.

Ознаки: затримки під час передавання даних, нехарактерні зміни у вмісті пакетів.

- Розповсюдження шкідливого програмного забезпечення — поширення ШПЗ мережею за допомогою вразливих пристроїв або реалізації фішингових атак.

Ознаки: наявність аномальних з'єднань (особливо з відомими шкідливими доменами), використання незвичних протоколів і портів.

- Фішинг (Phishing) — обманні дії щодо користувачів з метою отримання конфіденційної інформації.

Ознаки: завелика кількість вхідних листів, нехарактерні часові патерни в активності електронної пошти, поява нових невідомих доменів.

- Переповнення буфера (Buffer Overflow) — виконання довільного коду внаслідок надсилання зловмисником більшої кількості даних, аніж буфер здатен обробити.

Ознаки: великі розміри пакетів, що передаються мережею, часті перезапиту чи збої у з'єднаннях, сплески трафіку на певних портах.

- Атака за словником — злам облікових записів перебором паролів зі заздалегідь підготовленого списку (словника).

Ознаки: часті спроби входу та невдалі спроби автентифікації (зокрема для одного облікового запису).

- Активність ботнетів — множина заражених пристроїв, що керуються через командно-контрольні сервери (C2). Надалі ці сервери координують дії ботнету для реалізації інших атак.

Ознаки: неочікувані з'єднання в неробочі години, синхронна поведінка пристроїв, як-от надсилання однакових запитів і даних.

- Атака нульового дня (Zero-Day Attack) — використання невідомих вразливостей у програмному забезпеченні, для яких ще не були випущені відповідні патчі.

Не має чітко визначених характерних ознак, тому що вони можуть відрізнятися залежно від уразливості, що експлуатується.

Висновки до розділу 1

Вивчення аномалій мережевого трафіку є критично важливим для забезпечення безпеки інформаційних систем. Наявність аномалій може вказувати на різноманітні загрози, як-от мережеві атаки, технічні збої або інші небезпечні ситуації, які потребують негайного розгляду. Вони є важливими індикаторами, які допомагають виявляти та запобігати кібератакам, знижувати ризики порушення безпеки та підтримувати стабільну роботу мережевих інфраструктур.

Детектування відхилень сприяє зокрема вдосконаленню системи моніторингу й управління мережами. Це створює можливості для швидкого реагування на потенційні загрози та мінімізує можливі збитки від кібератак.

2 ВИКОРИСТАННЯ МАШИННОГО НАВЧАННЯ ДЛЯ ВИЯВЛЕННЯ АНОМАЛІЙ МЕРЕЖЕВОГО ТРАФІКУ

2.1 Типи задач машинного навчання

Машинне навчання — це галузь штучного інтелекту та інформатики, яка зосереджена на використанні даних та алгоритмів для імітації способу навчання людей з поступовим підвищенням його точності [1]. Алгоритми виявляють приховані закономірності та роблять прогнози на нових наборах даних без явного програмування для кожної задачі.

Виділяють три типи машинного навчання:

- З учителем — відбувається з використанням попередньо розмічених даних з метою отримання прогнозу для нових зразків. Під «учителем» мають на увазі набір тих самих вхідних навчальних даних, де вихідні сигнали (мітки) вже відомі. Підкатегоріями такого навчання є класифікація (передбачення категорії об'єкта згідно з визначеними заздалегідь класами) та регресія (прогнозування безперервного числового значення цільової змінної).
- Без учителя — дає змогу дослідити структуру даних для отримання значущої інформації без застосування міток і зворотного зв'язку. Серед підвидів виділяють кластеризацію (розбиття вибірки на кластери — групи подібних між собою об'єктів; за своєю суттю ця задача може вважатися подібною до класифікації без учителя) та зниження розмірності (представлення вибірки з використанням меншої кількості ознак зі збереженням значущих властивостей вихідних даних).
- З підкріпленням — метою є розробка системи, що покращує свої характеристики на основі взаємодії зі середовищем. Позаяк інформація про поточний стан середовища зазвичай містить так званий сигнал нагороди, можна вважати навчання з підкріпленням як споріднене до навчання з учителем. Однак у першому випадку зворотний зв'язок буде не

істинною міткою чи значенням, а мірою того, наскільки добре дія була оцінена функцією нагород.

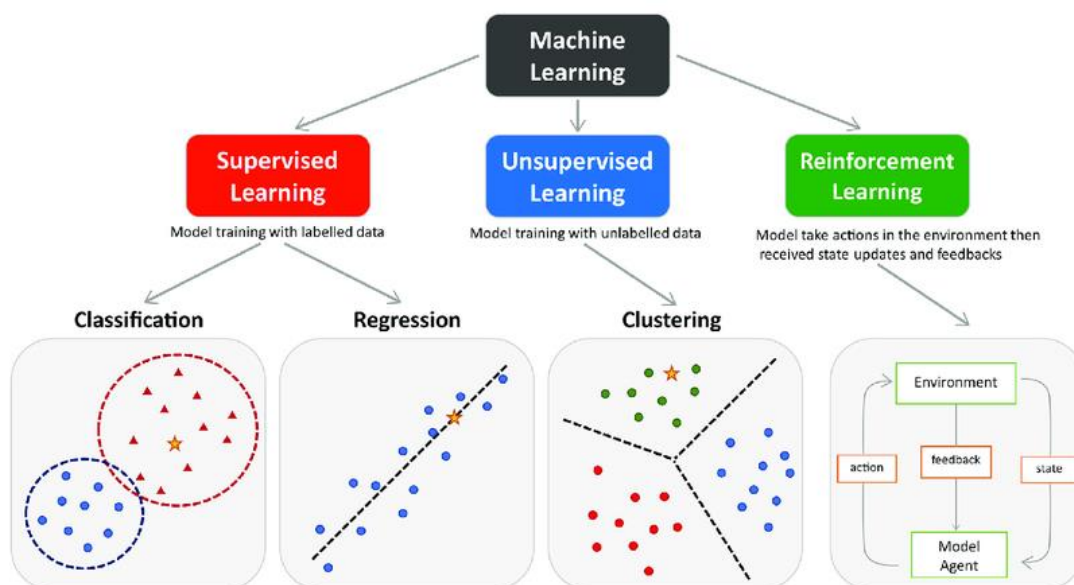


Рисунок 2.1 — Порівняльна схема типів машинного навчання

У контексті детектування аномалій широко застосовуються перші два типи навчання. Кожен має свої особливості, що дозволяє підійти до задачі аналізу з різних сторін. Далі в цьому розділі будуть розглянуті більш детально особливості цих алгоритмів, їх приклади, а також проаналізовано їх переваги та недоліки.

2.2 Використання класифікації для аналізу мережевого трафіку

2.2.1 Random Forest

Створює множину дерев рішень і використовує їх для передбачення класу об'єктів. Для побудови кожного дерева випадковим чином вибирається підмножина навчальних даних і підмножина ознак. Під час побудови дерева використовується критерій інформативності, що дозволяє вибрати найкращу ознаку для розбиття даних на кожному рівні. У результаті дерева в ансамблі створюються дещо різними, завдяки чому зменшується ефект перенавчання та підвищується якість передбачень. Далі для кожного об'єкта відбувається голосування за всіма деревами та вибір найбільш популярного класу.

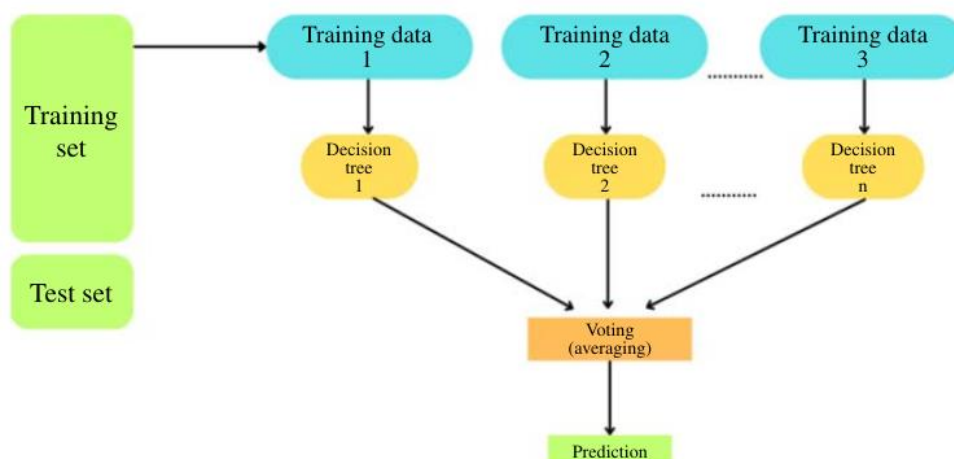


Рисунок 2.2 — Схема роботи Random Forest

2.2.2 Naive Bayes

Заснований на теоремі Баєса з припущенням про незалежність усіх предикторів один від одного. Тобто, наявність одної ознаки в класі не пов'язана з наявністю інших. Це означає, що кожна ознака робить внесок у рішення про класифікацію. Хоча наведене припущення нечасто виконується на реальних даних, цей класифікатор може добре спрацьовувати в складних задачах. Під час класифікації для кожного класу обчислюється апіорна ймовірність — ймовірність того, що випадково вибраний об'єкт належить певному класу без урахування будь-яких предикторів. Потім для кожної ознаки обчислюється умовна ймовірність того, що об'єкт належить певному класу за умови, що значення цієї ознаки дорівнює певному значенню.

2.2.3 K-Nearest Neighbors (KNN)

Призначає мітку класу нового об'єкта з урахуванням класів найближчих k сусідніх об'єктів. Алгоритм є чутливим до дисбалансу класів, тому може «зміщувати» рішення в сторону панівного класу. Як рішення цієї проблеми часто використовують метод зваженого голосування, коли вплив сусідніх точок на визначення класу нової точки буде обернено пропорційним до квадрата відстані між ними.

2.2.4 Support Vector Machine (SVM)

Використовує ядро для перетворення простору ознак, яке визначає функцію подібності між об'єктами в цьому просторі. Кожне ядро (лінійне, гауссове,

поліноміальне, сигмоїдальне) може мати свої параметри для підлаштування під конкретні дані. Алгоритм відрізняється високою точністю та здатністю працювати з високими розмірностями даних, проте є чутливим до шумів і викидів і може потребувати значних обчислювальних ресурсів на великих вибірках.

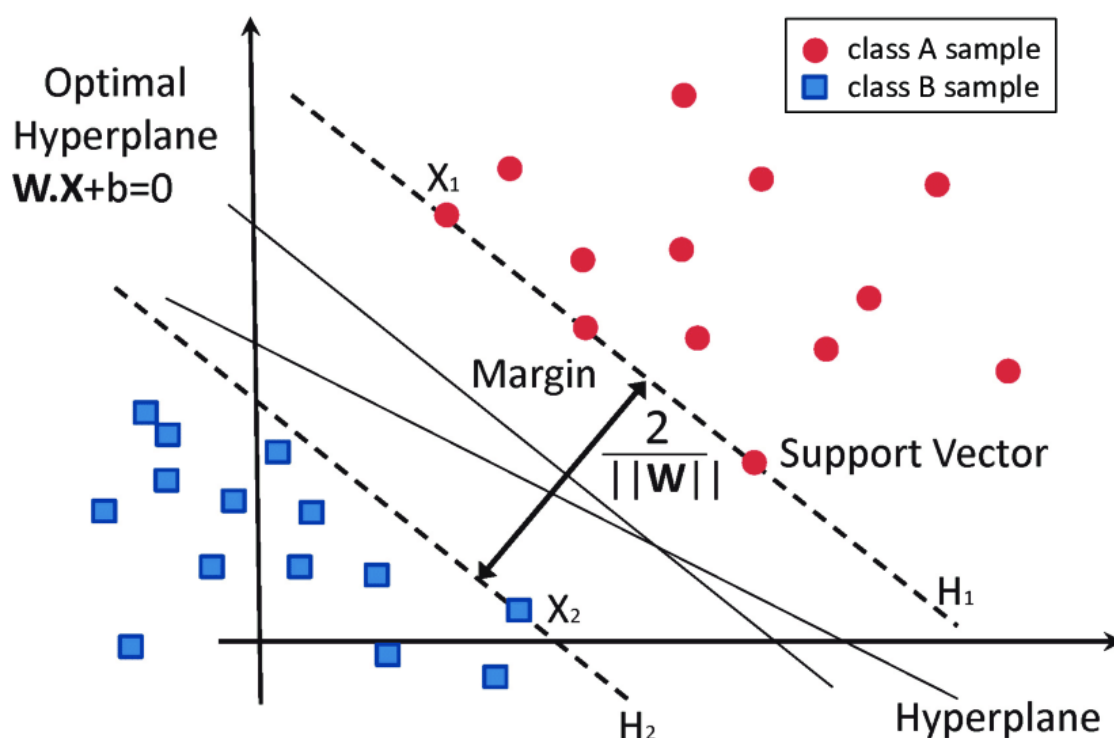


Рисунок 2.3 — Візуалізація роботи лінійного випадку SVM

2.3 Пошук аномалій методами кластеризації

2.3.1 K-Means

Найпростішим алгоритмом кластеризації є метод k -середніх. Спочатку визначається кількість кластерів k , на які будуть розділятися дані. Центри цих кластерів на першому етапі визначаються випадковим чином. Далі кожен об'єкт з набору даних відноситься до того кластера, центроїд якого знаходиться найближче до нього. Після цього відбувається оновлення значень центроїд — їм призначається середнє значення за всіма об'єктами, віднесеними до відповідних кластерів. Ці етапи повторюються до стабілізації центроїд (мінімізації сумарної квадратичної відстані між точками центрами кластерів).

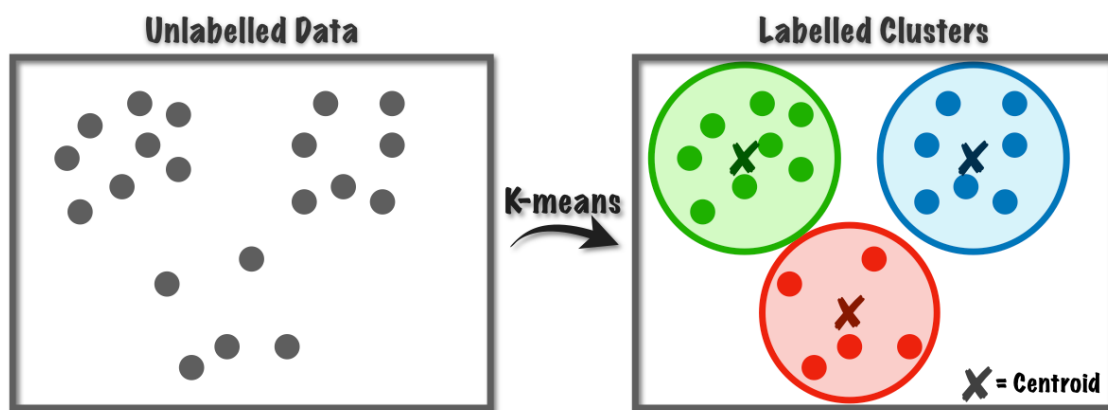


Рисунок 2.4 — Візуалізація роботи K-Means

2.3.2 Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

Базується на щільності даних та ідентифікує кластери на наборах даних, які містять шум і викиди. Метод приймає на вхід два параметри:

- *Epsilon*, ϵ — максимальна відстань між двома точками, за якої вони будуть вважатися «сусідами»;
- *Minimum Points*, *MinPts* — мінімальна кількість точок у радіусі ϵ , необхідна для того, щоб точка вважалася частиною скупчення.

Самі ж точки поділяють на:

- кореневі (core points) — містять хоча б *MinPts* сусідніх точок в ϵ -околі;
- досяжні (border points) — мають менше ніж *MinPts* сусідніх точок, але знаходяться в ϵ -околі основної точки;
- шумові (noise points).

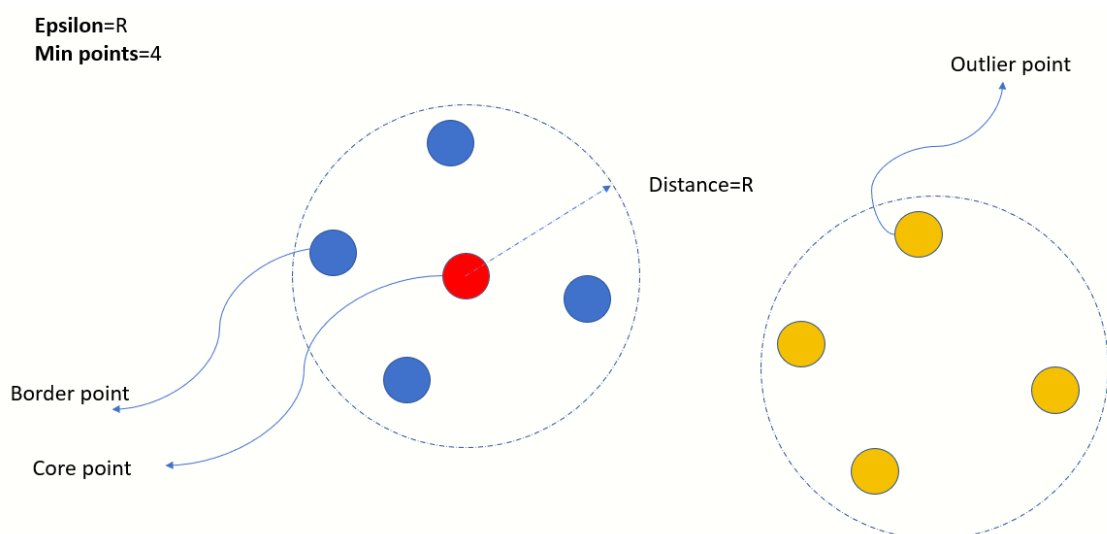


Рисунок 2.5 — Параметри та типи точок у DBSCAN

Спершу вибирається довільна точка та визначається її тип. Якщо вона є кореневою, починається процес розширення кластера шляхом додавання всіх точок з ϵ -околу. Досяжні точки також додаються до кластера, проте їхні околи далі не досліджуються, тому формування кластерів на них завершується. Процес триває доти, доки кожній точці не буде призначено її тип. Відповідно, ті точки, що не потрапили в жодний кластер, визначаються як шумові.

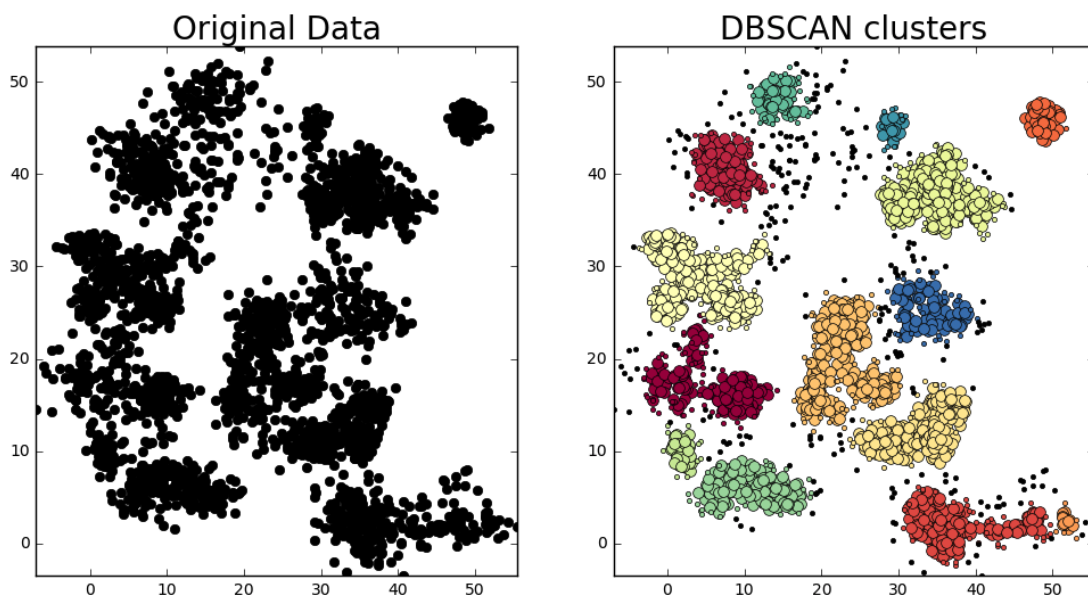


Рисунок 2.6 — Візуалізація роботи DBSCAN

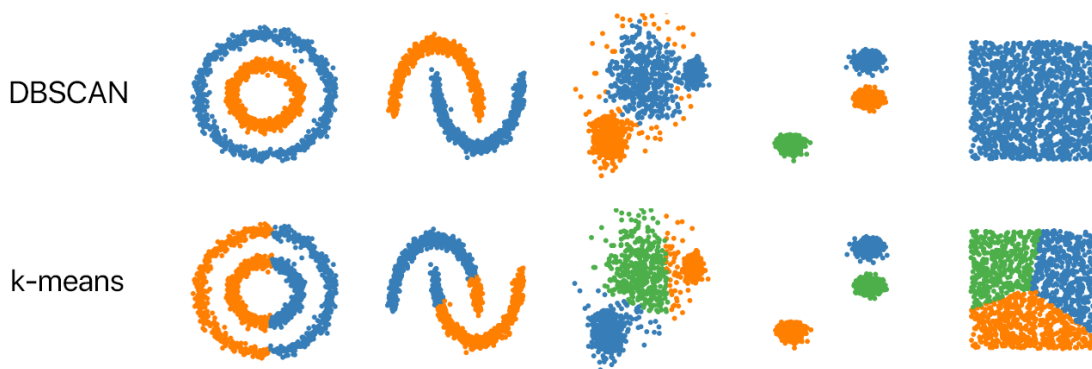


Рисунок 2.7 — Порівняння роботи DBSCAN і K-Means

2.3.3 Isolation Forest

Ґрунтується не на формуванні кластерів, а саме на пошуку аномалій серед даних. Алгоритм ізолює дані, вибираючи випадковий атрибут і довільно розбиваючи діапазон значення. Процес повторюється рекурсивно, створюючи дерево. Багато таких дерев формують ліс для ізоляції кожної точки в наборі даних. Таким чином, чим швидше певне спостереження відокремлюється від інших, тим

більша ймовірність того, що воно є аномалією. Метод є особливо корисним, коли на великих наборах даних потрібно швидко ідентифікувати аномалії, які є рідкісними та не мають чітко визначеної структури чи взаємозв'язків з нормальними об'єктами.

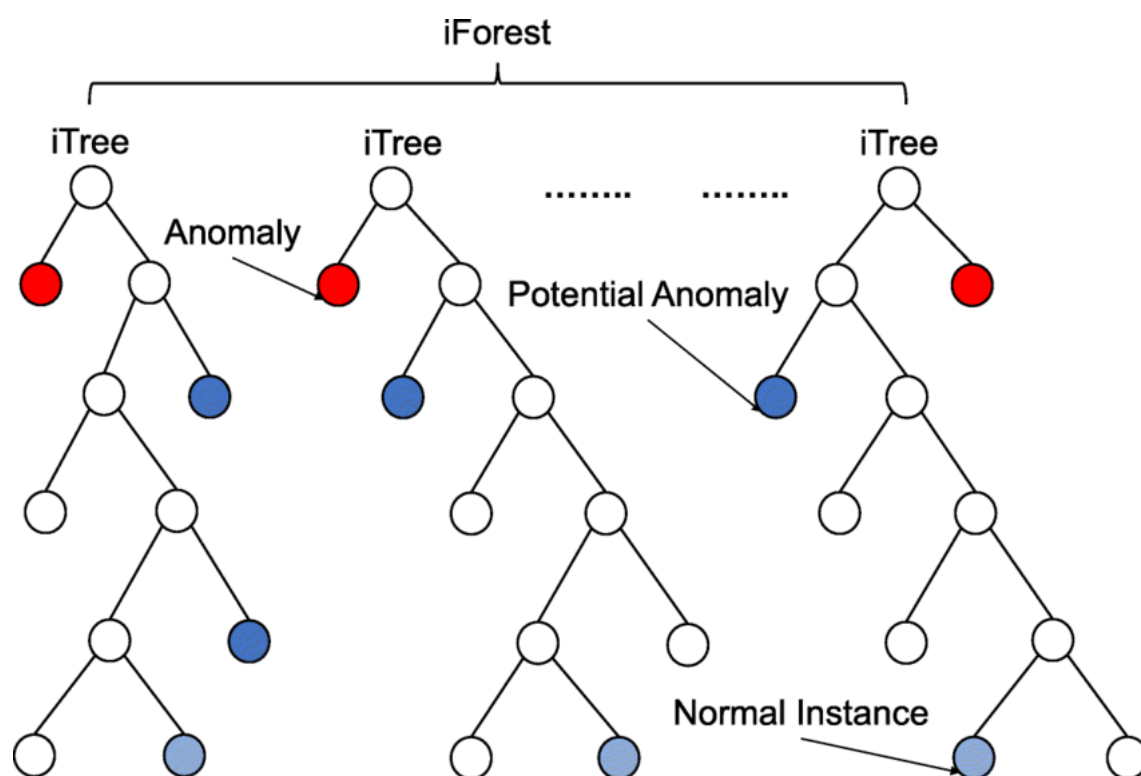


Рисунок 2.8 — Візуалізація роботи Isolation Forest

2.3.4 Gaussian Mixture Model (GMM)

Працює шляхом моделювання даних як лінійної комбінації кількох гауссових розподілів, кожен з яких характеризується середнім значенням і коваріаційною матрицею. GMM використовує для навчання алгоритм максимізації очікування для ітеративного поліпшення параметрів. Для спостережень обчислюється ймовірність належності до кожного з гауссових розподілів, після чого відбувається оновлення параметрів на основі обчислених ймовірностей. Процес повторюється до досягнення збіжності. Для виявлення аномалій можна використовувати перевірку ймовірностей належності: спостереження з низькими ймовірностями належності до будь-якого з кластерів можуть вважатися аномальними. Недоліками моделі є чутливість до початкових умов та обчислювальна складність.

Висновки до розділу 2

Моделі, що базуються на навчанні з учителем, можуть розпізнавати знайомі патерни атак на основі попередніх інцидентів. Дослідження вказують на високу точність виявлення відомих атак. Однак очевидним недоліком цих методів є необхідність у наявності попередньо розмічених навчальних наборів. І хоча такі зразки можна отримати із загальнодоступних ресурсів, коли йдеться про пошук підозрілого трафіку на реальних даних, задача ускладнюється.

Алгоритми ж кластеризації є потужним інструментом для роботи з нерозміченими даними. Вони можуть виявляти приховані структури, що дозволяє знаходити аномалії, які не вписуються в ці структури. Однак їх ефективність залежить від правильного підбору параметрів, а ресурси для обчислень можуть суттєво зростати зі збільшенням розмірності даних, що аналізуються.

Обидвоє методів дозволяють шукати аномалії трафіку в умовах значної невизначеності, надаючи значних можливостей для їх застосування для виявлення мережових атак.

3 КОМБІНОВАНИЙ МЕТОД ПОШУКУ АНОМАЛІЙ

3.1 Сутність комбінованого методу

З огляду на інформацію, викладену в розділі 2, пропонується застосовувати комбінований метод для аналізу даних мережевого трафіку. Ідея полягає в об'єднанні двох типів машинного навчання. Передбачається, що такий підхід дасть змогу поєднати переваги обох методів і частково нівелювати їхні недоліки.

Пропонується на першому кроці використовувати кластеризацію для попередньої розмітки невеликої частини набору «сирих» даних про мережевий трафік. Розміри цієї частини можуть залежати від активності в мережі. Далі ці дані стануть навчальною вибіркою для класифікатора, який буде працювати з іншою частиною датасету, а також на тому трафіку, що отримуватиметься в реальному часі. У результаті буде отримано модель навчання з учителем, де в ролі вчителя виступатиме навчання без учителя.

Розмічення даних необхідно проводити з певною періодичністю, яка визначається активністю в конкретній мережі. Наведена умова є доволі природною, позаяк неможливо один лише раз визначити прийнятний приклад трафіку та орієнтуватися лише на нього — підхід має бути динамічним і пристосовуватися до змін.

Очікувані переваги запропонованого способу:

- робота на нерозміченому мережевому трафіку;
- прогнозування міток на основі класифікації;
- зменшення витрат обчислювальних ресурсів внаслідок зниження розмірності кластеризованого набору;
- можливість знаходження прихованих структур і навчання класифікатора на основі них;
- дослідження нових даних відокремлено від попередніх.

3.2 Підбір алгоритмів і тестових даних

3.2.1 Використані алгоритми

Для кластеризації було вибрано метод DBSCAN. Важливим етапом перед його використанням є встановлення гіперпараметрів. Зазвичай їх підбирають, спираючись на особливості конкретного датасету чи шляхом перебору списку значень. Метою комбінованого методу є максимально автономна його робота, що не потребуватиме «ручного» налаштування. Тобто, необхідно задати універсальну формулу розрахування параметрів.

Нехай аргументи обчислюватимуться наступним чином. Позначимо k як $n + 1$, де n — кількість предикторів після застосування методу головних компонент, які пояснюють 95% варіації в даних. Тоді:

- ϵ буде визначатися як середнє значення відстані до k найближчих сусідів;
- $MinPts$ буде дорівнювати значенню k .

Після цього використовуватиметься наївний баєсів класифікатор. Основна його перевага полягає в простоті та швидкодії. Алгоритм не вимагає великої кількості обчислювальних ресурсів і даних для навчання та може ефективно працювати на датасетах з високою розмірністю, що містять багато предикторів.

3.2.1 Тестові дані

CSE-CIC-IDS2018 — спільний проєкт Канадського інституту кібербезпеки (Canadian Institute for Cybersecurity) та Служби безпеки зв'язку (Communications Security Establishment). Цей датасет містить дані мережевого трафіку та журнали системних подій для досліджень у галузі виявлення вторгнень та кібербезпеки.

Набір містить близько 16 мільйонів записів, зібраних протягом 10 днів. Включає 7 різних сценаріїв атак: Brute-force, Heartbleed, Botnet, DoS, DDoS, вебатаки, внутрішнє проникнення в мереж. Атаки були проведені з використанням 50 машин, тоді як організація-жертва складалася з 5 відділів із 420 машин і 30 серверів.

	Dst Port	Protocol	Flow Duration	Tot Fwd Pkts	Tot Bwd Pkts	TotLen Fwd Pkts	TotLen Bwd Pkts	Fwd Pkt Len Max	Fwd Pkt Len Min	Fwd Pkt Len Mean	...	Fwd Seg Size Min	Active Mean	Active Std	Active Max	Active Min
0	0	0	112641158	3	0	0	0	0	0	0.000000	...	0	0.0	0.000000	0	0
1	22	6	37366762	14	12	2168	2993	712	0	154.857143	...	32	1024353.0	649038.754495	1601183	321569
2	47514	6	543	2	0	64	0	64	0	32.000000	...	32	0.0	0.000000	0	0
3	0	0	112640703	3	0	0	0	0	0	0.000000	...	0	0.0	0.000000	0	0
4	0	0	112640874	3	0	0	0	0	0	0.000000	...	0	0.0	0.000000	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
401754	52065	6	2	3	0	31	0	31	0	10.333333	...	20	0.0	0.000000	0	0
401755	53	17	9132	1	1	29	45	29	29	29.000000	...	8	0.0	0.000000	0	0
401756	53	17	37016	1	1	44	104	44	44	44.000000	...	8	0.0	0.000000	0	0
401757	53	17	451	1	1	30	62	30	30	30.000000	...	8	0.0	0.000000	0	0
401758	80	6	77458	3	4	437	859	437	0	145.666667	...	20	0.0	0.000000	0	0

400000 rows × 79 columns

Рисунок 3.1 — Частина вмісту файлу CSE-CIC-IDS2018

Попередня обробка містила відбір перших 400 тисяч записів, кодування цільової змінної з мітками класів, розбиття даних на вибірки для кластеризації та класифікації (15% і 85% відповідно). Також було проведено заміну нескінченних значень на середні значення за 5 найближчими сусідами, стандартизацію та застосовано метод головних компонент.

3.3 Аналіз результатів роботи методу

Після тестування методу на даних з інформацією про різні атаки та різною кількістю аномалій було виявлено пряму залежність між співвідношенням класів трафіку та точністю прогнозування: чим вища частка аномальних подій, тим краще модель передбачає класи цих подій. Неточність зумовлено тим, що за налаштування гіперпараметрів на основі розмірності даних, а не ручного підбору, DBSCAN ідентифікує абсолютну більшість об'єктів як викиди. Це створює обмеження у використанні запропонованого методу та чітко визначає випадки, у яких він може бути корисним.

Нижче представлено результати роботи алгоритму для 5 різних файлів у порядку зростання відсотка аномального трафіку.

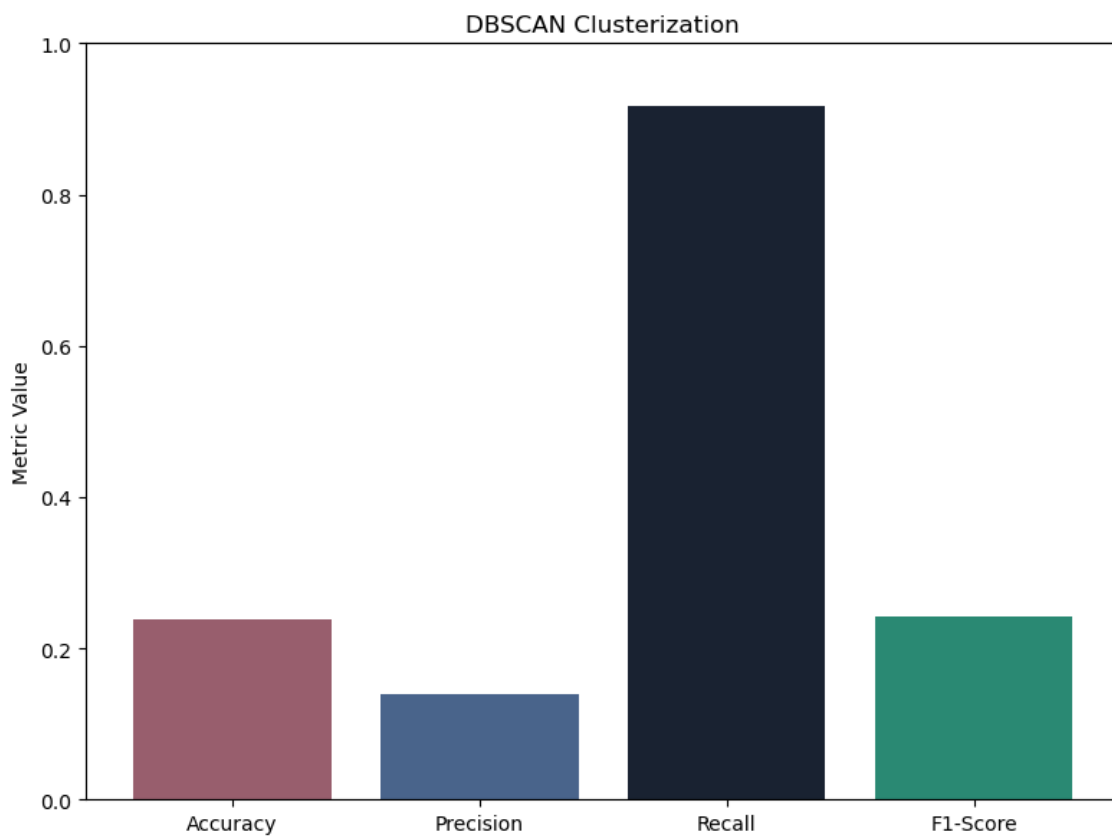


Рисунок 3.2 — Точність кластеризації для даних за 02-15-2018

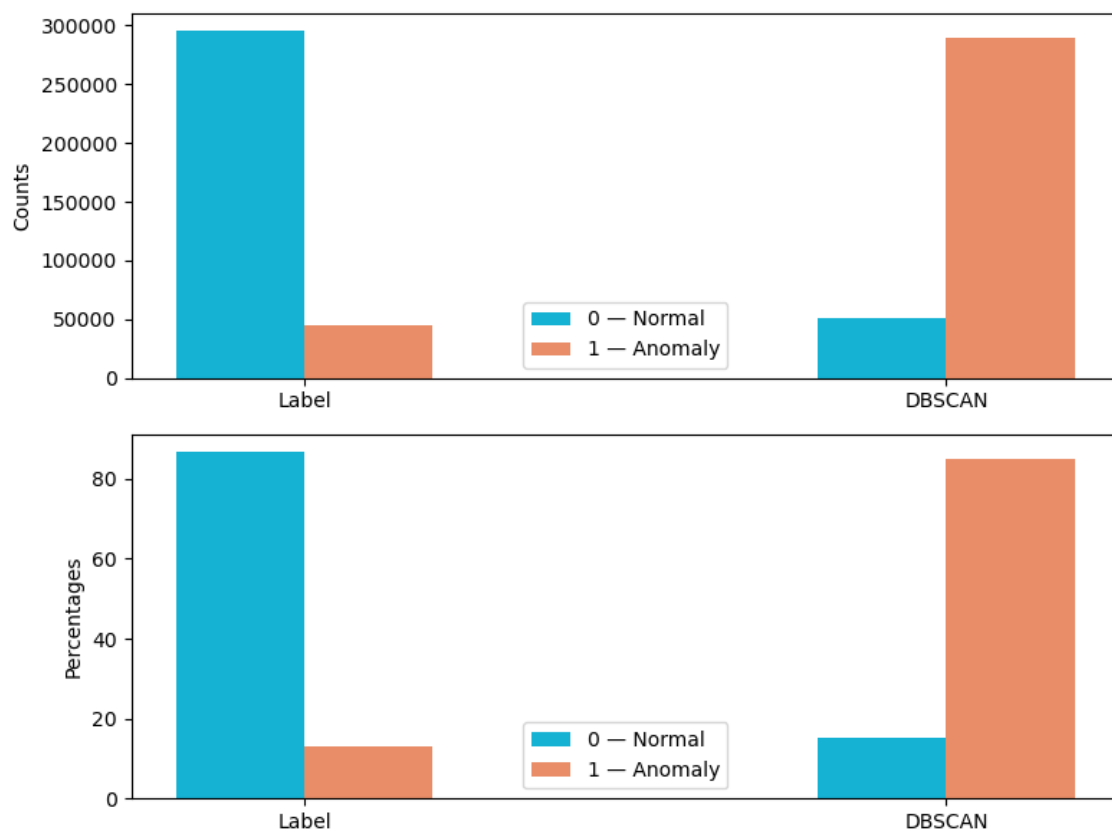


Рисунок 3.3 — Співвідношення істинних міток класів до передбачених на даних за 02-15-2018

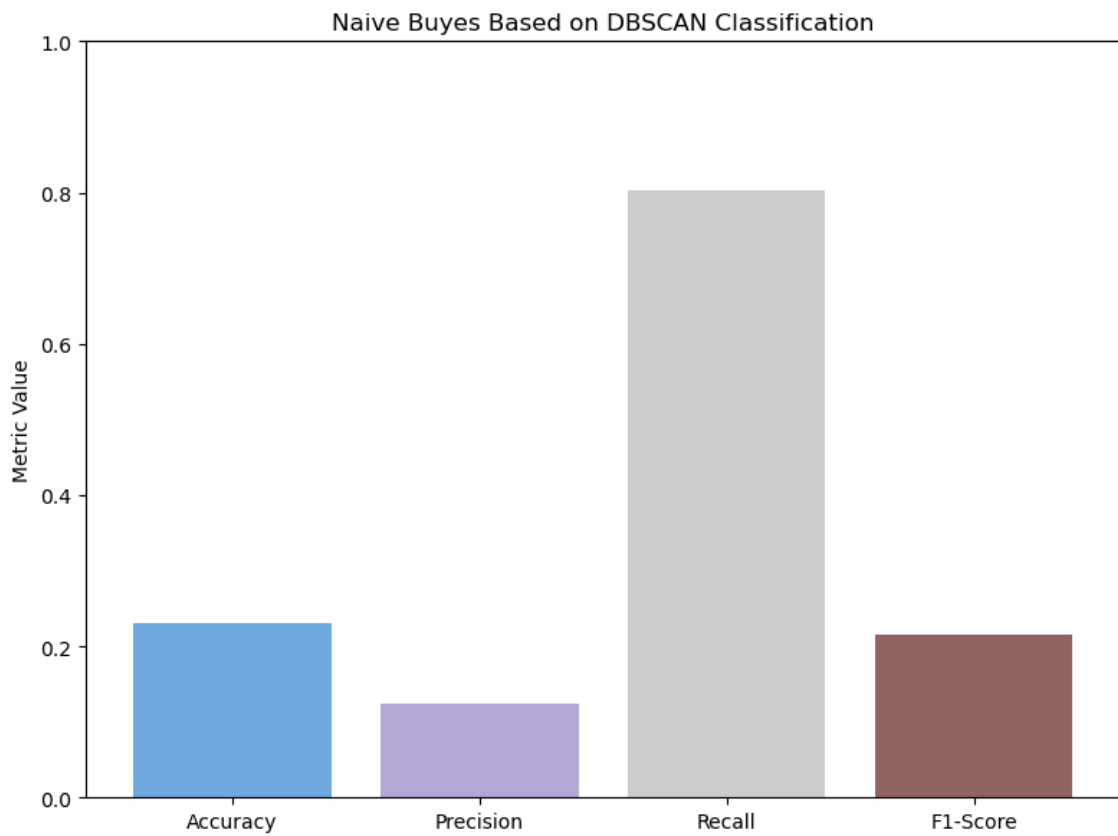


Рисунок 3.4 — Метрики якості передбачень класифікації на даних за 02-15-20

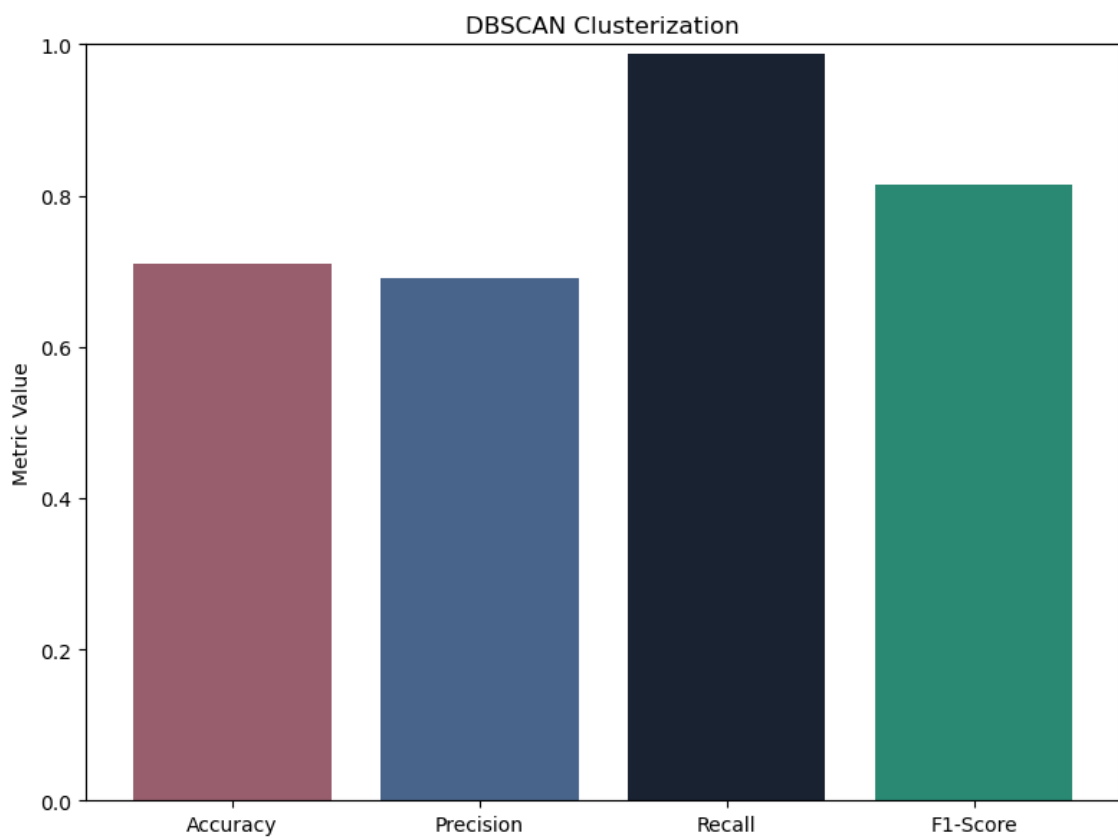


Рисунок 3.5 — Точність кластеризації для даних за 02-16-2018

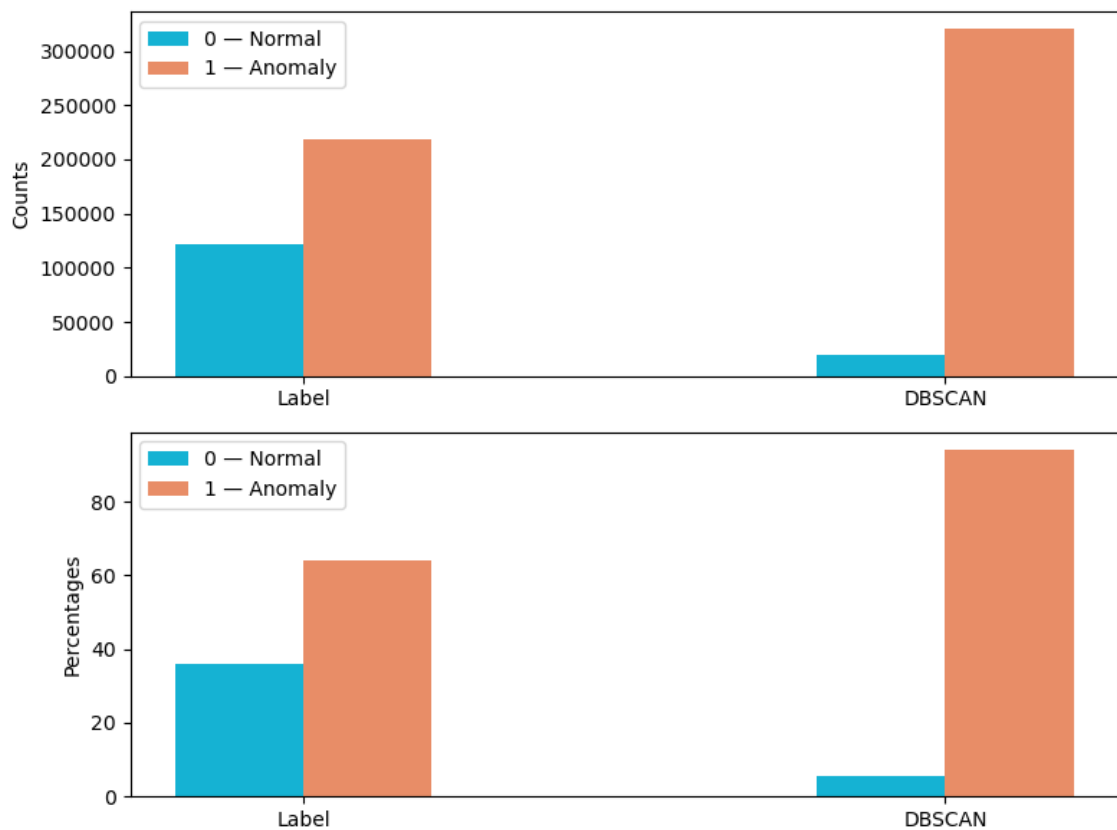


Рисунок 3.6 — Співвідношення істинних міток класів до передбачених на даних за 02-16-2018

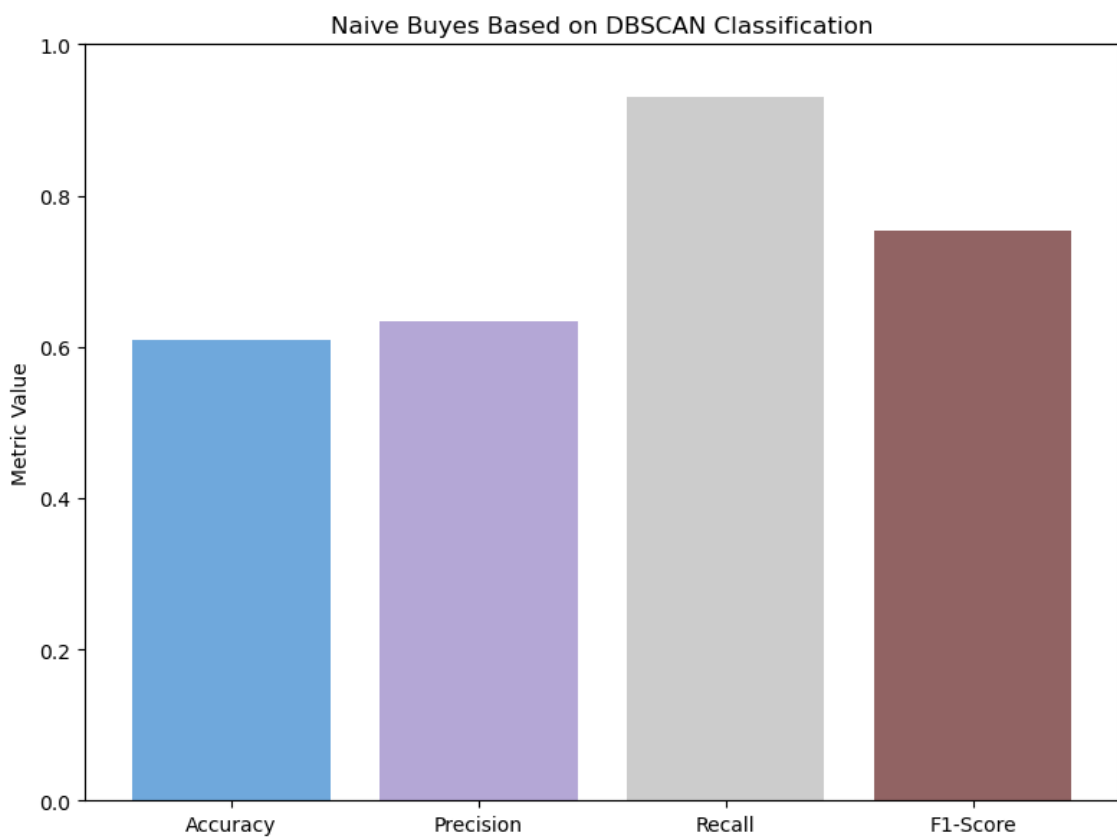


Рисунок 3.7 — Метрики якості передбачень класифікації на даних за 02-16-20

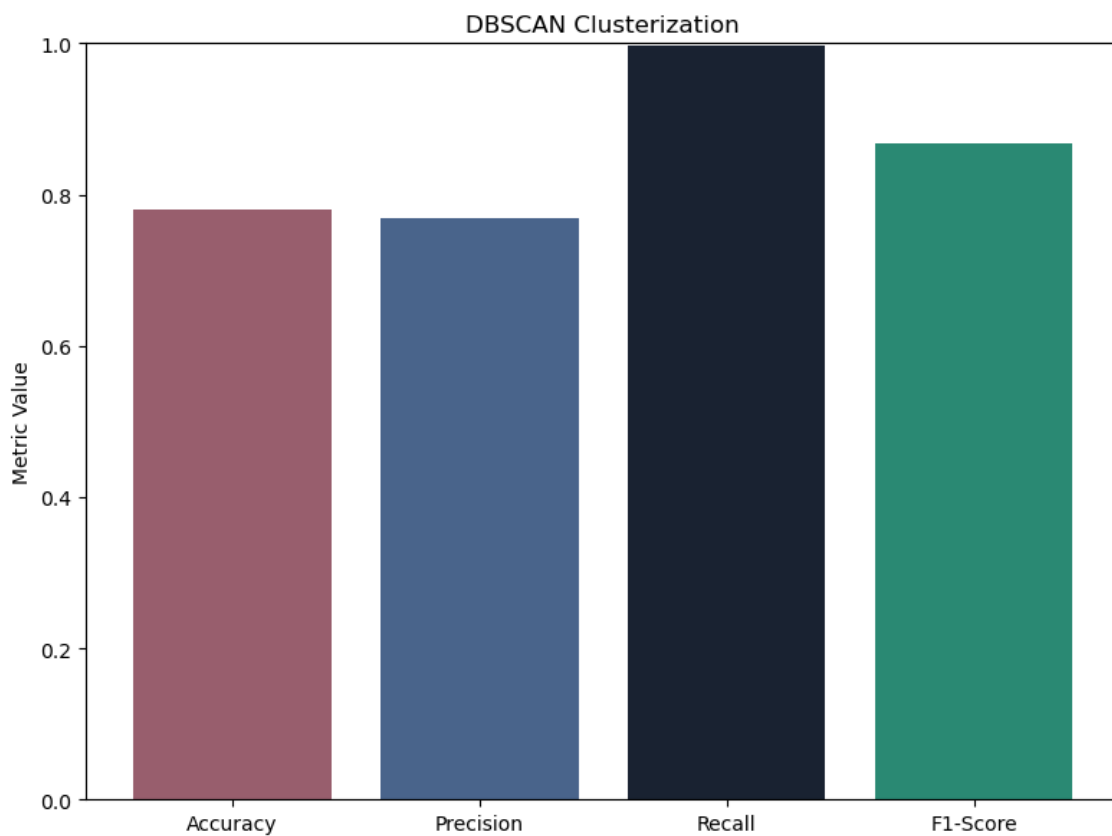


Рисунок 3.8 — Точність кластеризації для даних за 03-02-2018

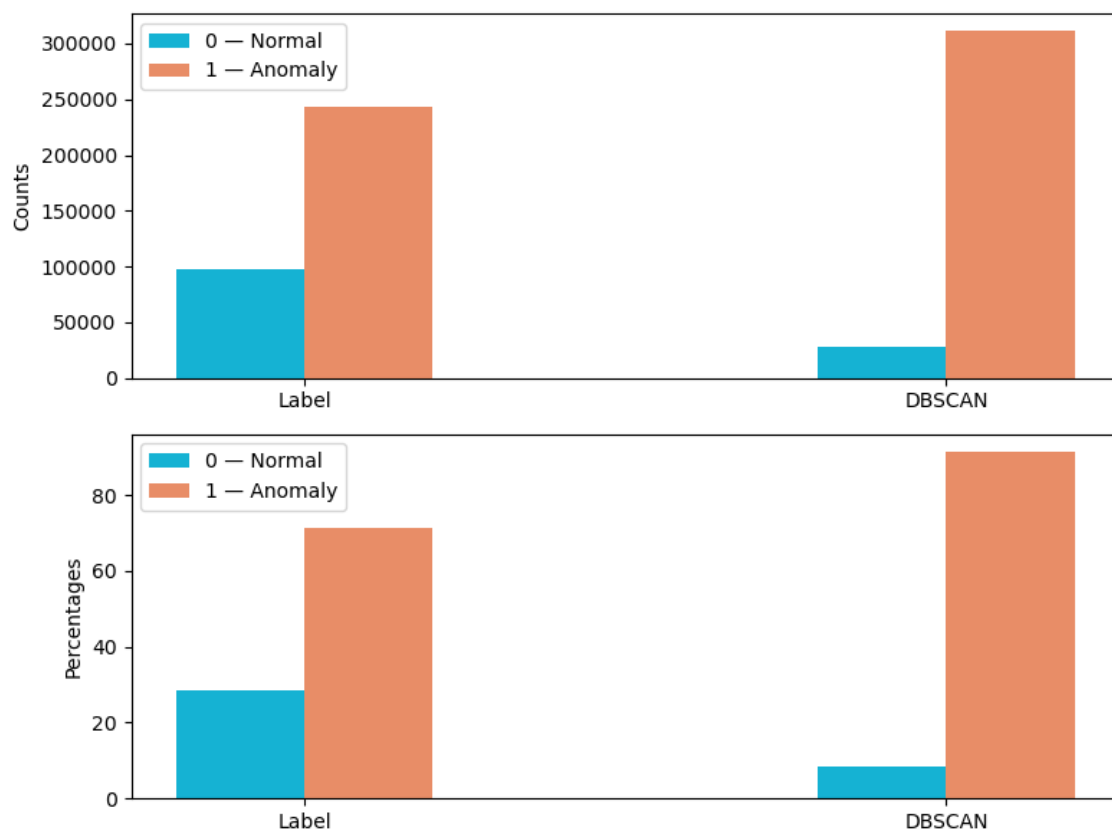


Рисунок 3.9 — Співвідношення істинних міток класів до передбачених на даних за 03-02-2018

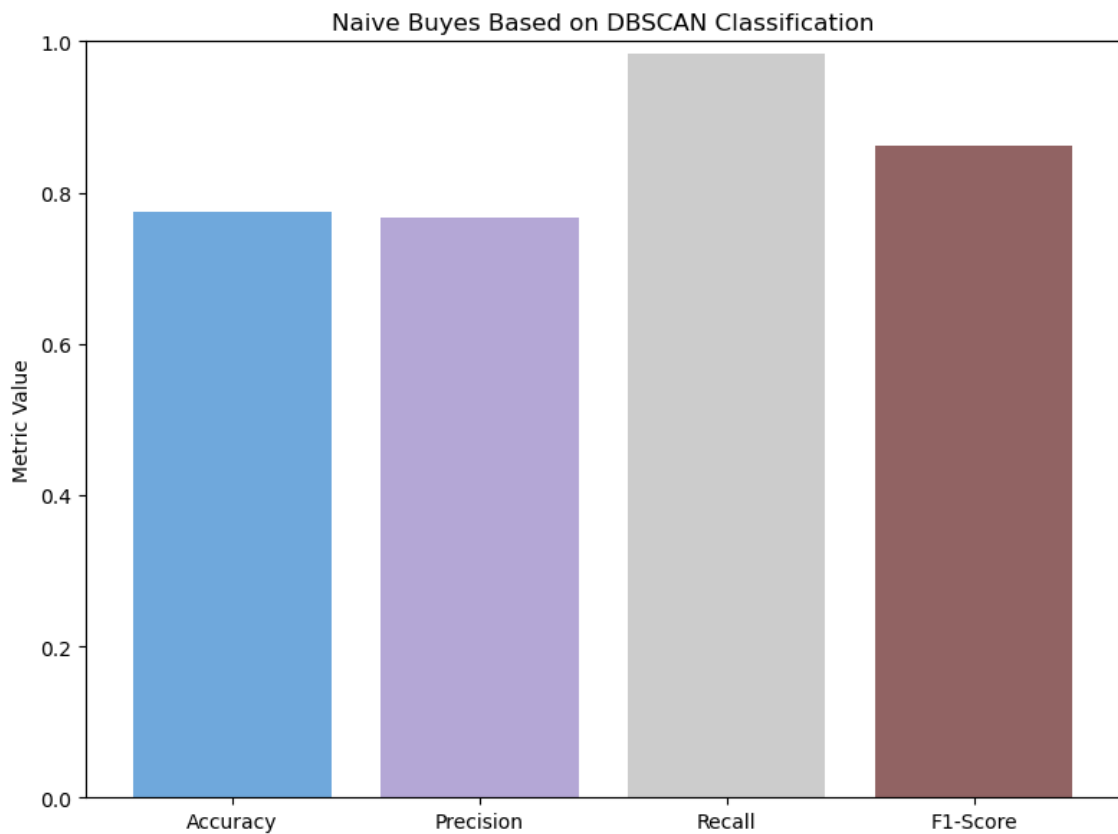


Рисунок 3.10 — Метрики якості передбачень класифікації на даних за 03-02-20

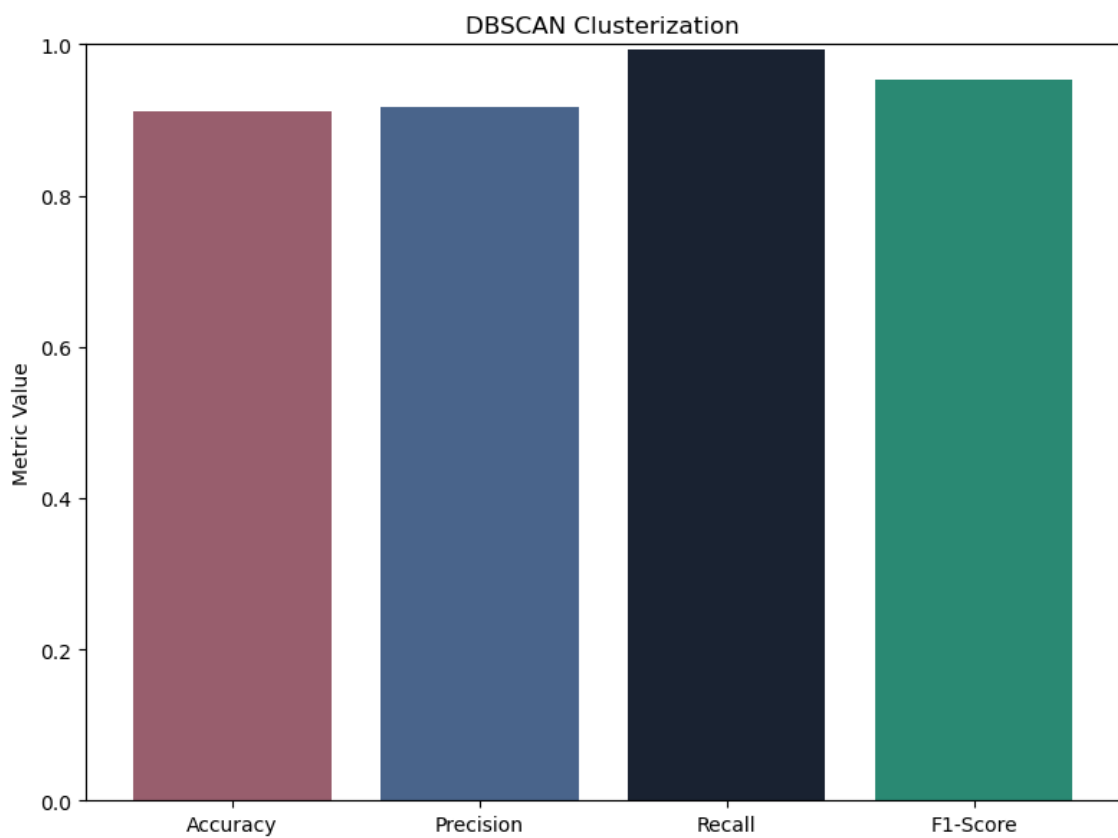


Рисунок 3.11 — Точність кластеризації для даних за 02-21-2018

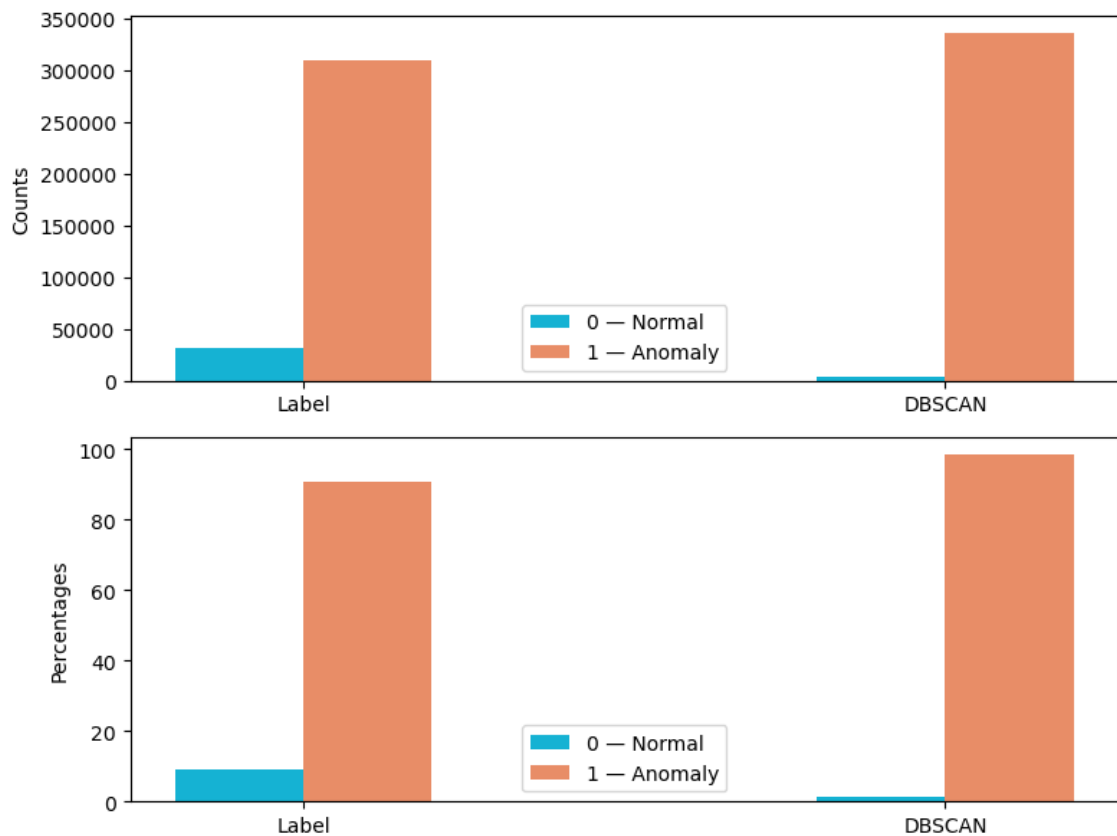


Рисунок 3.12 — Співвідношення істинних міток класів до передбачених на даних за 02-21-2018

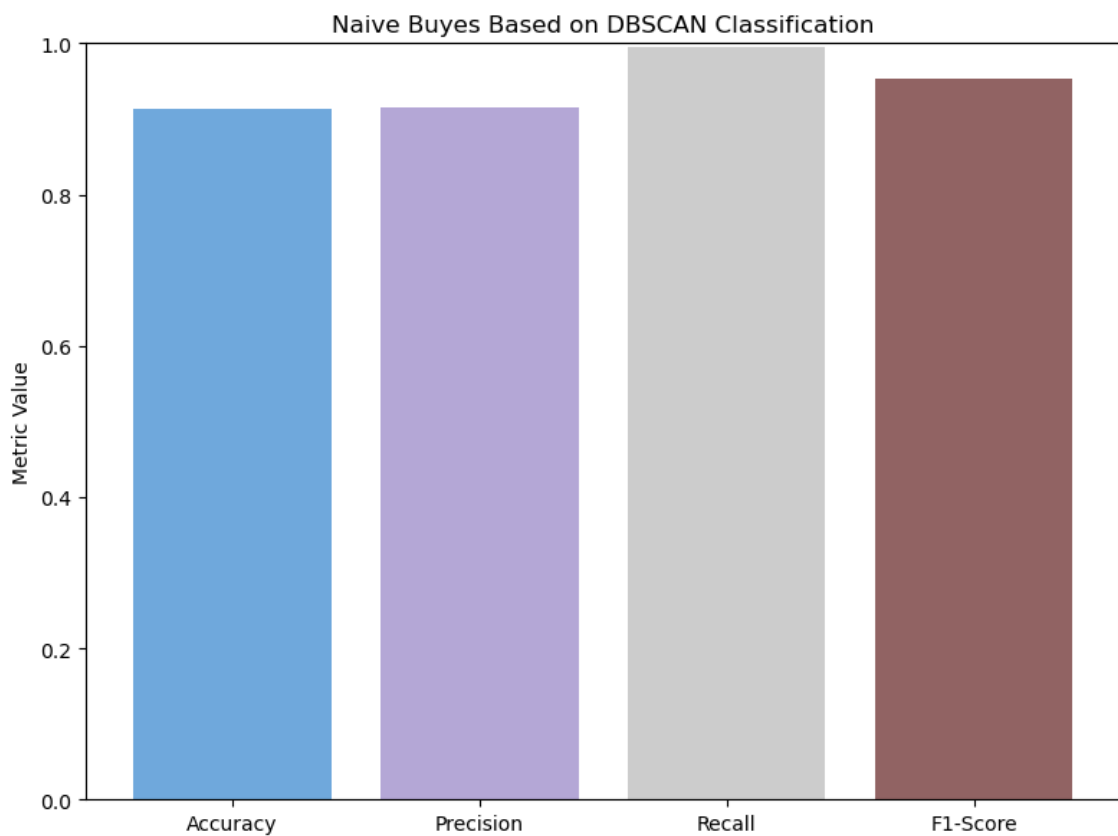


Рисунок 3.13 — Метрики якості передбачень класифікації на даних за 02-21-20

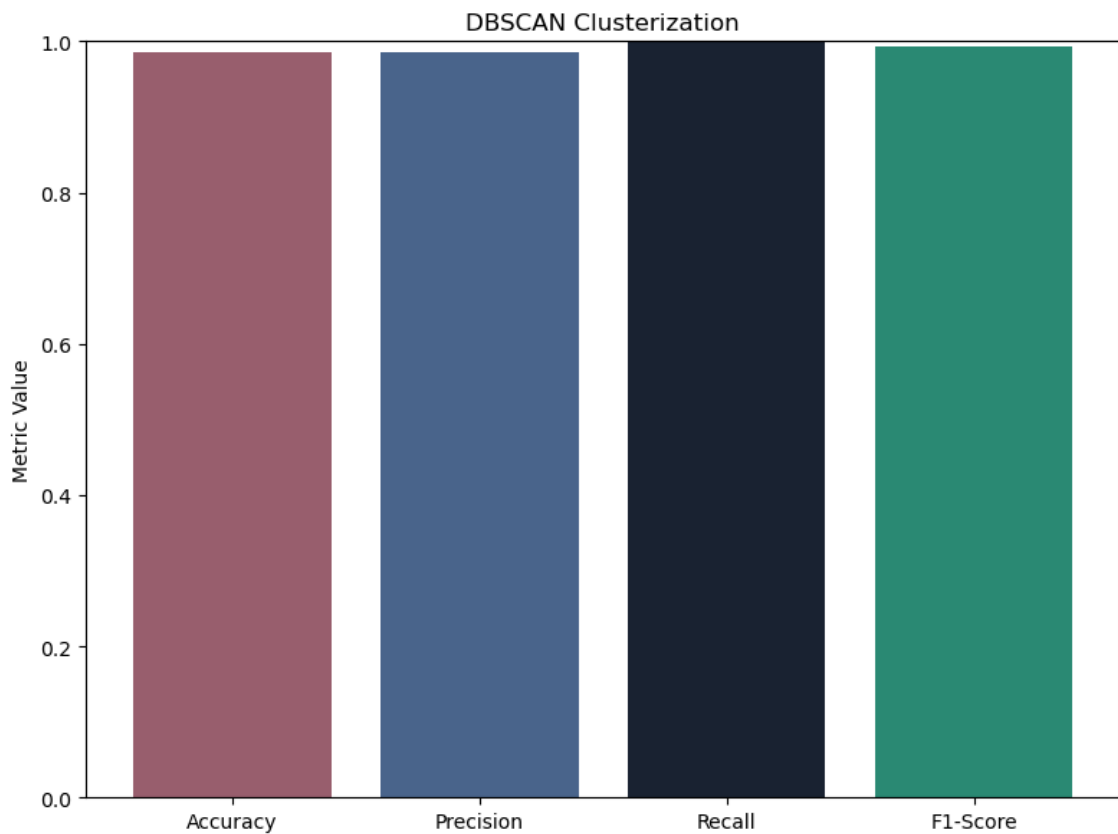


Рисунок 3.14 — Точність кластеризації для даних за 02-14-2018

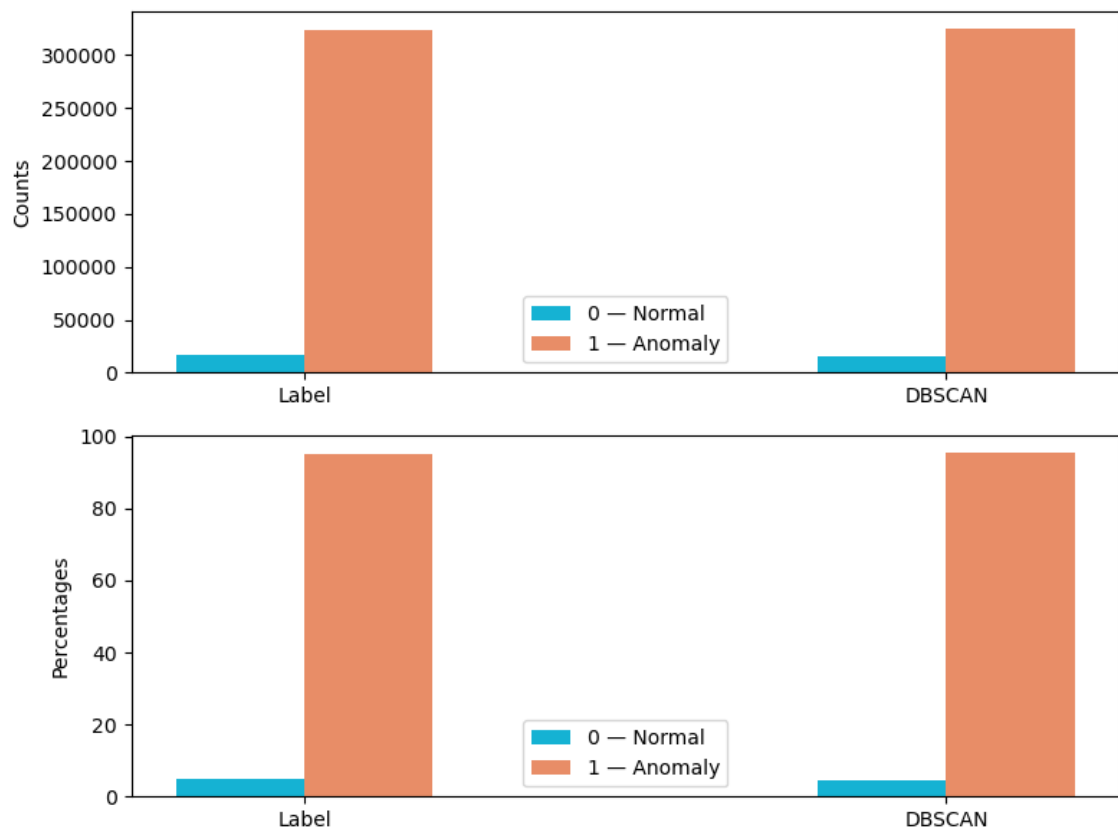


Рисунок 3.15 — Співвідношення істинних міток класів до передбачених на даних за 02-14-2018

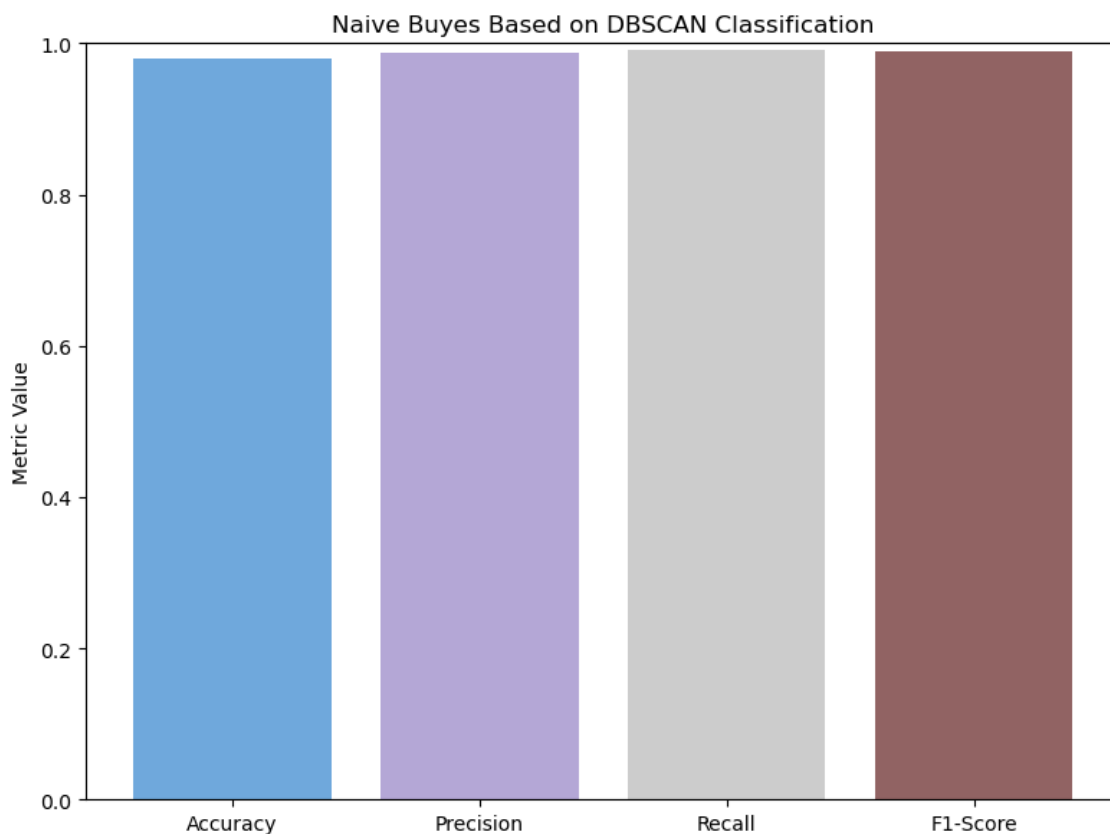


Рисунок 3.16 — Метрики якості передбачень класифікації на даних за 02-14-20

Таблиця 3.1 — Результати застосування методу

Нормальний трафік, %	Аномальний трафік, %	Accuracy	Precision	Recall	F1-Score
86,89	13,11	0,22	0,12	0,8	0,21
64,16	35,84	0,61	0,63	0,93	0,75
28,53	71,47	0,77	0,76	0,98	0,86
9,18	90,82	0,91	0,92	0,99	0,95
4,79	95,21	0,98	0,99	0,99	0,99

Висновки до розділу 3

Впровадження комбінованого методу пошуку аномалій в мережевому трафіку дозволяє значно покращити ефективність виявлення загроз. Об'єднання кластеризації та класифікації забезпечує комбінований підхід, що поєднує переваги обох методів, зменшуючи їх недоліки. Використання кластеризації для попередньої

розмітки даних дозволяє працювати з нерозміченими датасетами, знижуючи витрати на ручну розмітку та підготовку даних.

Запропонований метод також забезпечує динамічність і пристосовуваність до змін у мережевій активності. Автоматичний підбір гіперпараметрів для алгоритму DBSCAN і використання наївного баєсівського класифікатора дозволяють знизити обчислювальні витрати та підвищити швидкість обробки даних.

Натомість експерименти показали, що алгоритм кластеризації добре виявляє шкідливий трафік у тих ситуаціях, коли його кількість значно переважає над нормальною мережевою активністю. Якщо ж аномалій у мережі майже не спостерігається, метрики якості спрацювання кластеризації значно знижуються. Так, наприклад, за співвідношення 87:13 нормального до аномального трафіку значення $F1$ -міри складає близько 0,21. А за співвідношення 5:95 середнє гармонійне наближається до 0,99. Отже, такий метод має чітко визначені обмеження в його використанні.

ВИСНОВКИ

Під час виконання дипломної роботи було проаналізовано наявні підходи до виявлення мережових атак шляхом детектування аномалій методами машинного навчання. Було виділено основні переваги та недоліки таких підходів як кластеризація та класифікація, а також розглянуто конкретні приклади цих алгоритмів.

На основі цих міркувань створено концепцію метода, який би дозволяв об'єднати обидва види навчання моделей. Запропонований метод передбачає попередню розмітку даних мережевого трафіку за допомогою алгоритму DBSCAN і використання наївного баєсівського класифікатора для подальшої обробки вибірки. Розроблену програму протестовано на датасеті про мережевий трафік і мережеві атаки CSE-CIC-IDS2018.

У результаті оцінки роботи методу були виявлені такі його можливості:

- Запобігання залученню фахівців для попереднього аналізу нерозмічених даних.
- Автоматичний підбір параметрів для алгоритму кластеризації на основі розмірності даних.
- Застосування на реальних даних трафіку в режимі реального часу.
- Зниження використання обчислювальних ресурсів.

Разом із цим була виявлена доцільність його використання лише на даних, де частка аномальних даних мережевого трафіку значно перевищує частку нормальних

Отримані результати можуть стати підґрунтям для створення та подальших досліджень інших комбінованих методів, що може стати основою для впровадження нових підходів детектування аномалій у сучасних системах.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. What is machine learning (ML)? [Електронний ресурс] — Режим доступу: <https://www.ibm.com/topics/machine-learning>
2. Python Machine Learning / Vahid Mirjalili, Sebastian Raschka., 2015.
3. Buczak, A. L., & Guven, E. A survey of data mining and machine learning methods for cybersecurity intrusion detection [Електронний ресурс] / A. L. Buczak, E. Guven. — 2016. — Режим доступу: <https://ieeexplore.ieee.org/document/7307098>
4. Computer Network Attack [Електронний ресурс] — Режим доступу: https://csrc.nist.gov/glossary/term/computer_network_attack
5. What is anomaly detection? [Електронний ресурс] — Режим доступу: <https://www.ibm.com/topics/anomaly-detection#:~:text=In%20business%20data%2C%20three%20main,rest%20of%20the%20data%20set>
6. IDS 2018 Intrusion CSVs (CSE-CIC-IDS2018) [Електронний ресурс] — Режим доступу: <https://www.kaggle.com/datasets/solarmainframe/ids-intrusion-csv/data?select=02-16-2018.csv>
7. Autoencoding Variational Bayes [Електронний ресурс] — Режим доступу: <https://arxiv.org/pdf/1402.0856>
8. Applications of Artificial Intelligence in Cybersecurity [Електронний ресурс] — Режим доступу: <https://www.mdpi.com/2076-3417/13/7/4151>
9. Harnessing the Power of K-Means for Anomaly Detection [Електронний ресурс] — Режим доступу: <https://medium.com/@tommaso.romani2000/harnessing-the-power-of-k-means-for-anomaly-detection-24dc71d260a8>
10. Artificial Intelligence in Cybersecurity Analytics and Cyber Threat Detection.
11. Hands-On Artificial Intelligence for Cybersecurity / Alessandro Parisi., 2019.
12. Best Clustering Algorithms for Anomaly Detection [Електронний ресурс] — Режим доступу: <https://towardsdatascience.com/best-clustering-algorithms-for-anomaly-detection-d5b7412537c8>

13. Nguyen, T. T., & Armitage, G. A survey of techniques for internet traffic classification using machine learning. *IEEE Communications Surveys & Tutorials*, 10(4), 56-76.
14. Cybersecurity and AI Techniques [Электронный ресурс] — Режим доступа: <https://www.mdpi.com/1424-8220/24/6/1968>

ДОДАТОК А

```
import os
import pandas as pd

folder_path = 'CIC-IDS-2018/02-14-2018.csv'
df = pd.read_csv(folder_path)

df.columns = df.columns.str.strip()

df = df.drop(['Timestamp'], axis=1)

df = df.dropna()

df = df[df['Dst Port'] != 'Dst Port']

df = df.iloc[:400000]

df['Label'].unique()

from sklearn.model_selection import train_test_split

df_clustering, df_classification = train_test_split(df, test_size=0.85, random_state=42)

unique_value_counts = df_clustering.nunique()
columns_to_drop = unique_value_counts[unique_value_counts == 1].index
df_clustering = df_clustering.drop(columns=columns_to_drop)

df_clustering['Label'] = df_clustering['Label'].apply(lambda x: 0 if x == 'Benign' else 1)
```

```
import numpy as np
from sklearn.impute import KNNImputer
from sklearn.preprocessing import StandardScaler

features = df_clustering.drop(columns=['Label'])
true_labels = df_clustering['Label']

features.replace([np.inf, -np.inf], np.nan, inplace=True)
imputer = KNNImputer(n_neighbors=5)
features = imputer.fit_transform(features)

scaler = StandardScaler()
features_scaled = scaler.fit_transform(features)

print("Is NaN:")
print(np.isnan(features_scaled).sum())

from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

pca = PCA().fit(features_scaled)

explained_variance = np.cumsum(pca.explained_variance_ratio_)

plt.figure(figsize=(10, 6))
plt.plot(explained_variance, marker='o')
plt.xlabel('Number of Components')
plt.ylabel('Cumulative Explained Variance')
plt.title('Cumulative Explained Variance by PCA Components')
plt.grid()
```

```

plt.axhline(y=0.90, color='r', linestyle='--')
plt.axhline(y=0.95, color='g', linestyle='--')
plt.show()

n_components_90 = np.argmax(explained_variance >= 0.90) + 1
n_components_95 = np.argmax(explained_variance >= 0.95) + 1

n_components_90, n_components_95

pca = PCA(n_components=n_components_95)

features_pca = pca.fit_transform(features_scaled)

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score,
confusion_matrix
from sklearn.cluster import DBSCAN
from sklearn.neighbors import NearestNeighbors

k = n_components_95 + 1
nearest_neighbors = NearestNeighbors(n_neighbors=k)
neighbors = nearest_neighbors.fit(features_pca)
distances, indices = neighbors.kneighbors(features_pca)
distances = np.sort(distances[:, k-1], axis=0)

optimal_eps = distances.mean()
optimal_min_samples = k

dbscan = DBSCAN(eps=optimal_eps, min_samples=optimal_min_samples)
dbscan_labels = dbscan.fit_predict(features_pca)

```

```

dbscan_labels = np.where(dbscan_labels == -1, 0, 1)
df_clustering['DBSCAN'] = dbscan_labels

accuracy_dbscan = accuracy_score(true_labels, dbscan_labels)
precision_dbscan = precision_score(true_labels, dbscan_labels)
recall_dbscan = recall_score(true_labels, dbscan_labels)
f1_dbscan = f1_score(true_labels, dbscan_labels)
conf_matrix_dbscan = confusion_matrix(true_labels, dbscan_labels)

metrics_clustering = {
    'Accuracy': accuracy_dbscan,
    'Precision': precision_dbscan,
    'Recall': recall_dbscan,
    'F1-Score': f1_dbscan
}

print("DBSCAN")
print(f"Accuracy: {accuracy_dbscan}")
print(f"Precision: {precision_dbscan}")
print(f"Recall: {recall_dbscan}")
print(f"F1-Score: {f1_dbscan}")
print(f"Confusion Matrix:\n {conf_matrix_dbscan}")

import seaborn as sns

fig, ax = plt.subplots(figsize=(8, 6))
metrics_names = list(metrics_clustering.keys())
metrics_values = list(metrics_clustering.values())
colors = ["#985e6d", "#49648b", "#192231", "#2a8973"]

```

```

ax.bar(metrics_names, metrics_values, color=colors)
ax.set_ylim(0, 1)
ax.set_ylabel('Metric Value')
ax.set_title('DBSCAN Clusterization')

plt.tight_layout()
plt.show()

labels = ['Label', 'DBSCAN']

counts = {label: df_clustering[label].value_counts() for label in labels}
percentages = {label: df_clustering[label].value_counts(normalize=True) * 100 for label
in labels}

for label in labels:
    print("Counts for", counts[label])
    print("Percentage for", percentages[label], '\n')

fig, ax = plt.subplots(2, 1, figsize=(8, 6))

bar_width = 0.2
index = np.arange(len(labels))
colors = ['#57b884', '#f0e594']

for i, label in enumerate(labels):
    counts_0 = counts[label].get(0, 0)
    counts_1 = counts[label].get(1, 0)
    ax[0].bar(i - bar_width/2, counts_0, bar_width, color=colors[0], label='0 — Normal'
if i == 0 else '')

```

```
ax[0].bar(i + bar_width/2, counts_1, bar_width, color=colors[1], label='1 —
Anomaly' if i == 0 else '')
```

```
ax[0].set_ylabel('Counts')
ax[0].set_xticks(index)
ax[0].set_xticklabels(labels)
ax[0].legend()
```

```
for i, label in enumerate(labels):
```

```
    percentages_0 = percentages[label].get(0, 0)
    percentages_1 = percentages[label].get(1, 0)
    ax[1].bar(i - bar_width/2, percentages_0, bar_width, color=colors[0], label='0 —
Normal' if i == 0 else '')
    ax[1].bar(i + bar_width/2, percentages_1, bar_width, color=colors[1], label='1 —
Anomaly' if i == 0 else '')
```

```
ax[1].set_ylabel('Percentages')
ax[1].set_xticks(index)
ax[1].set_xticklabels(labels)
ax[1].legend()
```

```
plt.tight_layout()
plt.show()
```

```
unique_value_counts = df_classification.nunique()
columns_to_drop = unique_value_counts[unique_value_counts == 1].index
df_classification = df_classification.drop(columns=columns_to_drop)
```

```
df_classification['Label'] = df_classification['Label'].apply(lambda x: 0 if x == 'Benign'
else 1)
```

```

X_train = features_pca
y_train = df_clustering['DBSCAN']

X_test = df_classification.drop(columns=['Label'])

X_test.replace([np.inf, -np.inf], np.nan, inplace=True)
X_test = imputer.fit_transform(X_test)

X_test = scaler.fit_transform(X_test)

X_test = pca.fit_transform(X_test)

y_test = df_classification['Label']

from sklearn.naive_bayes import GaussianNB

nb_model = GaussianNB()
nb_model.fit(X_train, y_train)
y_pred = nb_model.predict(X_test)

df_classification['DBSCAN'] = y_pred

accuracy_nb = accuracy_score(y_test, y_pred)
precision_nb = precision_score(y_test, y_pred)
recall_nb = recall_score(y_test, y_pred)
f1_nb = f1_score(y_test, y_pred)
conf_matrix_nb = confusion_matrix(y_test, y_pred)

metrics_classification = {

```

```

    'Accuracy': accuracy_nb,
    'Precision': precision_nb,
    'Recall': recall_nb,
    'F1-Score': f1_nb
}

print(f"Naive Bayes")
print(f"Accuracy: {accuracy_nb}")
print(f"Precision: {precision_nb}")
print(f"Recall: {recall_nb}")
print(f"F1-Score: {f1_nb}")
print(f"Confusion Matrix:\n {conf_matrix_nb}\n")

counts = {label: df_classification[label].value_counts() for label in labels}
percentages = {label: df_classification[label].value_counts(normalize=True) * 100 for
label in labels}

for label in labels:
    print("Counts for", counts[label])
    print("Percentage for", percentages[label], '\n')

fig, ax = plt.subplots(2, 1, figsize=(8, 6))

bar_width = 0.2
index = np.arange(len(labels))
colors = ['#15b2d3', '#e88d67']

for i, label in enumerate(labels):
    counts_0 = counts[label].get(0, 0)
    counts_1 = counts[label].get(1, 0)

```

```
ax[0].bar(i - bar_width/2, counts_0, bar_width, color=colors[0], label='0 — Normal'
if i == 0 else "")
```

```
ax[0].bar(i + bar_width/2, counts_1, bar_width, color=colors[1], label='1 —
Anomaly' if i == 0 else "")
```

```
ax[0].set_ylabel('Counts')
```

```
ax[0].set_xticks(index)
```

```
ax[0].set_xticklabels(labels)
```

```
ax[0].legend()
```

```
for i, label in enumerate(labels):
```

```
    percentages_0 = percentages[label].get(0, 0)
```

```
    percentages_1 = percentages[label].get(1, 0)
```

```
ax[1].bar(i - bar_width/2, percentages_0, bar_width, color=colors[0], label='0 —
Normal' if i == 0 else "")
```

```
ax[1].bar(i + bar_width/2, percentages_1, bar_width, color=colors[1], label='1 —
Anomaly' if i == 0 else "")
```

```
ax[1].set_ylabel('Percentages')
```

```
ax[1].set_xticks(index)
```

```
ax[1].set_xticklabels(labels)
```

```
ax[1].legend()
```

```
plt.tight_layout()
```

```
plt.show()
```

```
fig, ax = plt.subplots(figsize=(8, 6))
```

```
metrics_names = list(metrics_classification.keys())
```

```
metrics_values = list(metrics_classification.values())
```

```
colors = ['#6fa8dc', '#b4a7d6', '#cccccc', '#916363']
```

```
ax.bar(metrics_names, metrics_values, color=colors)
ax.set_ylim(0, 1)
ax.set_ylabel('Metric Value')
ax.set_title('Naive Bayes Based on DBSCAN Classification')

plt.tight_layout()
plt.show()
```