МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КІБЕРБЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ
КАФЕДРА УПРАВЛІННЯ КІБЕРБЕЗПЕКОЮ ТА ЗАХИСТОМ ІНФОРМАЦІЇ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: “СИСТЕМИ ВИЯВЛЕННЯ ТА ПОПЕРЕДЖЕННЯ ВТОРГНЕНЬ (IDS/IPS)”

на здобуття освітнього ступеня бакалавра
зі спеціальності 125 Кібербезпека
освітньої програми Управління інформаційною та кібернетичною безпекою

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Андрій БАБЕНКО
(підпис) *Ім’я, ПРІЗВИЩЕ здобувача*

Виконав: здобувач вищої освіти гр. УБД-42

Андрій БАБЕНКО
Ім’я, ПРІЗВИЩЕ

Керівник: Світлана ЛЕГОМІНОВА
Д.е.н., професор Ім’я, ПРІЗВИЩЕ

Рецензент: Галина ГАЙДУР
Д.т.н., професор Ім’я, ПРІЗВИЩЕ

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КІБЕРБЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ

Кафедра Управління кібербезпекою та захистом інформації

Ступінь вищої освіти бакалавр

Спеціальність 125 Кібербезпека

Освітня програма Управління інформаційною та кібернетичною безпекою

ЗАТВЕРДЖУЮ
Завідувач кафедри УКБЗІ
_____ Світлана ЛЕГОМІНОВА
“____”_____2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Бабенку Андрію Віталійовичу
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи “Системи виявлення та попередження вторгнень (IDS/IPS)”, керівник кваліфікаційної роботи ЛЕГОМІНОВА Світлана, д.е.н., професор,
(ПРІЗВИЩЕ, Ім'я,, науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від “24“ лютого 2025 р. №56.

2. Строк подання кваліфікаційної роботи “12” травня 2025р.

3. Вихідні дані до кваліфікаційної роботи: *система, виявлення та попередження вторгнень, IDS, IPS, міжнародні стандарти, наукова та технічна література.*

4. Перелік питань, які мають бути розроблені:
4.1. Проаналізувати сучасні підходи до систем виявлення та попередження вторгнень (IDS/IPS)
4.2. Дослідити методи виявлення та попередження вторгнень (IDS/IPS)
4.3. Визначено ключові аспекти побудови IDS/IPS-систем, обґрунтовано вибір методу машинного навчання, розроблено програмну систему виявлення та попередження вторгнень і проведено її тестування.

5. Перелік ілюстративного матеріалу: *презентація PowerPoint*

6. Дата видачі завдання “10“ березня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Етапи кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1.	Визначення об’єкту, предмету, мети та завдань дослідження.	18.03.2025	виконано
2.	Збір та аналіз літератури.	29.03.2025	виконано
3.	Аналіз сучасних підходів до систем виявлення та попередження вторгнень	08.04.2025	виконано
4.	Методи виявлення та попередження вторгненням	15.04.2025	виконано
5.	Розробка та тестування системи виявлення та попередження вторгнень (IDS/IPS)	22.04.2025	виконано
6.	Формулювання висновків за результатами проведеного дослідження	29.04.2025	виконано
7.	Оформлення роботи.	06.05.2025	виконано
8.	Оформлення презентації.	09.05.2025	виконано
9.	Отримання рецензії на роботу.	__.06.2025	
10.	Захист в ЕК.	__.06.2025	

Здобувач вищої освіти _____ Андрій БАБЕНКО
(підпис) *(Ім’я, ПРІЗВИЩЕ)*

Керівник
кваліфікаційної роботи _____ Світлана ЛЕГОМІНОВА
(підпис) *(Ім’я, ПРІЗВИЩЕ)*

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КІБЕРБЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ

ПОДАННЯ
ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ
ЩОДО ЗАХИСТУ КВАЛІФІКАЦІЙНОЇ РОБОТИ
на здобуття освітнього ступеня бакалавра

Направляється здобувач Бабенко Андрій Віталійович до захисту кваліфікаційної роботи
(прізвище та ініціали)
за спеціальністю 125 Кібербезпека
(код, найменування спеціальності)
освітньої програми Управління інформаційною та кібернетичною безпекою
(назва)
на тему: “ Системи виявлення та попередження вторгнень (IDS/IPS)”

Кваліфікаційна робота і рецензія додаються.

Директор ННІКБЗІ _____ Євгенія ІВАНЧЕНКО
(підпис) *(Ім’я, ПРІЗВИЩЕ)*

Висновок керівника кваліфікаційної роботи

Здобувач БАБЕНКО Андрій у кваліфікаційній роботі проаналізував сучасні підходи до систем виявлення та попередження вторгнень (IDS/IPS), визначив ключові методи виявлення та попередження вторгнень (IDS/IPS), запропонував рекомендації щодо формування системи виявлення та попередження вторгнень (IDS/IPS). БАБЕНКО Андрій показав розуміння проблеми дослідження та бачення основних теоретичних і практичних напрямів її вирішення, довів володіння методами наукового дослідження, проявила себе як організований, відповідальний виконавець. Результати дослідження апробовані на конференції.

Все це дозволяє оцінити кваліфікаційну роботу здобувача БАБЕНКА Андрія на оцінку “відмінно” та присвоїти їй кваліфікацію бакалавра з кібербезпеки за освітньою програмою Управління інформаційною та кібернетичною безпекою.

Керівник кваліфікаційної роботи _____ Світлана ЛЕГОМІНОВА
(підпис) *(Ім’я, ПРІЗВИЩЕ)*

“____“_____2025 року

Висновок кафедри про кваліфікаційну роботу

Кваліфікаційна робота розглянута. Здобувач Бабенко А.В. допускається до захисту даної роботи в Екзаменаційній комісії.

Завідувач кафедри
управління кібербезпекою та
захистом інформації _____ Світлана ЛЕГОМІНОВА
(підпис) *(Ім’я, ПРІЗВИЩЕ)*

ВІДГУК РЕЦЕНЗЕНТА
на кваліфікаційну бакалаврську роботу

здобувача вищої освіти БАБЕНКА Андрія
на тему “Системи виявлення та попередження вторгнень (IDS/IPS)”

Актуальність. У сучасних умовах зростання кількості кібератак та ускладнення методів несанкціонованого доступу до інформаційних ресурсів, особливої ваги набувають системи виявлення та попередження вторгнень (IDS/IPS), що дозволяють оперативно ідентифікувати загрози на основі аналізу мережевого трафіку. Використання методів машинного навчання в таких системах забезпечує підвищення точності виявлення, зменшення кількості хибних спрацювань і здатність адаптуватися до нових, ще не ідентифікованих типів атак. У цьому контексті поєднання інтелектуального аналізу та системної обробки даних стає основою ефективного захисту цифрової інфраструктури.

З огляду на зазначене, дослідження методів і засобів виявлення та попередження вторгнень із використанням технологій машинного навчання є актуальним науковим і прикладним завданням.

Позитивні сторони.

1. У роботі комплексно досліджено підходи до побудови IDS/IPS-систем, класифіковано методи виявлення загроз і проаналізовано їх ефективність.
2. Реалізовано повнофункціональну програмну систему з інтеграцією моделі машинного навчання, проведено тестування на реальних даних.
3. Структура роботи є чіткою та логічною, усі положення підтверджено графічними матеріалами, таблицями та прикладами роботи програми.
4. Оформлення кваліфікаційної роботи відповідає встановленим вимогам, використано достатню кількість джерел, включно з англомовними.

Недоліки.

Доцільним було б надати ширше порівняння обраного алгоритму з іншими методами в умовах потокового аналізу трафіку. Разом з тим, це не знижує загальної наукової та практичної цінності проведеного дослідження.

Однак, вищезгадані зауваження не впливають на загальну позитивну оцінку кваліфікаційної роботи.

Висновок: Кваліфікаційна робота виконана на високому науково-практичному рівні, має завершений характер і заслуговує на оцінку “відмінно”, а здобувач заслуговує присвоєння кваліфікації бакалавра з кібербезпеки за освітньою програмою Інформаційна та кібернетична безпека.

Рецензент: _____ Галина ГАЙДУР
д.т.н., професор *підпис* Ім’я, ПРІЗВИЩЕ

РЕФЕРАТ

Кваліфікаційна робота присвячена розробці системи виявлення та попередження вторгнень у комп'ютерних мережах на основі технологій машинного навчання. Робота включає вступ, три розділи, що містять 36 рисунків і 11 таблиць, а також висновки й список використаних джерел з 50 найменувань. Загальний обсяг роботи становить 78 аркушів, з яких 6 аркушів займають додатки та список використаних джерел.

Метою роботи є розроблення ефективної IDS/IPS-системи з використанням методів машинного навчання, орієнтованої на своєчасне виявлення аномальної мережевої активності.

Об'єктом дослідження є процеси забезпечення кібербезпеки комп'ютерних мереж.

Предмет дослідження – методи та засоби автоматизованого виявлення і запобігання вторгненням на основі аналізу поведінки трафіку.

Методи дослідження. У роботі використано методи аналізу та синтезу наукових джерел, порівняння підходів до побудови IDS/IPS-систем, моделювання мережевої активності та експериментального тестування ефективності виявлення вторгнень.

У ході дослідження проаналізовано сучасні архітектури IDS/IPS, принципи моніторингу, методи обробки поведінкових ознак, обґрунтовано вибір методу швидкого дерева, реалізовано систему виявлення вторгнень, що пройшла успішне тестування в експериментальному середовищі.

Галузь застосування. Результати дослідження можуть бути використані в корпоративних мережах, освітніх установах та критичних інфраструктурах для посилення кіберзахисту шляхом інтеграції інтелектуальних систем контролю мережевого трафіку.

Ключові слова: ВИЯВЛЕННЯ ВТОРГНЕНЬ, IDS, IPS, АНАЛІЗ ПОВЕДІНКИ, МАШИННЕ НАВЧАННЯ, FASTTREE, МЕРЕЖЕВА БЕЗПЕКА.

ABSTRACT

The qualification work is devoted to the development of a system for detecting and preventing intrusions in computer networks based on machine learning technologies. The work includes an introduction, three sections containing 36 figures and 11 tables, as well as conclusions and a list of sources used from 50 items. The total volume of the work is 78 sheets, of which 6 sheets are occupied by appendices and a list of sources used.

The goal of the work is to develop an effective IDS/IPS system using machine learning methods, focused on the timely detection of anomalous network activity.

The object the study is the processes of ensuring cybersecurity of computer networks.

The subject of the study is methods and means of automated detection and prevention of intrusions based on traffic behavior analysis.

Research methods. The work uses methods of analysis and synthesis of scientific sources, comparison of approaches to building IDS/IPS systems, modeling of network activity, and experimental testing of intrusion detection efficiency.

The study analyzed modern IDS/IPS architectures, monitoring principles, behavioral feature processing methods, justified the choice of the fast tree method, and implemented an intrusion detection system that was successfully tested in an experimental environment.

Field of application. The research results can be used in corporate networks, educational institutions, and critical infrastructures to strengthen cyber defense by integrating intelligent network traffic control systems.

Keywords: INTRUSION DETECTION, IDS, IPS, BEHAVIOR ANALYSIS, MACHINE LEARNING, FASTTREE, NETWORK SECURITY.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ 9
ВСТУП 10
РОЗДІЛ 1 АНАЛІЗ СУЧАСНИХ ПІДХОДІВ ДО СИСТЕМ ВИЯВЛЕННЯ ТА ПОПЕРЕДЖЕННЯ ВТОРГНЕНЬ (IDS/IPS) 12
1.1 Сутність та класифікація систем виявлення та запобігання вторгненням . 12
1.2 Ключові принципи роботи IDS та IPS систем 18
1.3 Компаративний аналіз IDS та IPS в відповідь на новітні загрози та технологічні рішення 22
Висновки до розділу 1 25
РОЗДІЛ 2 МЕТОДИ ВИЯВЛЕННЯ ТА ПОПЕРЕДЖЕННЯ ВТОРГНЕННЯМ (IDS/IPS) 27
2.1 Аналіз сигнатурних та аномальних методів виявлення вторгнень 27
2.2 Вибір методу машинного навчання для виявлення вторгнення 30
2.3 Обґрунтування вибору тренувального набору даних 37
Висновки до розділу 2 46
РОЗДІЛ 3 РОЗРОБКА ТА ТЕСТУВАННЯ СИСТЕМИ ВИЯВЛЕННЯ ТА ПОПЕРЕДЖЕННЯ ВТОРГНЕНЬ (IDS/IPS) 48
3.1 Проектування системи виявлення та попередження вторгнень 48
3.2 Завантаження та обробка датасету для тренування моделі 54
3.3 Реалізація системи виявлення та попередження вторгнень на базі обраного методу 55
3.4 Тестування системи та її оцінка 63
Висновки до розділу 3 69
ВИСНОВКИ 71
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 73

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

НБК – наївний байєсівський класифікатор;

ОВ – опорний вектор, метод опорних векторів;

ШД – швидке дерево, метод машинного навчання для задач класифікації;

C# – мова програмування, що використовується для реалізації програмної частини системи;

CTI – Cyber Threat Intelligence, розвідка кіберзагроз;

HIDS – Host-based Intrusion Detection System, хостова система виявлення вторгнень;

HIPS – Host-based Intrusion Prevention System, хостова система попередження вторгнень;

IDS – Intrusion Detection System, система виявлення вторгнень;

IPS – Intrusion Prevention System, система попередження вторгнень;

ML.NET – програмна платформа для реалізації моделей машинного навчання в середовищі .NET;

NIDS – Network-based Intrusion Detection System, мережева система виявлення вторгнень;

NIPS – Network-based Intrusion Prevention System, мережева система попередження вторгнень;

SIEM – Security Information and Event Management, управління інформацією та подіями безпеки;

SOAR – Security Orchestration, Automation and Response, оркестрація, автоматизація й реагування на інциденти безпеки.

ВСТУП

Актуальність теми. У сучасному світі інформаційні системи стали основою функціонування критичної інфраструктури, державного управління, бізнесу та повсякденного життя. Проте стрімкий розвиток інформаційних технологій супроводжується постійним зростанням числа кіберзагроз та складністю атак на інформаційні ресурси [1]. Вторгнення у комп'ютерні мережі, спрямовані на крадіжку даних, порушення цілісності систем або виведення їх з ладу, стали буденністю для багатьох організацій.

У цьому контексті своєчасне виявлення та запобігання несанкціонованим діям набуває критичного значення. Системи виявлення та попередження вторгнень (IDS/IPS) є одними з ключових елементів багаторівневої архітектури кіберзахисту, дозволяючи ідентифікувати аномальну активність у мережі, виявляти спроби злому, а також своєчасно реагувати на інциденти безпеки [2]. Розвиток IDS/IPS-систем безпосередньо пов'язаний із впровадженням інтелектуальних методів аналізу даних, що використовують алгоритми машинного навчання для підвищення точності виявлення атак та зниження кількості хибних спрацьовувань.

З огляду на постійне удосконалення методів атак, що часто використовують новітні способи обходу традиційних засобів захисту, розроблення, дослідження та вдосконалення систем виявлення та попередження вторгнень є надзвичайно актуальним науковим та практичним завданням.

Об’єктом дослідження є процеси забезпечення кібербезпеки комп’ютерних мереж.

Предметом дослідження є методи та засоби автоматизованого виявлення і запобігання вторгненням на основі аналізу поведінки трафіку.

Мета роботи полягає у розробленні ефективної IDS/IPS-системи з використанням методів машинного навчання, орієнтованої на своєчасне виявлення аномальної мережевої активності.

Для досягнення цієї мети в роботі необхідно виконати наступні завдання:

– проаналізувати існуючі підходи до побудови систем виявлення та запобігання вторгненням, їх класифікацію та ключові принципи роботи;

– дослідити методи виявлення вторгнень, обґрунтувати вибір методу машинного навчання та відповідного тренувального набору даних;

– реалізувати програмну систему виявлення та попередження вторгнень на базі обраної моделі машинного навчання, забезпечивши її тестування та оцінку ефективності.

Методи дослідження. Для вирішення означеного вище наукового завдання в роботі використані методи аналізу та синтезу наукових джерел, порівняння існуючих підходів до побудови IDS/IPS-систем, класифікації методів виявлення аномалій, моделювання мережевої активності, експериментального тестування ефективності систем виявлення вторгнень, а також метод системного підходу до забезпечення інформаційної безпеки комп'ютерних мереж.

Практичне значення одержаних результатів. Результати роботи можуть бути використані для розроблення і впровадження ефективних систем виявлення та попередження вторгнень у корпоративних та публічних мережах. Запропоновані рішення дозволяють здійснити обґрунтований вибір методів машинного навчання для підвищення точності і швидкодії IDS/IPS-систем, сприяючи підвищенню рівня захисту інформаційних ресурсів від несанкціонованого доступу та кібератак.

Наукова новизна одержаних результатів. Наукова новизна роботи полягає в удосконаленні підходів до побудови систем виявлення та попередження вторгнень шляхом інтеграції методів машинного навчання для автоматичного виявлення аномальної мережевої активності з підвищеною точністю та швидкодією.

РОЗДІЛ 1 АНАЛІЗ СУЧАСНИХ ПІДХОДІВ ДО СИСТЕМ ВИЯВЛЕННЯ ТА ПОПЕРЕДЖЕННЯ ВТОРГНЕНЬ (IDS/IPS)

1.1 Сутність та класифікація систем виявлення та запобігання вторгненням

Інформаційна безпека у сучасних комп'ютерних мережах неможлива без ефективного моніторингу мережевого трафіку, своєчасного виявлення спроб несанкціонованого доступу та запобігання атак. Враховуючи постійне зростання кількості та складності кіберзагроз, системи виявлення та запобігання вторгненням стали невід'ємним елементом архітектури кіберзахисту інформаційних систем.

Система виявлення вторгнень (IDS, Intrusion Detection System) – це програмно-апаратний комплекс, основною функцією якого є моніторинг та аналіз подій у комп'ютерній мережі або окремих системах з метою виявлення ознак потенційних або фактичних атак [3]. IDS здійснює пасивне спостереження за трафіком чи поведінкою системи, генерує повідомлення про підозрілу активність, проте не перешкоджає її розвитку.

Система запобігання вторгнень (IPS, Intrusion Prevention System) є подальшим розвитком концепції IDS. Окрім функцій виявлення, IPS здатна автоматично блокувати шкідливий трафік або запобігати виконанню шкідливих операцій у режимі реального часу [4]. Це дозволяє не лише виявляти, а й активно реагувати на загрози без необхідності втручання людини.

Одним із ключових аспектів побудови систем IDS/IPS є визначення джерела даних, що використовується для аналізу та виявлення загроз. Джерело даних визначає характер інформації, яка підлягає обробці, методи її аналізу та можливості системи в контексті своєчасного виявлення аномальної або шкідливої активності [5]. Для розроблення високоефективної IDS/IPS важливо правильно обрати тип даних залежно від специфіки середовища, яке захищається, і очікуваних типів атак. Використання різних джерел дозволяє системам детальніше відслідковувати поведінку мережі або окремих пристроїв,

підвищуючи тим самим точність виявлення загроз. У межах розробки сучасних рішень на базі машинного навчання саме якість і релевантність даних визначає успіх побудови моделі прогнозування [6]. Тому розуміння класифікації IDS/IPS за джерелом даних є необхідною основою для створення ефективних систем мережевої безпеки.

Для наочного уявлення класифікації систем IDS/IPS за джерелом даних на рис. 1.1 подано діаграму, що відображає основні напрямки цієї класифікації та їх особливості.

Рис. 1.1. Класифікація IDS/IPS за джерелом даних

Як видно з діаграми на рис. 1.1, класифікація IDS/IPS за джерелом даних поділяється на два основні напрями: мережеві IDS/IPS (NIDS/NIPS) та хостові IDS/IPS (HIDS/HIPS). Мережеві IDS/IPS аналізують мережевий трафік, орієнтуючись на виявлення аномалій у мережі та пошук атак на рівні окремих мережевих пакетів. Вони дозволяють ідентифікувати зовнішні загрози ще на стадії їхнього розповсюдження у мережевому середовищі. Хостові IDS/IPS, у свою чергу, здійснюють моніторинг активності конкретних пристроїв, аналізуючи журнали подій, системні виклики та контролюючи зміни у файловій системі. Такий підхід дозволяє виявити підозрілу активність безпосередньо на

кінцевих точках, що є критично важливим для запобігання локальним загрозам та збереження цілісності даних.

Ефективність роботи систем виявлення та запобігання вторгненням значною мірою залежить від обраного способу виявлення загроз. Саме методика виявлення визначає здатність системи оперативно реагувати на спроби атак та мінімізувати ризики для інформаційної безпеки мережі [7]. У практиці розроблення IDS/IPS-рішень питання способу виявлення є центральним, оскільки правильний вибір дозволяє або швидко виявляти вже відомі типи атак, або забезпечити виявлення нових, раніше невідомих загроз. З огляду на постійне ускладнення способів атак, сучасні системи повинні адаптуватися до змін у поведінці зловмисників, використовуючи як традиційні підходи, так і інноваційні методи на основі аналізу поведінки.

З метою систематизації основних підходів до класифікації IDS/IPS за способом виявлення загроз на рис. 1.2 подано діаграму послідовності, яка ілюструє основні напрями такого поділу.

Рис. 1.2. Класифікація IDS/IPS за способом виявлення загроз

Як показано на рис. 1.2, класифікація систем IDS/IPS за способом виявлення загроз поділяється на сигнатурні та аномальні системи. Сигнатурні системи базуються на виявленні загроз шляхом порівняння вхідних даних із попередньо створеною базою сигнатур відомих атак. Вони є високоефективними для швидкої ідентифікації вже задокументованих загроз, однак демонструють слабкість перед новими та невідомими видами атак, оскільки не мають відповідних записів у базі сигнатур. Аномальні системи працюють за іншим принципом: вони формують профіль нормальної поведінки мережі або хоста та виявляють відхилення від цього профілю [8]. Такий підхід дозволяє знаходити нові або модифіковані атаки, однак має недолік у вигляді підвищеного рівня хибних спрацьовувань, оскільки будь-яке нетипове, але безпечне відхилення також може сприйматися як загроза.

Отже, спосіб виявлення загроз у системах IDS/IPS визначає їхню здатність балансувати між точністю виявлення та чутливістю до змін у середовищі. Комбінація сигнатурного та аномального підходів дозволяє досягати найкращих результатів у складних умовах сучасних мереж. Особливо важливим є правильний вибір підходу на етапі розроблення систем, орієнтованих на виявлення невідомих атак за допомогою методів машинного навчання.

Вибір способу інтеграції системи в мережеве середовище визначає її можливості у виявленні та реагуванні на загрози. Розташування системи відносно мережевого потоку безпосередньо впливає на її функціональні характеристики: чи буде вона лише здійснювати спостереження, чи зможе активно втручатися у мережевий трафік для запобігання атакам [9]. У сучасних умовах побудови IDS/IPS-рішень правильне визначення способу розгортання є критичним для балансування між продуктивністю мережі та рівнем безпеки. Відповідно до обраної стратегії система може бути спрямована на моніторинг активності або на активне блокування загроз у режимі реального часу.

У рамках дослідження здійснено класифікацію систем IDS/IPS за способом розгортання, що представлено на діаграмі послідовності на рис. 1.3.

Рис. 1.3. Класифікація IDS/IPS за способом розгортання

На рис. 1.3 показано, що класифікація за способом розгортання передбачає два основні підходи: пасивні системи та інлайн-системи. Пасивні системи, характерні переважно для IDS, працюють шляхом спостереження за мережевим трафіком без прямого втручання у нього. Вони призначені для виявлення підозрілої активності та створення повідомлень про інциденти без впливу на потік даних. Інлайн-системи, які є типовими для IPS, встановлюються безпосередньо на шляху проходження мережевого трафіку, що дозволяє їм у режимі реального часу блокувати шкідливу активність та запобігати розвитку атак [10]. Такий підхід забезпечує проактивний захист, але вимагає ретельної оптимізації, щоб уникнути затримок у передаванні даних.

При розробленні систем виявлення та запобігання вторгненням важливе значення має рівень їх інтеграції у загальну інформаційну інфраструктуру. Від обраного підходу до інтеграції залежить як ефективність реагування на загрози, так і гнучкість системи, її вартість, продуктивність та зручність розгортання. У контексті сучасних IDS/IPS особливої актуальності набуває вибір між

окремими спеціалізованими пристроями і програмними рішеннями, що інтегруються у вже наявне обладнання чи сервери. Це рішення повинно враховувати особливості середовища, критичність об'єктів захисту та бюджетні обмеження організації. При розробленні систем на основі машинного навчання правильний вибір моделі інтеграції безпосередньо впливає на швидкість обробки даних, надійність системи та можливості її масштабування. Тому розгляд класифікації IDS/IPS за рівнем інтеграції є важливою складовою побудови ефективних рішень у сфері кібербезпеки.

У рамках дослідження здійснено класифікацію систем IDS/IPS за рівнем інтеграції, яка відображена на рис. 1.4.

Рис. 1.4. Класифікація IDS/IPS за рівнем інтеграції

Як показано на рис. 1.4, класифікація IDS/IPS за рівнем інтеграції поділяється на окремі пристрої (appliance) та програмні рішення. Окремі пристрої є спеціалізованими апаратними засобами, які забезпечують виявлення та запобігання вторгненням із високим рівнем продуктивності та простотою налаштування, однак характеризуються вищою вартістю. Такі пристрої здатні працювати автономно і не залежать від ресурсів основної інфраструктури. Програмні рішення, у свою чергу, інтегруються у вже наявні системи,

використовуючи ресурси хост-обладнання. Вони забезпечують гнучкість впровадження, нижчу вартість, проте їх ефективність може залежати від характеристик існуючих серверів чи робочих станцій.

Сучасні тенденції розвитку IDS/IPS-систем полягають у гібридизації підходів, що поєднують сигнатурне та аномальне виявлення, застосуванні методів машинного навчання для зниження кількості хибних спрацьовувань, а також інтеграції з іншими компонентами систем кіберзахисту в рамках концепцій на кшталт SIEM (Security Information and Event Management) та SOAR (Security Orchestration, Automation and Response) [11].

Таким чином, системи виявлення та запобігання вторгненням є базовими інструментами захисту інформаційних систем, що забезпечують можливість своєчасного реагування на сучасні кіберзагрози та зменшення ризику компрометації інформаційних ресурсів.

1.2 Ключові принципи роботи IDS та IPS систем

Системи IDS/IPS є невід'ємною складовою сучасних комплексів захисту інформаційних систем. Їхня основна функція полягає у моніторингу активності в мережах або на окремих хостах для своєчасного виявлення аномальної або шкідливої поведінки та реалізації заходів з попередження загроз. Робота таких систем ґрунтується на певних ключових принципах, що визначають їхню ефективність, гнучкість та здатність протидіяти різноманітним типам атак.

Безперервний моніторинг активності є одним із фундаментальних принципів роботи IDS/IPS. Під цим поняттям розуміють постійний процес спостереження за мережевим трафіком або поведінкою систем для фіксації змін, що можуть свідчити про наявність шкідливої або нетипової активності [12]. Такий моніторинг дозволяє виявляти ознаки атак на ранніх стадіях їхнього розвитку та мінімізувати потенційні наслідки для інформаційної інфраструктури. У межах сучасних систем IDS/IPS для здійснення безперервного моніторингу використовуються різноманітні методи: аналіз потоків даних, контроль системних журналів, обробка мережевих пакетів у

реальному часі та профілювання поведінки об'єктів. Кожен з цих методів має свою специфіку застосування залежно від типу системи (мережева або хостова) та поставлених завдань безпеки. Ретельний вибір та комбінація методів моніторингу дозволяють сформувати багаторівневу модель захисту, орієнтовану на виявлення як відомих, так і нових загроз.

Для систематизації основних підходів, які використовуються у межах безперервного моніторингу активності в IDS/IPS, у таблиці 1.1 наведено їх узагальнене порівняння.

Таблиця 1.1.

Основні підходи до безперервного моніторингу активності в IDS/IPS

Метод моніторингу	Опис методу	Сфера застосування	Приклади використання
Аналіз мережевих потоків	Збір метаданих про з'єднання без аналізу вмісту пакетів	Мережеві IDS/IPS (NIDS/NIPS)	NetFlow, IPFIX
Обробка мережевих пакетів	Повний аналіз вмісту пакетів даних у реальному часі	Мережеві IDS/IPS	Snort, Suricata
Моніторинг системних журналів	Аналіз записів системних подій, змін доступу, авторизацій тощо	Хостові IDS/IPS (HIDS/HIPS)	OSSEC, Wazuh
Контроль системних викликів	Спостереження за діями процесів на рівні ядра операційної системи	Хостові IDS/IPS	Auditd, Sysmon
Профілювання поведінки	Створення моделей нормальної поведінки об'єктів і виявлення відхилень	Аномальні IDS/IPS	Machine Learning IDS, AI-based IDS

Безперервний моніторинг активності забезпечує основу для своєчасного виявлення порушень безпеки та реагування на загрози. Комбінування різних методів моніторингу дозволяє системам IDS/IPS охоплювати різні рівні інфраструктури – від мережевого трафіку до поведінки окремих хостів. У подальших етапах побудови систем безпеки важливим є правильний вибір методів моніторингу відповідно до особливостей середовища, що захищається.

Аналіз поведінки є одним із ключових механізмів виявлення загроз у сучасних системах виявлення та запобігання вторгненням. Суть цього підходу полягає у побудові профілю нормальної поведінки користувачів, пристроїв або

мережевих об'єктів та подальшому виявленні відхилень від цього профілю [13]. На відміну від класичних сигнатурних методів, аналіз поведінки не вимагає наявності попередньо відомих шаблонів атак і дозволяє виявляти невідомі або модифіковані загрози на основі змін у діях об'єктів. Для реалізації аналізу поведінки використовуються як традиційні методи статистичного аналізу, так і сучасні алгоритми машинного навчання, що дозволяють адаптивно вивчати нормальну активність у мережі чи на хостах [14]. Виявлені відхилення оцінюються за різними критеріями, такими як частота подій, обсяг переданих даних, типи взаємодій або часові закономірності. У контексті побудови інтелектуальних IDS/IPS систем аналіз поведінки є критичним етапом для створення моделей, здатних ідентифікувати аномалії без прив'язки до бази відомих загроз.

Для більшої наочності основні підходи, що застосовуються в рамках аналізу поведінки, наведено в табл. 1.2.

Таблиця 1.2.

Основні підходи до аналізу поведінки в системах IDS/IPS

Підхід до аналізу	Суть підходу	Сфера застосування	Приклади реалізації
Статистичний аналіз	Визначення нормальних параметрів активності та фіксація відхилень	Мережеві та хостові IDS/IPS	Моніторинг обсягів трафіку, кількості запитів
Профілювання користувачів	Формування моделей типової поведінки окремих облікових записів	Хостові IDS/IPS, SIEM-системи	Аналіз звичної активності користувачів у мережі
Машинне навчання (класифікація)	Навчання моделей на основі нормальних даних для подальшого виявлення аномалій	Аномальні IDS/IPS	Використання класифікаторів, наприклад, Random Forest, SVM
Машинне навчання (кластеризація)	Автоматичне групування поведінкових даних без попередніх міток	Аномальні IDS/IPS	Алгоритми K-Means, DBSCAN
Аналіз часових закономірностей	Виявлення змін у поведінці залежно від часу доби чи періоду	Мережеві та хостові IDS/IPS	Виявлення нетипової активності у нічний час

Аналіз поведінки дозволяє системам виявлення та запобігання вторгненням працювати у динамічних середовищах, де класичні сигнатурні підходи можуть бути недостатніми. Важливою складовою цього процесу є правильний вибір методів побудови моделей поведінки, що відповідають особливостям захищуваного середовища [15]. Розвиток методів машинного навчання відкриває нові можливості для підвищення адаптивності та гнучкості систем IDS/IPS при виявленні загроз.

Реагування на виявлені інциденти є обов'язковим компонентом функціонування IDS/IPS і охоплює дії, що виконуються після фіксації підозрілої чи шкідливої активності з метою запобігання розвитку атаки, мінімізації наслідків та відновлення нормальної роботи системи [16]. У IDS основною функцією є створення сповіщень або подій для операторів безпеки, тоді як IPS реалізує активне втручання: блокування трафіку, завершення сесій чи застосування політик у реальному часі. Методи реагування варіюються від пасивного інформування до активних заходів впливу на мережеве середовище [17], що дозволяє зменшити ризик ескалації інцидентів і забезпечити контроль над загрозами.

Для систематизації основних способів реагування в IDS/IPS системах у табл. 1.3 наведено порівняльну характеристику відповідних підходів.

Таблиця 1.3.

Основні способи реагування на виявлені інциденти в IDS/IPS

Спосіб реагування	Суть реагування	Сфера застосування	Приклади реалізації
Генерація сповіщення	Створення повідомлення про інцидент для оператора безпеки	Переважно IDS	SIEM-системи, журналювання подій
Блокування трафіку	Автоматичне переривання передавання підозрілих даних	IPS	Drop Rule в Suricata, Firewall Blocking
Завершення сесії	Примусове закриття небезпечних мережевих з'єднань	IPS	TCP Reset, Session Termination
Динамічна зміна політик	Модифікація правил брандмауера або маршрутів для обмеження активності	IPS	Динамічне оновлення ACL, quarantine policies
Передача інформації іншим системам	Інтеграція реагування із зовнішніми механізмами безпеки	IDS/IPS	Взаємодія з SIEM, SOAR системами

Реагування на виявлені інциденти дозволяє системам IDS/IPS переходити від фази спостереження до фази активного захисту, забезпечуючи безперервний контроль за розвитком подій у середовищі. Вибір способу реагування залежить від цілей захисту, критичності середовища та типу загроз, які потрібно нейтралізувати. Ретельно продумана стратегія реагування є основою для побудови ефективної архітектури безпеки, здатної оперативно адаптуватися до змін у характері атак.

Загалом принципи роботи IDS та IPS систем спрямовані на створення надійної, гнучкої та адаптивної системи захисту, здатної виявляти як відомі, так і нові загрози. У контексті побудови систем із використанням машинного навчання особливе значення набуває здатність до самонавчання, постійної адаптації профілів безпеки та автоматизації процесів реагування на інциденти.

1.3 Компаративний аналіз IDS та IPS в відповідь на новітні загрози та технологічні рішення

Еволюція технік атак, поява поліморфного шкідливого програмного забезпечення, використання шифрованого трафіку та складні багатоетапні атаки вимагають від IDS/IPS систем високої гнучкості, адаптивності та інтелектуального аналізу даних [18]. Традиційні підходи, що базуються виключно на сигнатурному виявленні, вже не забезпечують необхідного рівня захисту в умовах динамічного змінення загрозового середовища. У зв’язку з цим сучасні рішення у сфері захисту інтегрують методи машинного навчання, поведінковий аналіз і технології автоматизованого реагування на інциденти, що дозволяє системам IDS/IPS не лише виявляти відомі атаки, але й вчасно виявляти нові або модифіковані загрози, які раніше не фігурували у базах даних [19]. Крім того, інтеграція інтелектуальних механізмів сприяє зниженню навантаження на операторів безпеки завдяки автоматизації обробки великої кількості подій та інцидентів.

Основним завданням IDS є постійний моніторинг мережевого або хостового середовища з метою виявлення спроб несанкціонованого доступу,

порушення політик безпеки або інших аномальних дій [20]. Системи виявлення вторгнень працюють у пасивному режимі, не змінюючи трафік, що дає змогу фіксувати широкий спектр загроз без ризику впливу на продуктивність мережі чи серверів. Залежно від архітектури та сфери застосування, IDS можуть бути орієнтовані на мережевий трафік або події на рівні окремих пристроїв [21]. У контексті розвитку новітніх загроз велике значення має також методика виявлення – від класичного сигнатурного аналізу до поведінкових і статистичних методів. Правильний вибір системи виявлення вторгнень залежить від особливостей інфраструктури, обсягів оброблюваних даних та цільових вимог до безпеки.

Для систематизації основних відмінностей між найбільш поширеними системами виявлення вторгнень у табл. 1.4 наведено порівняльну характеристику трьох IDS-рішень за загальними характеристиками.

Таблиця 1.4.

Порівняльна характеристика систем IDS за загальними ознаками

Характеристика	Snort	Suricata	OSSEC
Тип системи	Мережева IDS	Мережева IDS	Хостова IDS
Метод виявлення	Сигнатурний та аномальний	Сигнатурний та статистичний	Аналіз логів та поведінки
Принцип роботи	Аналіз мережевих пакетів	Паралельна обробка потоків	Моніторинг системних подій
Підтримка режиму IPS	Так (з додатковим налаштуванням)	Так (вбудована підтримка)	Ні
Особливості масштабування	Вимагає оптимізації при великих обсягах трафіку	Оптимізована для високих навантажень	Призначена для окремих серверів
Підтримка шифрованого трафіку	Обмежена	Підтримка через додаткові модулі	Непряма (аналіз локальних подій)
Оновлення баз знань	Через вручну оновлювані правила	Автоматизовані оновлення	Підтримка зовнішніх правил
Орієнтація на середовище	Корпоративні мережі	Провайдерські та великі мережі	Сервери, критичні хости

Таким чином, аналіз характеристик найбільш поширених систем IDS дозволяє зробити висновок, що вибір конкретного рішення має ґрунтуватися на вимогах до масштабованості, типу середовища, специфіки оброблюваного

трафіку та необхідності інтеграції з іншими системами захисту. Різні системи по-різному підходять до забезпечення виявлення загроз, і саме правильний підбір технології дає змогу досягти найкращих результатів у межах заданої інформаційної інфраструктури.

Окрім систем виявлення вторгнень, важливу роль у забезпеченні комплексного захисту інформаційних систем відіграють системи запобігання вторгненням. На відміну від IDS, які обмежуються виявленням і повідомленням про підозрілу активність, IPS діють проактивно, здійснюючи автоматичне блокування або обмеження шкідливої активності у мережі в режимі реального часу. Основною задачею IPS є не лише виявлення загроз, але й негайне реагування на них з метою запобігання розвитку атаки або нанесенню шкоди критичним елементам інфраструктури [22]. Системи IPS інтегруються безпосередньо у шлях передавання трафіку, аналізуючи пакети даних перед їх передаванням до кінцевого призначення, що дозволяє миттєво припиняти небажану активність. У сучасних рішеннях IPS використовуються як класичні сигнатурні методи виявлення, так і поведінковий аналіз та машинне навчання для виявлення невідомих атак. Ретельний вибір системи IPS залежить від вимог до часу реагування, допустимого рівня затримок у трафіку та особливостей архітектури захищуваної мережі.

Для узагальнення особливостей різних рішень у сфері запобігання вторгненням у табл. 1.5 наведено порівняльну характеристику трьох поширених систем IPS за основними критеріями.

Таблиця 1.5.

Порівняльна характеристика систем IPS за загальними ознаками

Характеристика	Cisco Firepower IPS	Suricata IPS	Snort IPS
1	2	3	4
Тип системи	Апаратно-програмна IPS	Програмна IPS	Програмна IPS
Метод виявлення	Сигнатурний, політики безпеки, поведінковий аналіз	Сигнатурний та статистичний аналіз	Сигнатурний аналіз
Принцип роботи	Інлайн-аналіз трафіку та активне блокування	Інлайн-обробка мережевого потоку	Інлайн-обробка пакетів

Продовження таблиці 1.5.

1	2	3	4
Підтримка шифрованого трафіку	Так (з дешифрацією)	Часткова підтримка через проксі	Обмежена
Можливості масштабування	Висока завдяки спеціалізованим апаратним платформам	Підтримка високопродуктивних мереж	Потребує оптимізації при великих навантаженнях
Оновлення баз знань	Автоматичне через Cisco Talos	Підтримка зовнішніх правил і оновлень	Через оновлення баз правил
Орієнтація на середовище	Великі корпоративні мережі, критичні об'єкти	Провайдерські мережі, великі організації	Малі та середні корпоративні мережі
Реалізація політик безпеки	Поглиблене управління політиками, профілювання додатків	Гнучке налаштування правил	Базова підтримка налаштувань правил

Таким чином, системи запобігання вторгненням орієнтовані на активний захист інформаційних ресурсів завдяки можливості миттєвого реагування на виявлені загрози. Порівняння характеристик різних IPS-рішень дозволяє обрати оптимальну систему залежно від вимог до рівня захисту, продуктивності та масштабованості мережевої інфраструктури. Грамотна інтеграція IPS у мережеве середовище є критичним чинником побудови ефективної стратегії кіберзахисту.

Висновки до розділу 1

У рамках даного розділу було проведено комплексний аналіз сучасних підходів до побудови систем виявлення та запобігання вторгненням з метою забезпечення теоретичної основи для подальшого розроблення практичного рішення. Розглянуто сутність IDS та IPS систем, здійснено їх класифікацію за ключовими ознаками, зокрема за джерелом даних, способом виявлення загроз, способом розгортання та рівнем інтеграції у інформаційне середовище. Це дозволило сформувати уявлення про основні типи систем та їх місце в загальній архітектурі захисту інформаційних ресурсів.

Детально проаналізовано ключові принципи функціонування IDS та IPS систем, зокрема підходи до безперервного моніторингу активності, методи

аналізу поведінки об'єктів та способи реагування на виявлені інциденти. Узагальнення відповідних методів надало можливість визначити механізми, які є найбільш перспективними для інтеграції в системи нового покоління з орієнтацією на виявлення складних і невідомих загроз.

Проведено компаративний аналіз найбільш поширених систем IDS та IPS у контексті новітніх загроз та технологічних рішень. У межах порівняльних таблиць розглянуто характеристики систем Snort, Suricata, OSSEC, Cisco Firepower IPS, Suricata IPS та Snort IPS, що дозволило виявити їхні сильні сторони, особливості застосування та обмеження в різних сценаріях використання.

Отримані результати аналізу створюють підґрунтя для обґрунтованого вибору методів виявлення та запобігання вторгненням, що буде використано у наступному розділі для побудови методики розроблення системи виявлення загроз із використанням сучасних підходів до обробки даних і машинного навчання.

РОЗДІЛ 2 МЕТОДИ ВИЯВЛЕННЯ ТА ПОПЕРЕДЖЕННЯ ВТОРГНЕННЯМ (IDS/IPS)

2.1 Аналіз сигнатурних та аномальних методів виявлення вторгнень

Методи виявлення вторгнень становлять основу ефективного функціонування систем IDS/IPS і визначають здатність системи своєчасно ідентифікувати шкідливу активність у мережах або на хостах. У загальному випадку під методами виявлення розуміють механізми обробки даних про події з метою виявлення ознак порушення безпеки. Вибір відповідного методу безпосередньо впливає на чутливість системи до новітніх загроз, її здатність мінімізувати кількість хибних спрацьовувань і забезпечити належний рівень продуктивності. У межах систем виявлення вторгнень традиційно виокремлюють два основних напрями: сигнатурні методи та аномальні методи.

Сигнатурні методи виявлення базуються на зіставленні спостережуваної активності з базою відомих шаблонів атак, що дозволяє ефективно виявляти загрози, які вже були класифіковані і документовані [23]. Аномальні методи, своєю чергою, спрямовані на виявлення відхилень від профілю нормальної поведінки системи або користувачів, що дозволяє ідентифікувати нові, раніше невідомі типи атак.

У табл. 2.1 наведено компаративну характеристику трьох методів за загальними ознаками.

Таблиця 2.1.

Порівняльна характеристика систем IPS за загальними ознаками

Характеристика	Сигнатурний аналіз	Аномальний аналіз	Гібридний підхід
1	2	3	4
Принцип роботи	Пошук відповідності шаблонам атак	Виявлення відхилень від нормальної поведінки	Комбінація сигнатурного та аномального аналізу
Здатність виявляти нові загрози	Низька	Висока	Висока

Продовження таблиці 2.1.

1	2	3	4
Частота хибних спрацьовувань	Низька	Вища	Середня
Необхідність оновлення баз знань	Висока	Низька (періодичне перенавчання)	Залежить від компонентів
Потреба в попередніх даних про атаки	Обов'язкова	Необов'язкова	Часткова
Сфера ефективного застосування	Відомі типи атак	Невідомі або модифіковані атаки	Універсальне застосування
Вимоги до обчислювальних ресурсів	Помірні	Високі	Високі
Приклади реалізації	Snort, Cisco IPS	Machine Learning IDS, anomaly-based IPS	Suricata, Hybrid IDS/IPS

Отже, аналіз основних методів виявлення вторгнень демонструє, що вибір підходу має базуватися на характері загрозливого середовища, можливостях обчислювальної інфраструктури та вимогах до рівня безпеки. Сигнатурні методи забезпечують швидке виявлення відомих атак із мінімальним рівнем хибних спрацьовувань, тоді як аномальні методи дозволяють своєчасно реагувати на нові загрози. У сучасних умовах найбільш перспективними є гібридні рішення, що комбінують переваги обох підходів для досягнення балансу між точністю, адаптивністю та продуктивністю систем IDS/IPS.

Поряд із сигнатурними підходами дедалі більшої актуальності у сучасних системах IDS/IPS набувають аномальні методи виявлення вторгнень [24]. Використання виключно сигнатурного аналізу в умовах постійної еволюції кіберзагроз уже не є достатнім для забезпечення належного рівня захисту інформаційних систем. У зв’язку з цим розробляються й впроваджуються методи, що дозволяють фіксувати нетипову або підозрілу активність без попереднього знання про конкретний тип атаки.

Аномальні методи базуються на концепції створення моделей нормальної поведінки користувачів, мережевого трафіку чи системних процесів, із подальшим виявленням відхилень від цих моделей. Формування такої базової лінії дозволяє системі визначати не тільки стандартні дії, а й своєчасно фіксувати спроби порушення безпеки, навіть якщо конкретна атака раніше не була задокументована. Особливостями аномальних методів є здатність до

адаптації, можливість виявляти невідомі загрози та робота у змінних умовах функціонування інформаційного середовища [25]. Водночас побудова та підтримка актуальності моделей нормальної поведінки потребують використання складних математичних і статистичних інструментів, зокрема методів машинного навчання, кластеризації, аналізу часових рядів тощо.

Для систематизації основних підходів у межах аномальних методів виявлення вторгнень у табл. 2.2 наведено порівняльну характеристику трьох базових методів за загальними характеристиками.

Таблиця 2.2.

Порівняльна характеристика основних аномальних методів

Характеристика	Статистичний аналіз	Кластеризація	Методи виявлення викидів
Принцип роботи	Створення порогових моделей на основі статистичних норм	Групування об'єктів за схожістю та виявлення аномалій через відхилення від кластерів	Виявлення точок, що істотно відрізняються від основної маси даних
Потреба у тренувальних даних	Потрібні дані нормальної поведінки	Потрібні набори нормальної активності	Потрібні історичні дані для побудови моделей
Спосіб виявлення	Порівняння з допустимими межами	Виявлення об'єктів, що не належать до кластерів	Виявлення об'єктів за межами статистичних норм
Тип даних	Статистичні показники активності	Векторні представлення поведінки	Будь-які числові або категоріальні дані
Вимоги до обчислювальних ресурсів	Помірні	Високі	Залежить від обсягу даних
Актуальність в реальному часі	Потребує адаптації для потокових даних	Може бути обмеженою	Підходить для пакетного аналізу
Приклади застосування	Аномалії в обсягах трафіку	Виявлення незвичних комунікаційних патернів	Виявлення внутрішніх порушень політик

Отже, аномальні методи виявлення вторгнень надають системам IDS/IPS здатність протидіяти новітнім загрозам за рахунок аналізу нетипової поведінки об'єктів у системі. Комплексне застосування статистичних підходів, кластеризації та виявлення викидів дозволяє підвищити гнучкість систем

безпеки й забезпечити їхню адаптацію до динамічно змінюваного ландшафту загроз.

2.2 Вибір методу машинного навчання для виявлення вторгнення

Вибір методу машинного навчання для побудови системи виявлення та запобігання вторгненням є ключовим етапом розроблення ефективного рішення. Оскільки задачі IDS/IPS часто зводяться до класифікації об'єктів мережевого або системного трафіку за ознаками нормальної чи аномальної поведінки, обраний метод має забезпечувати високу точність виявлення атак, швидку обробку даних та здатність працювати з великими обсягами інформації у реальному часі. Залежно від поставленої задачі та характеристик середовища використання можуть застосовуватися різні алгоритми, серед яких найбільш поширеними є дерева рішень, методи ансамблю моделей, методи опорних векторів та алгоритми нейронних мереж.

Особливо важливо при виборі методу враховувати такі чинники, як інтерпретованість моделі, стійкість до нерівномірності класів у даних, можливість масштабування на великі датасети та ефективність у режимі обробки потокових даних. Методи, які поєднують у собі високу продуктивність і здатність до роботи з реальними системами без значного ускладнення обчислювальних процесів, є найбільш придатними для використання в системах IDS/IPS.

Одним із популярних та ефективних методів машинного навчання, який поєднує високу швидкість роботи з високою точністю прогнозування, є метод швидке дерево (ШД, FastTree). Його застосування особливо актуальне в задачах класифікації та регресії, де важливо забезпечити баланс між продуктивністю та якістю моделі [26]. На рис. 2.1 подано структурну схему роботи методу ШД, яка ілюструє основні етапи його функціонування.

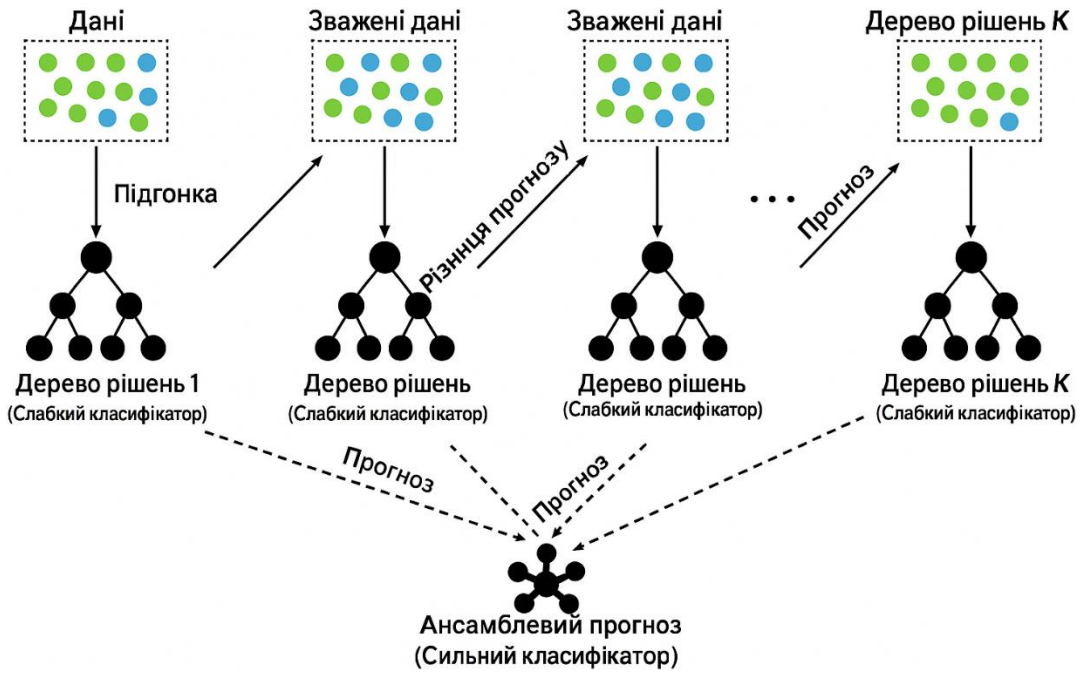

Рис. 2.1. Структурна схема роботи методу ШД

На початковому етапі метод використовує наявний набір даних, для якого здійснюється побудова першого дерева рішень. Це дерево є слабким класифікатором, що намагається моделювати залежності між вхідними ознаками та цільовою змінною. Після формування першого дерева обчислюються залишки – різниця між фактичними та передбаченими значеннями. Ці залишки відображають недоліки поточної моделі та визначають напрямок подальшого вдосконалення.

На основі залишків модифікуються ваги навчальних прикладів, що приводить до формування нового набору зважених даних. Саме цей набір даних використовується для навчання наступного дерева рішень, яке спрямоване на виправлення помилок попереднього дерева. Процес ітеративно повторюється: після кожної побудови дерева знову обчислюються залишки, оновлюються ваги даних і формується наступне дерево. Кожне дерево у цьому ансамблі залишається слабким класифікатором, однак разом вони забезпечують побудову потужної агрегованої моделі.

У завершальному етапі відбувається об'єднання прогнозів усіх побудованих дерев у єдиний ансамблевий прогноз. При цьому результати окремих слабких моделей комбінуються із застосуванням певних вагових

коефіцієнтів, що дозволяє суттєво підвищити загальну точність системи. Таким чином, ШД реалізує концепцію градієнтного бустингу дерев рішень із оптимізованими алгоритмічними підходами для пришвидшення процесу навчання, що робить його одним із найбільш придатних для задач, де важлива як висока швидкість обробки даних, так і якість прийнятих рішень.

Переваги методу швидке дерево:

– висока швидкість навчання і прогнозування завдяки оптимізованим алгоритмам обробки великих обсягів даних [27];

– забезпечення хорошого балансу між точністю класифікації та продуктивністю навіть при роботі з великими та складними датасетами [28];

– відносна стійкість до перенавчання завдяки використанню спеціальних механізмів оптимізації розбиття даних на вузли [29].

Недоліки методу швидке дерево:

– чутливість до нерівномірного розподілу класів у тренувальній вибірці, що може впливати на якість класифікації [30];

– обмежена інтерпретованість отриманої моделі порівняно з класичними деревами рішень через більшу складність побудови структури [31];

– залежність результатів від якості попередньої обробки даних і правильного налаштування параметрів моделі [32].

Отже, метод швидке дерево є ефективним інструментом для розв’язання задач класифікації в системах виявлення вторгнень, особливо коли важливими є швидкість обробки та здатність працювати з великими масивами інформації. Використання ШД дає змогу будувати моделі з високою продуктивністю при помірних витратах ресурсів. Проте при застосуванні цього методу слід враховувати особливості структури даних і можливі виклики у налаштуванні для досягнення оптимальної якості результатів.

Серед методів машинного навчання, які застосовуються для задач виявлення вторгнень також особливу увагу привертає метод *опорних векторів* (ОВ, Support Vector Machine) [33]. Завдяки своїй здатності ефективно працювати у просторах великої розмірності та будувати оптимальні розділяючі

гіперплощини між класами даних, цей метод активно використовується в системах IDS/IPS для класифікації трафіку, виявлення аномалій та прогнозування поведінки [34]. Метод опорних векторів демонструє високу стійкість до перенавчання при правильному налаштуванні параметрів і дозволяє досягати добрих результатів навіть за наявності обмеженої кількості навчальних даних [35]. Структурна схема роботи методу опорних векторів подана на рис. 2.2.

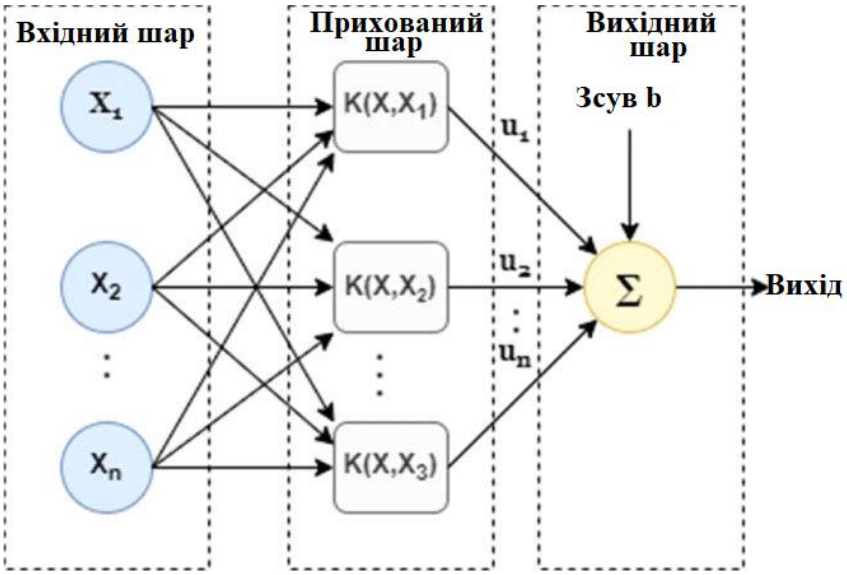

Рис. 2.2. Схема роботи методу ОВ

На вхідному шарі здійснюється подача ознак об'єкта (X_1, X_2, ..., X_n), які характеризують відповідний зразок даних. Кожен вхід з'єднаний із прихованим шаром, де відбувається обчислення функцій ядра $K(X, X_i)$, що визначають міру подібності між вхідним зразком і опорними векторами. Функція ядра дозволяє здійснювати проєкцію вхідних даних у простір вищої розмірності для пошуку лінійної розділяючої поверхні навіть для нелінійно розділених даних. Отримані результати зважуються коефіцієнтами (u_1, u_2, ..., u_n), що були визначені під час навчання моделі, після чого передаються на вихідний шар. На вихідному шарі виконується підсумовування з урахуванням зсуву b, що відповідає зміщенню гіперплощини розділення. Результатом обчислень є вихідне значення моделі, яке визначає клас вхідного об'єкта або вірогідність його належності до певної категорії.

Переваги методу опорних векторів:

– висока ефективність у задачах класифікації, особливо при роботі з даними великої розмірності та складними межами розділення [36];

– можливість побудови нелінійних моделей за допомогою використання різних функцій ядра без потреби явної трансформації ознак [37];

– стійкість до перенавчання, зокрема у випадках обмеженої кількості навчальних даних, за рахунок використання принципу максимізації зазору між класами [38].

Недоліки методу ОВ:

– високі обчислювальні витрати при роботі з великими наборами даних, що ускладнює застосування у реальному часі без спеціальної оптимізації [39];

– чутливість до вибору функції ядра та налаштування гіперпараметрів, що без належної оптимізації може призвести до зниження якості моделі [40];

– складність інтерпретації побудованої моделі, особливо у випадку використання складних або комбінованих ядерних функцій [41].

Таким чином, метод опорних векторів є потужним інструментом для задач класифікації та виявлення аномалій, особливо у випадках, коли класи мають складну структуру. Його застосування дозволяє забезпечити високу точність розмежування класів навіть у високорозмірнісних просторах. Разом із тим ефективне використання методу потребує уважного налаштування і оптимізації параметрів, що варто враховувати при його впровадженні в системи виявлення вторгнень.

У задачах виявлення вторгнень широко застосовуються методи, що базуються на теорії ймовірностей і статистичному аналізі даних [42]. Серед них наївний байєсівський класифікатор вирізняється простотою реалізації, високою швидкістю роботи та здатністю забезпечувати якісну класифікацію навіть за великої кількості вхідних ознак. Цей метод використовується в системах IDS/IPS для категоризації мережевого трафіку, виявлення аномальної активності та оцінки ризиків на основі ймовірнісних моделей.

Структурний принцип роботи наївного байєсівського класифікатора подано на рис. 2.3.

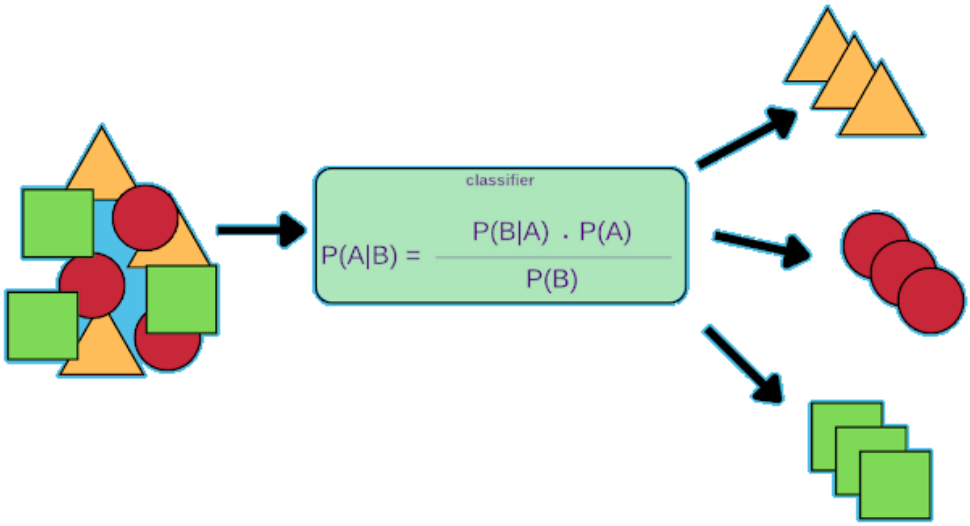

Рис. 2.3. Схема роботи методу НБК

Вхідний набір даних складається з об'єктів, що мають різні ознаки і можуть належати до різних класів. Кожен об'єкт описується набором атрибутів, які розглядаються як умовно незалежні один від одного. Усередині класифікатора обчислюється ймовірність належності об'єкта до кожного з можливих класів на основі формули Байєса, яка враховує апостеріорну ймовірність класу за умови заданих ознак. Після розрахунку ймовірностей об'єкти автоматично розподіляються відповідно до класу з найбільшою ймовірністю, що дозволяє ефективно здійснювати процес класифікації на основі статистичної оцінки. Як видно з рисунка, об'єкти із загального набору групуються за категоріями на підставі результатів ймовірнісного аналізу.

Переваги наївного байєсівського класифікатора:

– висока швидкість навчання і класифікації навіть при роботі з великими обсягами даних [43];

– стійкість до високої розмірності простору ознак, що дозволяє ефективно працювати з даними, які мають велику кількість параметрів [44];

– простота реалізації та невисокі вимоги до обчислювальних ресурсів, що робить метод придатним для широкого кола практичних застосувань [45].

Недоліки наївного байєсівського класифікатора:

– припущення про незалежність ознак часто не відповідає реальним даним, що може впливати на точність класифікації [46];

– обмежена здатність моделювати складні залежності між ознаками через спрощену ймовірнісну модель [47];

– залежність результатів від якості попередньої обробки даних, зокрема від правильності представлення ознак та їх масштабування [48].

Отже, наївний байєсівський класифікатор є ефективним рішенням для задач швидкої класифікації, особливо у випадках великої кількості ознак і обмежених ресурсів. Завдяки своїй простоті та швидкодії він залишається актуальним для використання в системах виявлення вторгнень. Однак у випадках складної залежності між ознаками слід розглядати можливість використання більш гнучких або комбінованих методів.

Враховуючи розглянуті особливості методів машинного навчання, доцільно здійснити їх порівняння за основними характеристиками, що є критичними при побудові систем виявлення вторгнень. До таких характеристик належать швидкість навчання та прогнозування, стійкість до перенавчання, здатність працювати з великими обсягами даних, вимоги до обчислювальних ресурсів та рівень складності налаштування. Для узагальнення відомостей про методи швидке дерево, опорних векторів та наївний байєсівський класифікатор у таблиці 2.4 наведено їх компаративну характеристику за зазначеними критеріями.

Таблиця 2.4.

Порівняльна характеристика методів МН для виявлення вторгнень

Характеристика	Швидке дерево	Метод опорних векторів	Наївний байєсівський класифікатор
Швидкість навчання та прогнозування	Висока	Помірна	Висока
Стійкість до перенавчання	Висока	Висока	Помірна
Робота з великими обсягами даних	Оптимізована	Ускладнена при великих даних	Підтримка великих обсягів
Вимоги до обчислювальних ресурсів	Помірні	Високі	Низькі
Інтерпретованість результатів	Помірна	Обмежена	Висока
Складність налаштування	Помірна	Висока	Низька
Підтримка роботи в реальному часі	Підтримується	Обмежена	Підтримується

Для розробки системи виявлення та попередження вторгнень доцільно обрати метод швидке дерево, оскільки він забезпечує високу швидкість навчання та прогнозування при помірних вимогах до обчислювальних ресурсів. Метод демонструє хорошу стійкість до перенавчання, що є критично важливим для підтримання стабільної якості класифікації в умовах динамічно змінюваних даних. Завдяки оптимізації обробки великих обсягів інформації ШД дозволяє ефективно працювати з реальними мережевими потоками та великими датасетами. Враховуючи баланс між продуктивністю, адаптивністю та практичною реалізованістю, метод швидке дерево є обґрунтованим вибором для побудови системи виявлення загроз.

2.3 Обґрунтування вибору тренувального набору даних

Однією з ключових умов якісного функціонування системи виявлення та попередження вторгнень є використання релевантного, репрезентативного та повного тренувального набору даних, який достовірно відображає реальну мережеву активність, охоплюючи нормальні, аномальні та зловмисні потоки. Наявність повноти та різноманітності вхідних даних безпосередньо впливає на здатність моделі розрізняти типові та нетипові шаблони поведінки, що є критичним для зниження кількості хибнопозитивних і хибнонегативних спрацьовувань. З цією метою для навчання моделі було обрано датасет LUFlow, створений у межах дослідницької ініціативи університету Ланкастера, який спеціалізується на потоковому аналізі мережевого трафіку [49]. Його структура охоплює широкий спектр сценаріїв взаємодії в мережі, що дозволяє моделі адаптуватися до різноманітних умов експлуатації. Унікальною рисою цього набору є його постійне оновлення завдяки автоматизованому механізму маркування, який базується на кореляції із зовнішніми джерелами розвідки кіберзагроз (CTI – Cyber Threat Intelligence), що забезпечує фіксацію як відомих, так і новітніх атак. Таким чином, LUFlow формує динамічну й

актуальну основу для навчання моделей машинного навчання в контексті мережевої безпеки.

У табл. 2.5 наведено опис основних атрибутів, які використовуються для аналізу потоків і формування ознак моделі машинного навчання.

Таблиця 2.5.

Опис атрибутів датасету

№	Атрибут	Опис
1	src_ip	Вихідна IP-адреса потоку. Анонімізована відповідно до автономної системи
2	src_port	Вихідний порт потоку (номер порту джерела)
3	dest_ip	Призначена IP-адреса потоку. Також анонімізована як і src_ip
4	dest_port	Порт призначення (номер порту отримувача)
5	proto	Номер мережевого протоколу, наприклад TCP – 6
6	bytes_in	Кількість байтів, переданих від джерела до отримувача
7	bytes_out	Кількість байтів, переданих від отримувача до джерела
8	num_pkts_in	Кількість пакетів, переданих від джерела до отримувача
9	num_pkts_out	Кількість пакетів, переданих від отримувача до джерела
10	entropy	Середня ентропія (в бітах на байт) даних потоку. Варіюється в межах від 0 до 8
11	total_entropy	Загальна ентропія в байтах по всіх даних потоку
12	avg_ipt	Середній інтервал між прибуттям пакетів у потоці (mean inter-packet time)
13	time_start	Час початку потоку у секундах з початку епохи UNIX (timestamp)
14	time_end	Час завершення потоку у секундах з початку епохи UNIX.
15	duration	Тривалість потоку з мікросекундною точністю.
16	label	Мітка класу потоку: benign (нормальний), outlier (аномальний), malicious (зловмисний).

Для забезпечення ефективного навчання моделі машинного навчання важливо не лише обрати релевантний набір даних, а й здійснити попередній аналіз кореляцій між ознаками. Такий аналіз дозволяє виявити надлишкові, слабоінформативні або надто залежні між собою атрибути, що можуть вплинути на якість побудови моделі та її узагальнювальні здатності. У межах дослідження проведено обчислення коефіцієнтів кореляції між числовими ознаками датасету LUFlow. Результати візуалізовано у вигляді теплової карти на рис. 2.4, де інтенсивність кольору відображає силу та напрямок лінійного зв’язку між ознаками.

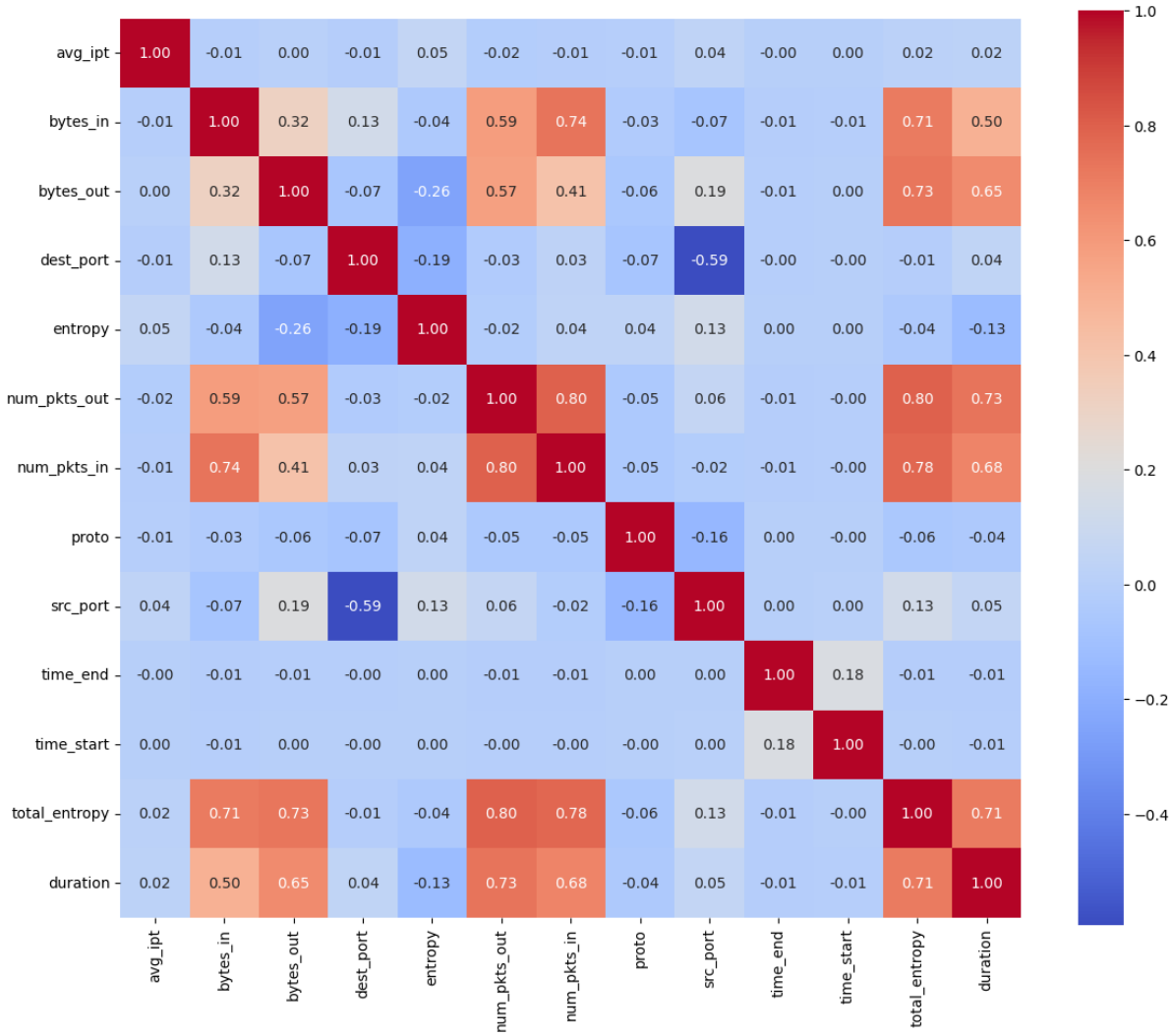

Рис. 2.4. Теплова карта кореляції

На основі отриманої теплової карти кореляції можна виділити кілька ключових спостережень. Найсильніший позитивний зв’язок спостерігається між ознаками bytes_out і num_pkts_out, а також між bytes_in і num_pkts_in, що є очікуваним, оскільки збільшення кількості пакетів зазвичай супроводжується зростанням переданих байтів. Схожі закономірності простежуються і для ознаки total_entropy, яка суттєво корелює з обсягами вхідних та вихідних даних (bytes_in, bytes_out) і кількістю пакетів. Помірна кореляція виявлена між duration і кількісними характеристиками потоку, такими як bytes_out, num_pkts_in та total_entropy, що може свідчити про залежність тривалості потоку від його інтенсивності. Водночас атрибути proto, src_port, time_start і time_end не мають виражених лінійних зв’язків з іншими ознаками, що вказує на їхню потенційну незалежність або складнішу природу взаємозв’язку, яка не

виявляється у лінійному вигляді. Варто зазначити також відсутність критичних мультиколінеарних зв’язків, що дозволяє зберегти більшість ознак у моделі без ризику втрати стійкості навчання.

Для виявлення відмінностей у поведінці мережевих потоків залежно від їх належності до певного типу (нормального або аномального), доцільно здійснити порівняльний аналіз розподілу тривалості потоків. На рис. 2.5 наведено графік щільності розподілу ознаки duration для класів benign (нормальний трафік) та outlier (аномальний трафік), що дозволяє оцінити характерні відмінності у часових характеристиках потоків.

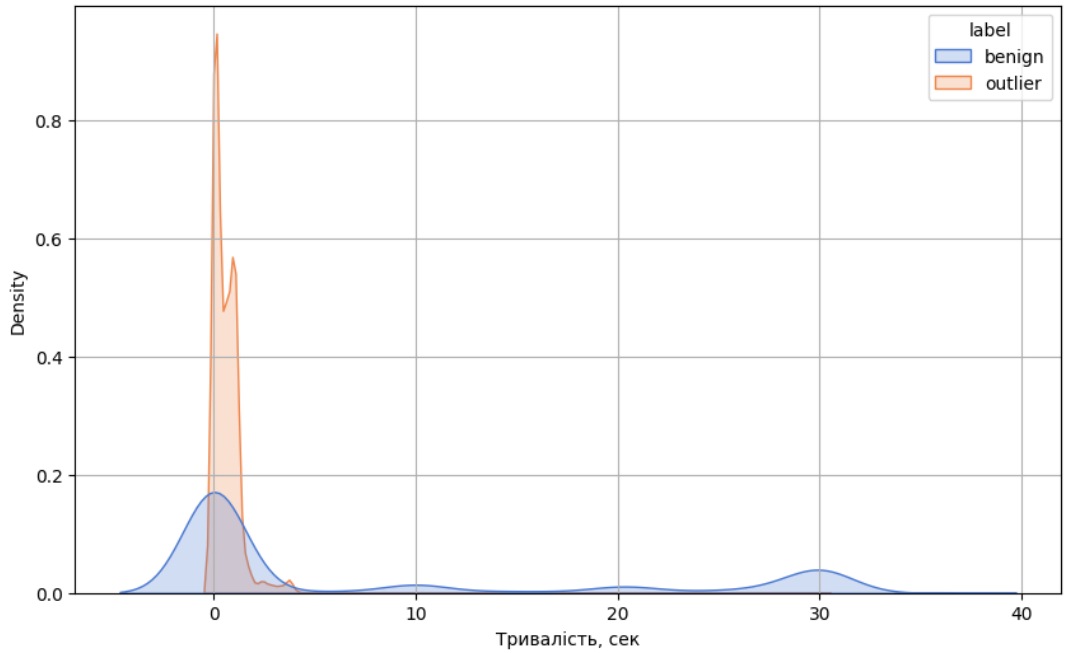

Рис. 2.5. Розподіл тривалості (duration) за типом трафіку

Як видно з графіка, розподіл тривалості для аномального трафіку зосереджений у вузькому діапазоні – більшість потоків мають тривалість менше 2 секунд із чітко вираженим піком поблизу нуля. Така компактність свідчить про типову короткочасність нетипових потоків, що не притаманне для звичайного трафіку. У свою чергу, нормальні потоки (benign) мають значно ширший розподіл: хоча також спостерігається пік при низьких значеннях, трапляються й потоки з тривалістю до 30–40 секунд, що відповідає звичайній поведінці сервісів з довшими сесіями передачі даних. Ці відмінності є важливими при побудові моделей класифікації, оскільки дозволяють

використовувати тривалість як один з інформативних параметрів для диференціації між типами трафіку.

Рис. 2.6 ілюструє розподіл кількості записів у датасеті LUFlow за мітками класів трафіку, що дозволяє оцінити збалансованість вибірки перед початком навчання моделі. На діаграмі видно, що обидва класи – benign (нормальний трафік) та outlier (аномальний трафік) – представлені приблизно рівномірно. Зокрема, кількість записів нормального трафіку становить трохи більше 25 000, тоді як кількість аномальних записів наближається до 24 000. Такий розподіл є прийнятним з точки зору балансу класів і не потребує спеціальних заходів для вирівнювання ваг у процесі навчання моделі.

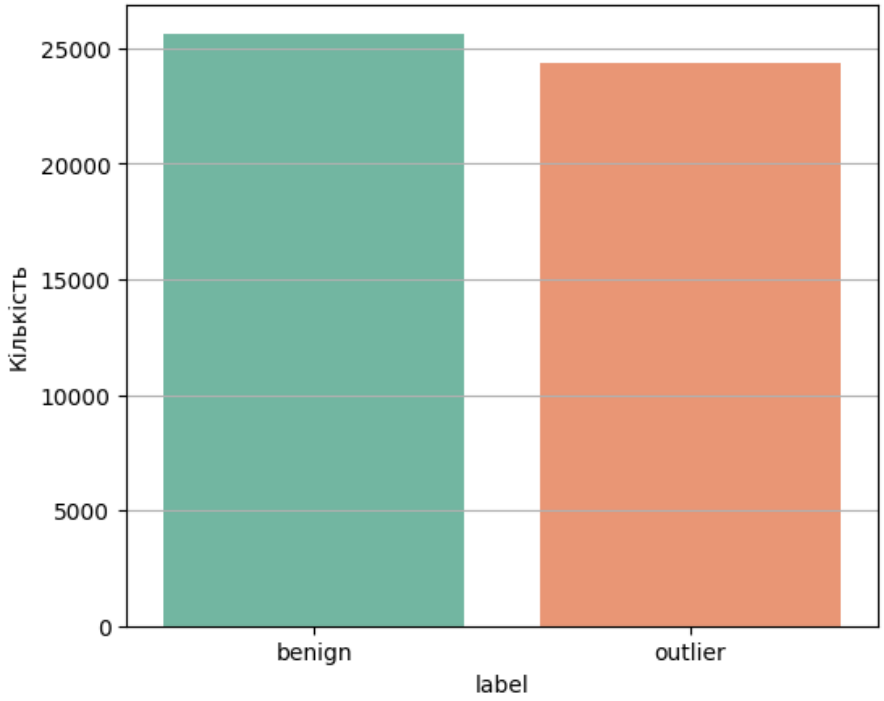

Рис. 2.6. Розподіл кількості записів за класами

Збалансованість класів є сприятливою умовою для побудови класифікаційної моделі, оскільки знижує ймовірність зміщення результатів на користь переважного класу. Це дозволяє тренувальному алгоритму ефективно вивчати характеристики як нормального, так і нетипового трафіку, покращуючи здатність системи точно виявляти потенційні інциденти.

Рис. 2.7 ілюструє попарні залежності між основними числовими ознаками датасету LUFlow, а саме: avg_ipt, entropy, total_entropy та duration. Такий тип візуалізації дозволяє виявити латентні структури, скупчення, а також

потенційні межі між класами benign та outlier, що є критично важливим при побудові ефективної моделі класифікації.

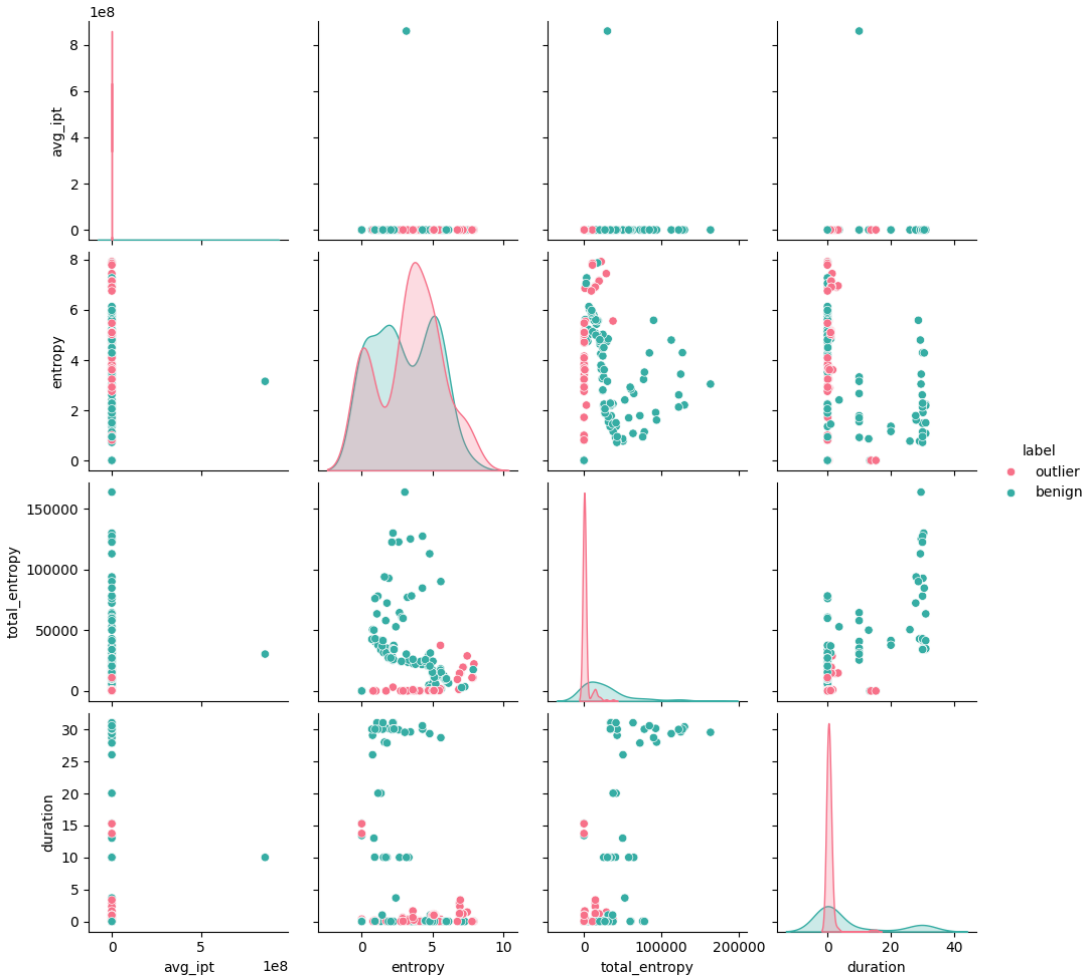

Рис. 2.7. Попарні залежності основних ознак

З рисунка видно, що найбільш розрізнені значення спостерігаються в ознаці duration, де аномальні потоки (outlier) концентруються в межах малих значень, тоді як нормальні (benign) мають ширший розкид до 40 секунд. У свою чергу, ознака entropy демонструє часткове перекриття між класами, однак певні відмінності у формі розподілу можуть бути використані як евристичні маркери. Total_entropy має тенденцію до вищих значень у класі benign, що узгоджується з ширшими потоками даних. Ознака avg_ipt виявляє сильну скупченість біля нуля, що свідчить про переважання коротких інтервалів між пакетами, проте зустрічаються і виняткові значення з великими відхиленнями.

Рис. 2.8 відображає матрицю розсіювання між ключовими числовими ознаками набору даних LUFlow: avg_ipt, entropy, total_entropy та duration. Така

візуалізація є важливим аналітичним інструментом для виявлення взаємозв'язків, скупчень, крайніх значень та потенційно нелінійних залежностей, що можуть впливати на ефективність класифікаційної моделі.

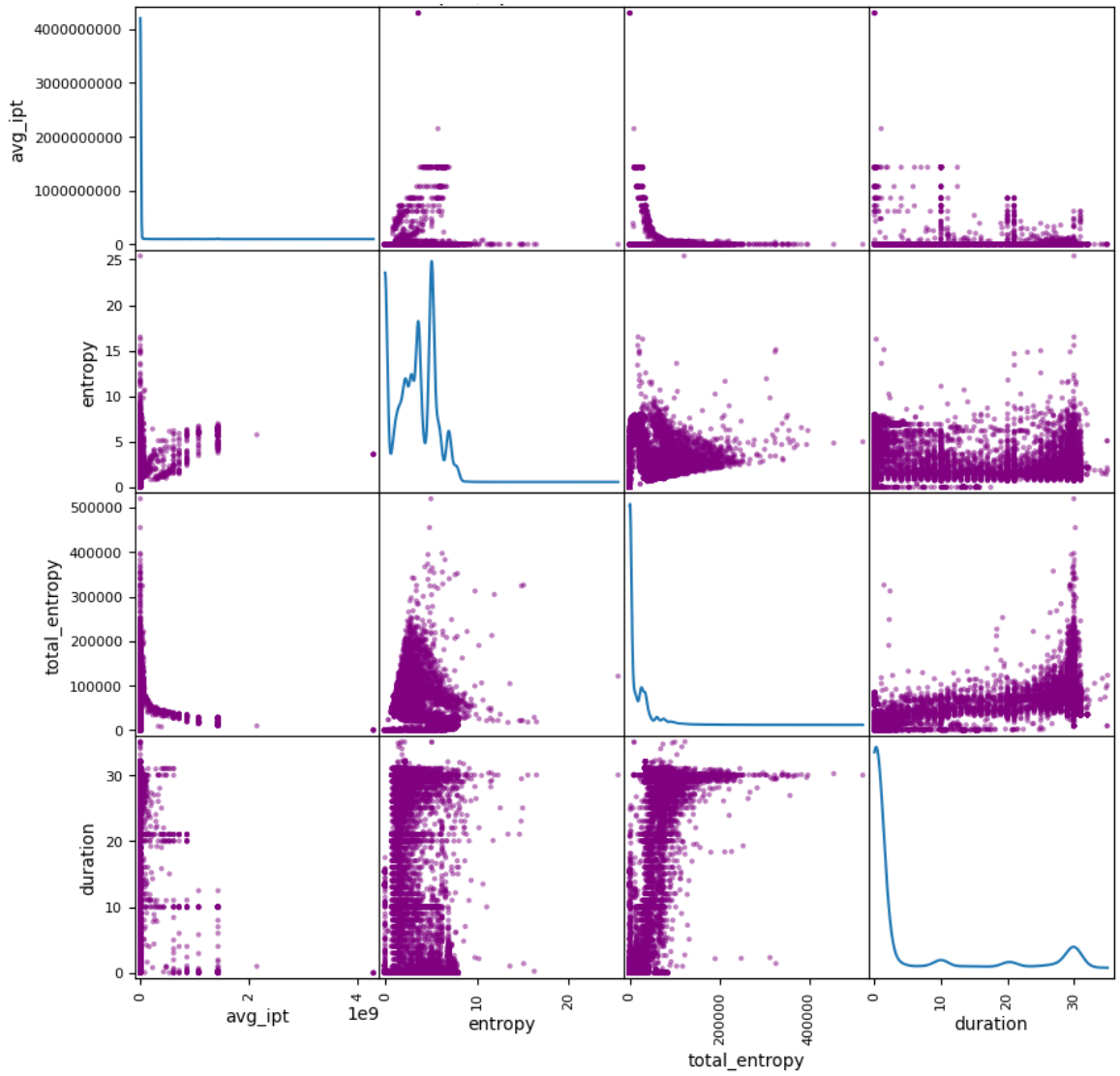

Рис. 2.8. Матриця розсіювання ключових ознак

Значення ознаки avg_ipt характеризуються крайньою варіативністю, включаючи численні великі значення, що вказує на наявність викидів або значень, які варто піддати нормалізації або логарифмічному перетворенню. Ознака entropy демонструє більш контрольований розподіл, з концентрацією значень у межах від 4 до 8, однак при цьому простежується загальна обернена залежність між entropy та total_entropy, що може бути обумовлено впливом кількості байтів у потоці. Total_entropy демонструє експоненціальний ріст при зростанні розміру потоку, що підтверджується поступовим розширенням хмар

точок зліва направо. Ознака duration також має розтягнутий розподіл із численними скупченнями у діапазоні до 10 секунд, після чого спостерігаються розрідженіші точки.

Рис. 2.9 ілюструє залежність між кількістю вихідних пакетів (num_pkts_out) та обсягом вихідних даних (bytes_out) у потоках мережевого трафіку, з розподілом за класами benign та outlier. Такий аналіз дозволяє виявити закономірності у поведінці трафіку, зокрема, наскільки обсяг переданих даних відповідає кількості переданих пакетів.

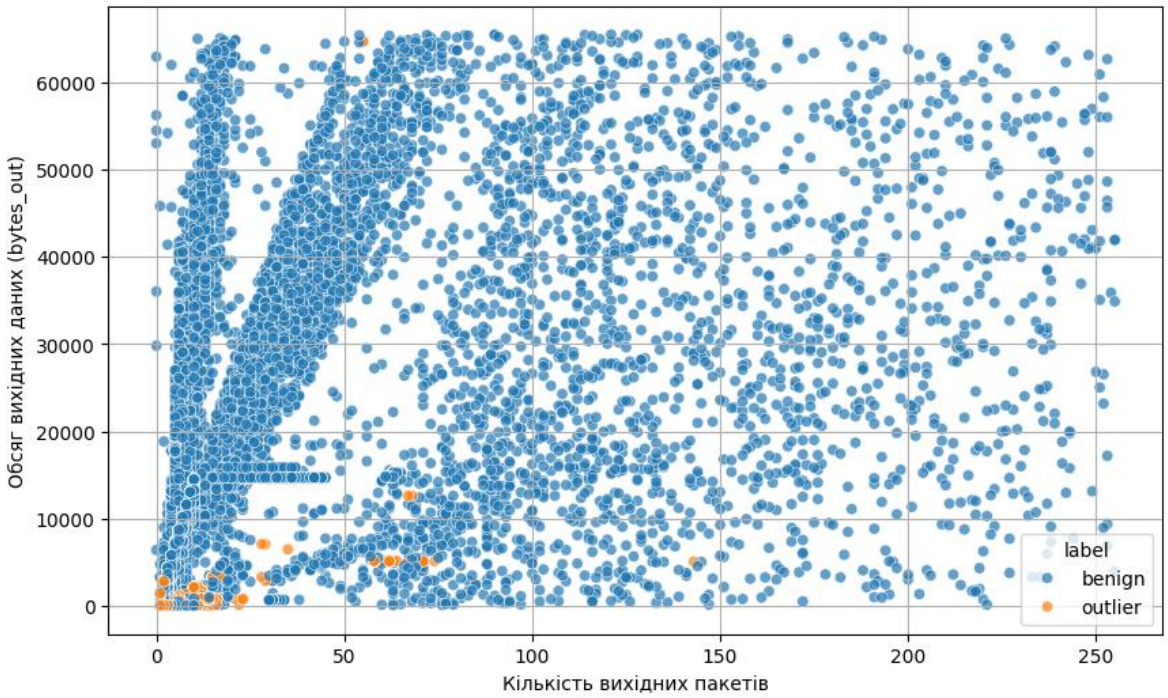

Рис. 2.9. Залежність між кількістю пакетів та обсягом вихідних даних

З рисунка видно наявність сильної позитивної залежності між кількістю вихідних пакетів і обсягом вихідних даних для нормального трафіку (benign). Ці потоки рівномірно заповнюють площину графіка і формують характерні лінійні смуги, які можуть відповідати типово використовуваним розмірам пакетів у певних протоколах. У свою чергу, аномальні потоки (outlier) концентруються переважно у нижньому лівому куті, що вказує на те, що більшість з них мають низьку активність як за кількістю пакетів, так і за обсягом даних. Це дозволяє припустити, що такі потоки можуть представляти собою короткі або нетипові сесії, які відхиляються від нормальної мережевої поведінки.

Отримані результати підтверджують інформативність обох параметрів – num_pkts_out і bytes_out – для задач класифікації, оскільки міжкласові відмінності є достатньо помітними та структурованими для використання у моделі виявлення вторгнень.

Рис. 2.10 демонструє результати кластеризації даних методом KMeans у двовимірному просторі головних компонент (PCA), що дозволяє візуалізувати внутрішню структуру датасету та оцінити розподіл записів між виявленими групами. Після пониження розмірності до двох компонент (pca1 і pca2), було застосовано кластеризацію з числом кластерів k=2, а об'єкти були розфарбовані відповідно до належності до кластерів (кластер 0 – червоний, кластер 1 – синій) та одночасно позначені символами відповідно до істинної мітки (benign – крапки, outlier – хрестики).

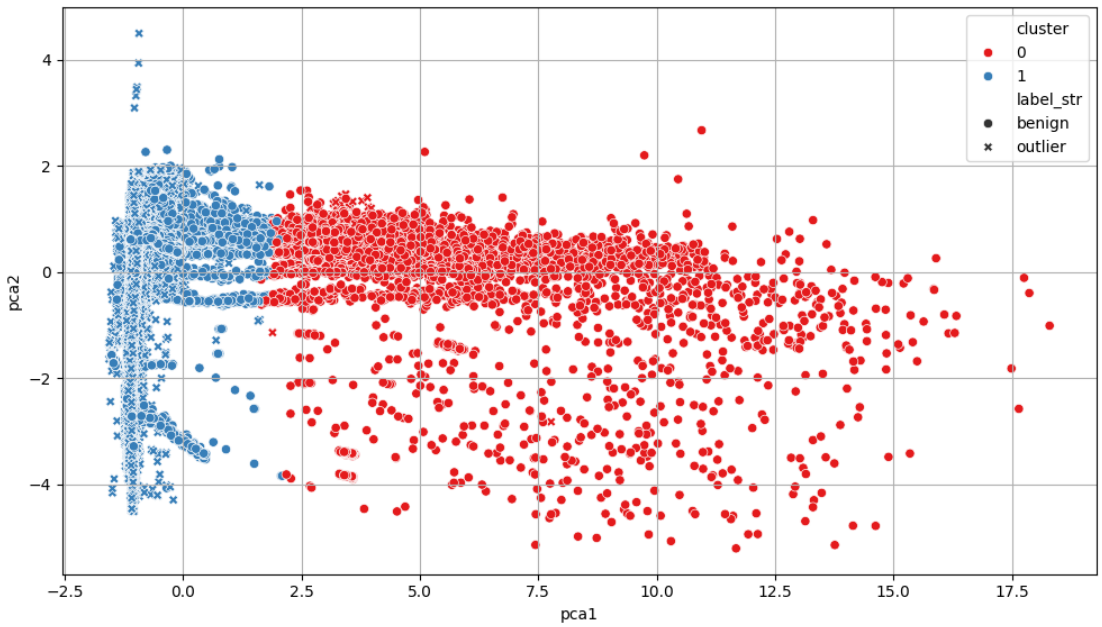

Рис. 2.10. Кластери KMeans у 2D PCA-просторі

З рисунка видно, що обидва кластери формують чітко відокремлені скупчення у просторі PCA. Синій кластер (1) здебільшого відповідає нормальному трафіку (benign), а червоний кластер (0) містить значну частину аномальних записів (outlier), розміщених у більш розпорошеній зоні. Водночас певна кількість аномалій проникає і в область, що переважає нормальними потоками, що може свідчити про їхню маскування або перехідні

характеристики. Це підтверджує наявність внутрішньої структури в даних, що дозволяє розділити їх на змістовні групи навіть без використання міток.

Отже, датасет LUFlow має добре виражену кластерну структуру з наявністю відмінностей між нормальними та аномальними потоками. Це підтверджує доцільність його використання для задач побудови моделей виявлення вторгнень на основі класифікації та кластерного аналізу, оскільки дані містять інформативні ознаки, що забезпечують можливість відокремлення різних типів трафіку у багатовимірному просторі.

Висновки до розділу 2

У рамках даного розділу було здійснено поглиблений аналіз методів виявлення та попередження вторгнень, що дозволило закласти основу для розробки ефективної IDS/IPS-системи. Розглянуто та систематизовано сигнатурні, аномальні та гібридні підходи до виявлення атак, проведено їх порівняльну характеристику, що дозволило оцінити особливості застосування кожного з них залежно від типу загроз. Детально проаналізовано машинні методи класифікації – метод швидкого дерева, метод опорних векторів та наївний байєсівський класифікатор – із подальшим вибором оптимального варіанту на основі критеріїв продуктивності, швидкодії та масштабованості; у результаті обґрунтовано доцільність застосування методу ШД.

Окрему увагу приділено обґрунтуванню вибору набору даних LUFlow, який надає розмітку потокового мережевого трафіку з урахуванням актуальних загроз. Проведено аналіз його атрибутів і здійснено багатовимірне дослідження ознак: побудовано теплову карту кореляції, вивчено розподіли тривалості, обсягів трафіку, кількості записів за класами, попарні залежності та кластерну структуру в просторі головних компонент. Отримані результати дозволяють сформувати інформативний простір ознак та підтверджують наявність відмінностей між нормальним і аномальним трафіком, що забезпечить високу якість подальшого навчання моделі.

Таким чином, у процесі виконання завдань цього розділу було створено концептуальне, алгоритмічне та емпіричне підґрунтя для розробки програмної реалізації системи виявлення та попередження вторгнень, що буде реалізовано в наступному розділі.

РОЗДІЛ 3 РОЗРОБКА ТА ТЕСТУВАННЯ СИСТЕМИ ВИЯВЛЕННЯ ТА ПОПЕРЕДЖЕННЯ ВТОРГНЕНЬ (IDS/IPS)

3.1 Проектування системи виявлення та попередження вторгнень

Під час проєктування системи виявлення та попередження вторгнень (IDS/IPS) особливу увагу необхідно приділити побудові архітектури, яка б забезпечувала масштабованість, надійність та можливість подальшого розширення функціоналу. Зважаючи на те, що система повинна одночасно обробляти потоки даних, виконувати аналіз трафіку за допомогою машинного навчання та забезпечувати зручний інтерфейс для взаємодії з користувачем, важливо дотримуватись структурного розподілу обов'язків між її компонентами. Такий підхід дозволяє зменшити внутрішню залежність між підсистемами та спростити підтримку системи в цілому.

У зв'язку з цим доцільним є використання трьохрівневої архітектури, яка є однією з найпоширеніших моделей у побудові сучасних інформаційних систем безпеки [50]. На першому рівні (рівень презентації) забезпечується взаємодія з користувачем, зокрема – візуалізація результатів аналізу та ініціювання перевірок трафіку. Другий рівень (логічний рівень) виконує основну бізнес-логіку, обробку даних, ініціалізацію моделі машинного навчання та ухвалення рішень щодо потенційних загроз. Третій рівень (рівень зберігання даних) відповідає за взаємодію з базою даних, збереження телеметрії, журналів перевірок, параметрів моделі та ін. Такий розподіл дозволяє розмежувати зони відповідальності, забезпечує масштабованість за кожним із напрямів та дає можливість гнучко інтегрувати систему в наявну IT-інфраструктуру організації.

На етапі реалізації рівня доступу до даних важливо забезпечити чітку структурованість і розмежування обов'язків між класами, які працюють з базою даних. Для цього була спроєктована відповідна об'єктна модель, яка представлена на рис. 3.1 у вигляді діаграми класів рівня даних. Вона охоплює основні сутності системи виявлення та попередження вторгнень, а також класи-

провайдери, що реалізують взаємодію з базою даних і забезпечують доступ до збереженої інформації про користувачів, моделі, логи, мережеві потоки та результати прогнозування.

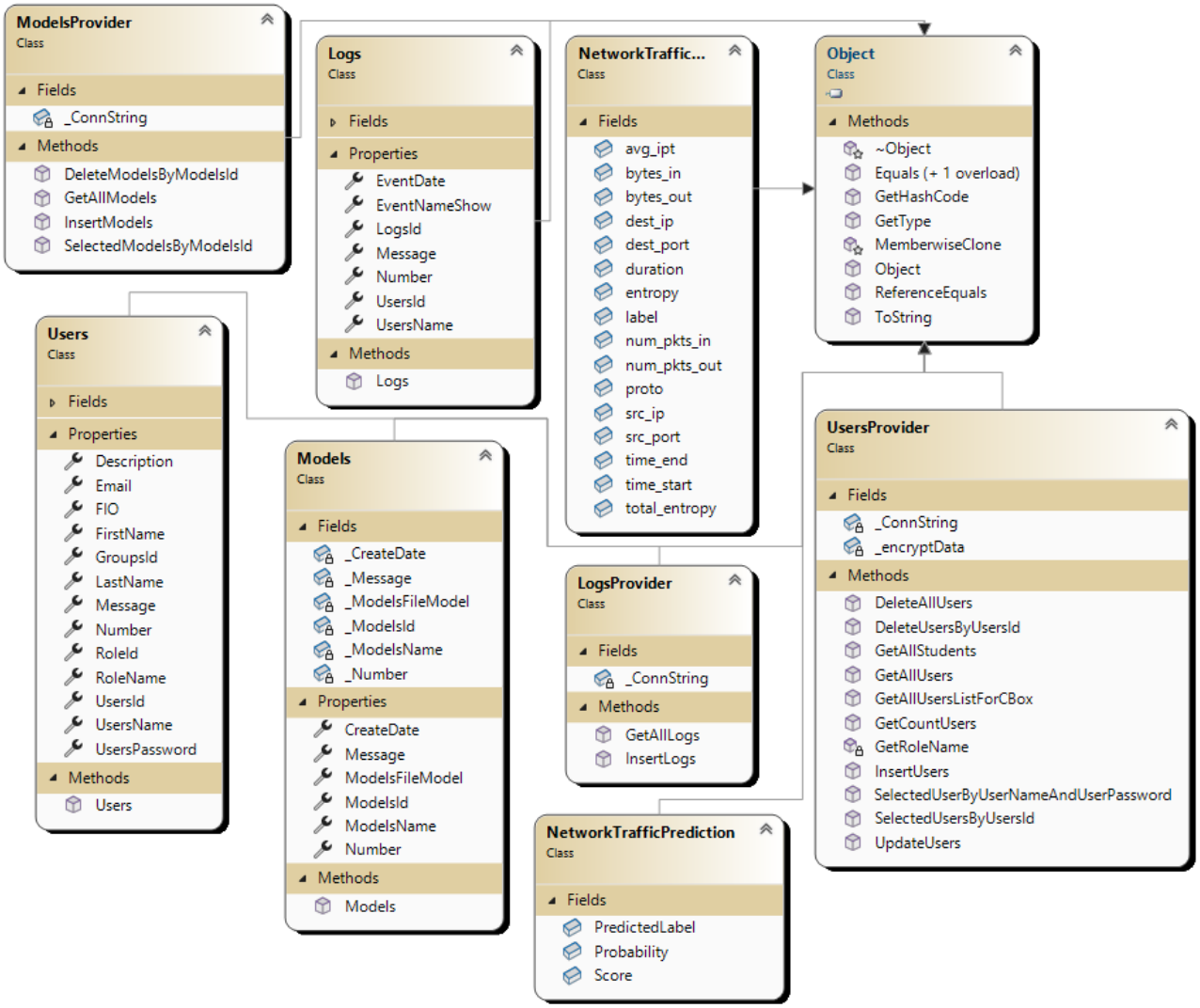

Рис. 3.1. Діаграма класів рівня даних

Діаграма класів рівня даних складається з:

– класу UsersProvider, який забезпечує повноцінну обробку інформації про користувачів. Він містить методи для додавання, оновлення, видалення, авторизації користувачів, а також пошуку за іменем або ідентифікатором. Крім того, в ньому реалізовано внутрішнє шифрування паролів;

– класу ModelsProvider, який відповідає за доступ до збережених моделей машинного навчання. Він дозволяє отримувати перелік доступних моделей, здійснювати їх вибірку за ідентифікатором, додавати нові або видаляти непотрібні моделі;

– класу LogsProvider, який керує журналами подій системи. Він реалізує функції запису логів у базу та отримання історії подій для подальшого аудиту або відображення в інтерфейсі;

– класу Users, що описує структуру користувача: ім’я, прізвище, email, роль, логін, пароль тощо. Він виступає моделлю даних для таблиці користувачів у базі;

– класу Models, який слугує відображенням збережених ML-моделей: включає назву, дату створення, стисле повідомлення, тип моделі та її представлення у вигляді файлу;

– класу Logs, що є об’єктною репрезентацією записів подій у системі: описує тип події, повідомлення, дату, користувача та ідентифікатор події;

– класу NetworkTrafficData, який описує потік мережевого трафіку за такими параметрами, як IP-адреси, порти, протокол, кількість пакетів, ентропія, середній інтервал між пакетами, тривалість та інші технічні характеристики. Цей клас використовується для навчання та прогнозування;

– класу NetworkTrafficPrediction, який містить результати класифікації моделі: передбачений клас (PredictedLabel), ймовірність та оціночний бал (Score).

Загалом, така архітектура рівня даних дозволяє організувати ефективну, безпечну та масштабовану взаємодію з базою даних, що є критично важливим для системи виявлення вторгнень з високими вимогами до швидкодії та достовірності збережених подій.

У процесі розробки інтерфейсу користувача для системи виявлення та попередження вторгнень (IDS/IPS) було спроєктовано набір форм, що охоплюють усі функціональні аспекти взаємодії з користувачем. Ці форми реалізують логіку управління користувачами, роботу з моделями, моніторинг активності, перегляд журналів подій, а також забезпечують персоналізацію та авторизацію доступу. На рис. 3.2 подано діаграму класів рівня користувацького інтерфейсу, яка демонструє структуру форм, їх взаємозв’язки та основні методи.

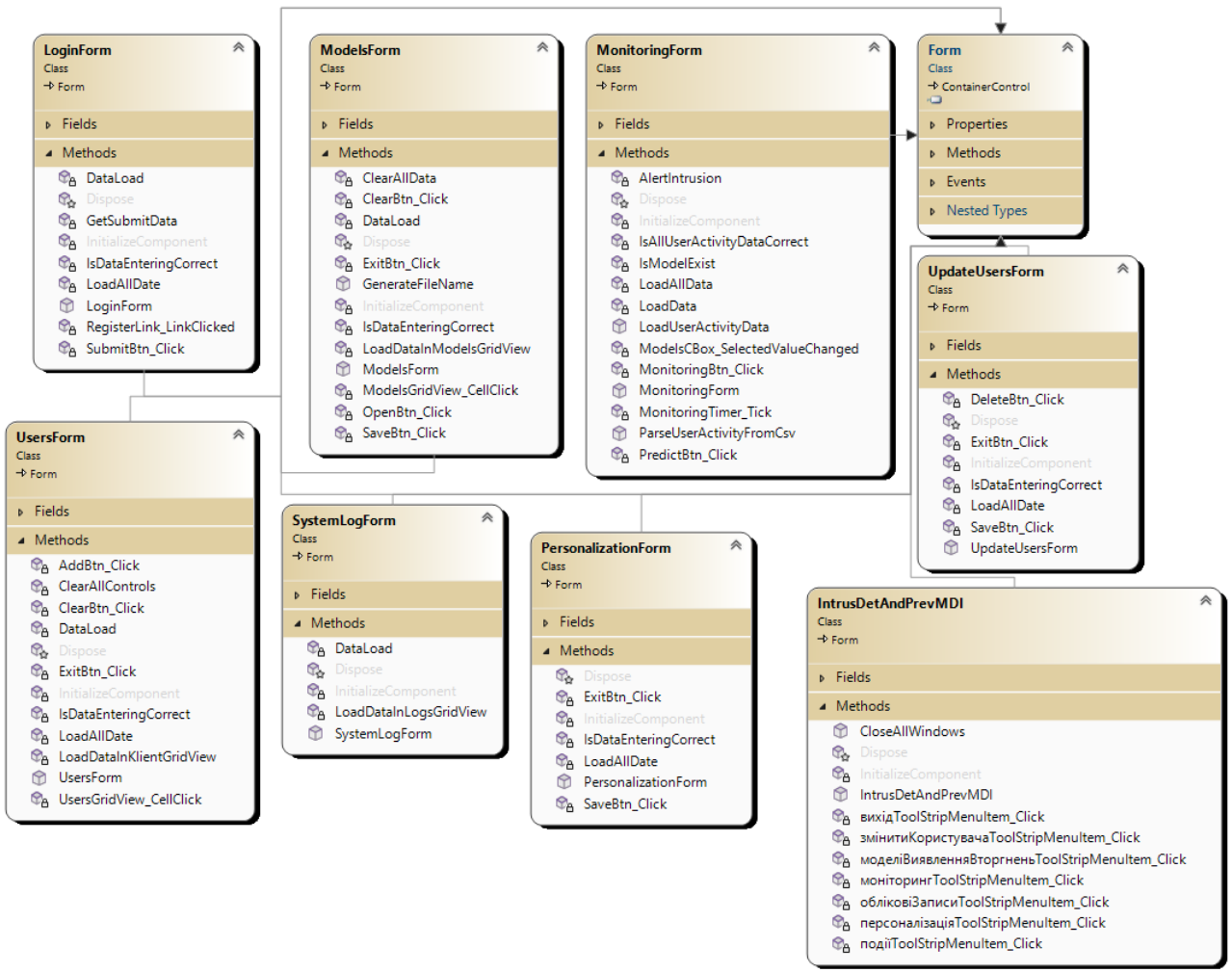

Рис. 3.2. Діаграма класів рівня користувацького інтерфейсу

Діаграма класів рівня користувацького інтерфейсу складається з:

– класу LoginForm, що відповідає за автентифікацію користувача. Реалізує перевірку правильності введених облікових даних, ініціалізацію сесії користувача, а також перехід до реєстраційної форми за допомогою елемента RegisterLink_LinkClicked;

– класу UsersForm, призначеного для управління обліковими записами користувачів. Забезпечує додавання, редагування, видалення, виведення списку користувачів та обробку подій натискання на таблицю даних (UsersGridView_CellClick);

– класу UpdateUsersForm, який реалізує функціональність оновлення даних окремого користувача. Забезпечує заповнення форми на основі поточного запису, валідацію введених даних та збереження змін у базу;

– класу MonitoringForm, що виконує роль основної форми моніторингу активності. Забезпечує завантаження даних користувацької активності, запуск таймера, обробку подій прогнозування (PredictBtn_Click) і реалізацію алертів при виявленні вторгнення (AlertIntrusion);

– класу ModelForm, який дає змогу працювати з моделями машинного навчання. Реалізує завантаження, вибір, збереження, очищення, відкриття файлів моделей, а також обробку взаємодії з таблицею моделей (ModelsGridView_CellClick);

– класу SystemLogForm, який дозволяє переглядати системні події та історію дій користувачів. Забезпечує завантаження логів у відповідний компонент інтерфейсу для аналітичного перегляду (LoadDataInLogsGridView);

– класу PersonalizationForm, який надає можливість персоналізувати інтерфейс користувача. Реалізує налаштування профілю, збереження змін та вихід з форми;

– класу IntrusDetAndPrevMDI, що є основною MDI-формою системи. Забезпечує навігацію по пунктах меню, відкриття внутрішніх форм системи, закриття вікон, виклик форм користувачів, моніторингу, логів та модулів;

– класу Form, який є базовим контейнерним класом WinForms, від якого успадковуються всі інші форми. Він визначає базові властивості, події та методи для віконної взаємодії в інтерфейсі.

Узагальнюючи, рівень користувацького інтерфейсу системи побудований на чіткому розмежуванні функціоналу за відповідними формами, що забезпечує зрозумілу, модульну та масштабовану архітектуру. Це дозволяє системі легко адаптуватися до змін, а також забезпечує інтуїтивну взаємодію з користувачем.

На етапі розробки логіки обробки даних та реалізації правил бізнес-процесів системи виявлення та попередження вторгнень було сформовано відповідний рівень бізнес-логіки. Саме на цьому рівні зосереджена перевірка правильності введених даних, реалізація шифрування, визначення доступу за ролями, а також інтерфейсне забезпечення повідомлень і стандартів обробки.

На рис. 3.3 наведено діаграму класів рівня бізнес-логіки, яка відображає основні об’єкти, їх взаємозв’язки та відповідні методи.

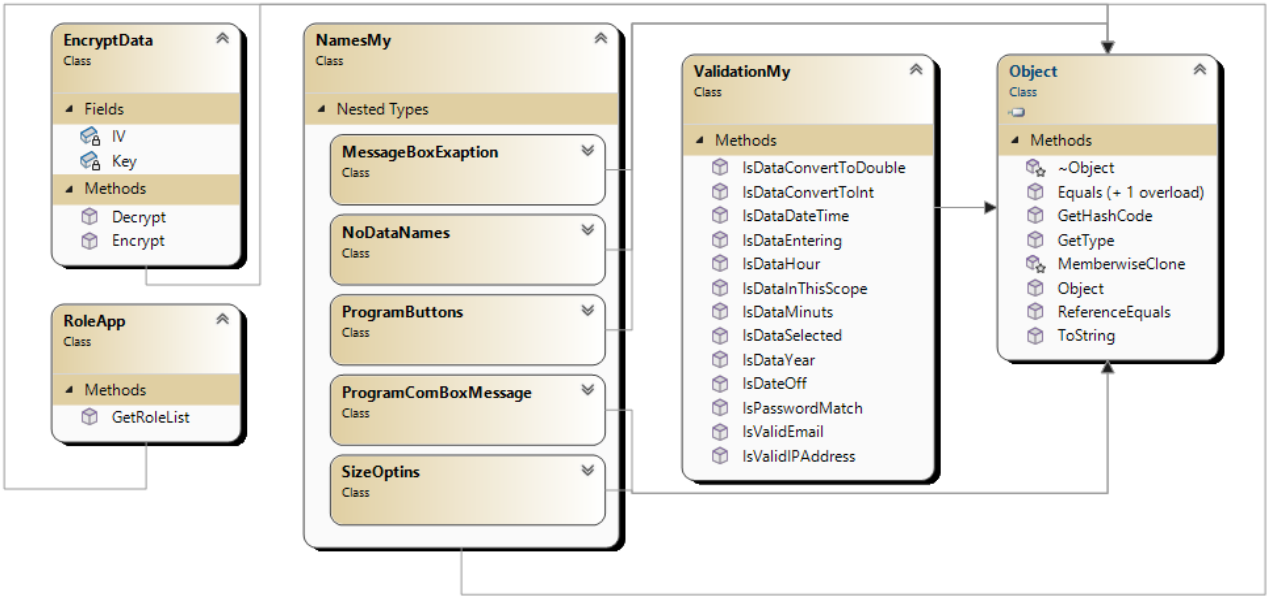

Рис. 3.3. Діаграма класів рівня бізнес-логіки

Діаграма класів рівня бізнес-логіки складається з:

– класу EncryptData, який відповідає за реалізацію механізмів шифрування та дешифрування даних у системі. Клас містить два основні поля IV та Key, які використовуються як ініціалізаційний вектор і ключ шифрування відповідно, а також методи Encrypt та Decrypt для забезпечення конфіденційності при збереженні або передачі даних;

– класу RoleApp, призначеного для роботи з ролями користувачів. Він забезпечує метод GetRoleList, який дозволяє отримати список можливих ролей, необхідних для реалізації авторизації та обмеження доступу до функцій системи;

– класу NamesMy, що об'єднує в собі кілька вкладених класів, пов’язаних з іменуванням і візуальним представленням інтерфейсних елементів. Зокрема, MessageBoxExaption містить шаблони повідомлень про помилки, NoDataNames – стандартні тексти про відсутність даних, ProgramButtons – імена для кнопок, ProgramComBoxMessage – для повідомлень у випадаючих списках, а SizeOptins – параметри розмірів для інтерфейсу;

– класу ValidationMy, який є центральним елементом перевірки вхідних даних на всіх рівнях роботи з формами. Він включає набір методів перевірки коректності: числових, текстових, часових, адресних та логічних значень, таких як IsValidEmail, IsValidIPAddress, IsPasswordMatch, IsDateOff, IsDataEntering та інші.

У підсумку, рівень бізнес-логіки системи забезпечує модульну та багатоаспектну обробку даних, від перевірки введеної інформації до реалізації механізмів захисту та інтерфейсних повідомлень. Завдяки чіткому розділенню відповідальностей між класами досягається гнучкість розширення, зручність супроводу та відповідність стандартам безпечного програмування.

3.2 Завантаження та обробка датасету для тренування моделі

Завантаження та обробка датасету для тренування моделі розпочато з імпорту CSV-файлу, що містив потоки мережевого трафіку з відповідними мітками класів. Проведено валідацію кожного запису датасету: перевірено відповідність типів даних, цілісність записів, відсутність порожніх значень у критичних полях. У разі виявлення нечислових або аномально відсутніх значень останні видалялися або замінювалися на агреговані величини – середнє або медіану залежно від природи розподілу.

Виконано очистку набору від зайвих та надлишково корельованих ознак. Зокрема, вилучено часові ознаки time_start та time_end, які не несуть суттєвої дискримінативної інформації для класифікації. На основі теплової карти кореляцій залишено лише інформативні атрибути: avg_ipt, bytes_in, bytes_out, entropy, num_pkts_in, num_pkts_out, proto, total_entropy, duration. Ці характеристики демонструють відносно помірну взаємну кореляцію та здатні відображати внутрішні закономірності в потоках трафіку.

Проведено балансування класів з метою запобігання зміщенню моделі в бік частішого класу. Вихідна кількість прикладів становила 294593 для класу benign та 247023 для outlier. Для забезпечення рівноваги виконано випадкову вибірку з класу benign до рівня класу outlier, внаслідок чого було отримано

однакову кількість записів для кожної категорії – по 247023. Це дозволяє уникнути домінування одного класу над іншим під час навчання.

У фінальній фазі попередньої обробки виконано нормалізацію числових ознак із застосуванням масштабування, що забезпечує однакову вагу різним за розмірністю характеристикам. Таким чином, сформовано очищений, збалансований і готовий до навчання датасет, який містить узгоджений набір інформативних параметрів для ефективного застосування методів машинного навчання.

3.3 Реалізація системи виявлення та попередження вторгнень на базі обраного методу

Реалізація системи виявлення та попередження вторгнень здійснюється з використанням раніше обґрунтованого методу швидке дерево, який добре зарекомендував себе в задачах класифікації мережевого трафіку. Для цього у середовищі C# із використанням ML.NET створено відповідні класи, що дозволяють коректно завантажувати, інтерпретувати та передавати дані в модель.

На рис.. 3.4 подано структуру класів NetworkTrafficData і NetworkTrafficPrediction, які відповідають за представлення ознак вхідного потоку та результатів прогнозування відповідно.

```
public class NetworkTrafficData {
    [LoadColumn(0)]
    public float avg_ipt;

    [LoadColumn(1)]
    public float bytes_in;

    [LoadColumn(2)]
    public float bytes_out;

    [LoadColumn(3)]
    public string dest_ip;

    [LoadColumn(4)]
    public string dest_port;

    [LoadColumn(5)]
    public float entropy;

    [LoadColumn(6)]
    public float num_pkts_out;

    [LoadColumn(7)]
    public float num_pkts_in;

    [LoadColumn(7)]
    public float num_pkts_in;

    [LoadColumn(8)]
    public string proto;

    [LoadColumn(9)]
    public string src_ip;

    [LoadColumn(10)]
    public string src_port;

    [LoadColumn(11)]
    public string time_end;

    [LoadColumn(12)]
    public string time_start;

    [LoadColumn(13)]
    public float total_entropy;

    [LoadColumn(14)]
    public string label;

    [LoadColumn(15)]
    public float duration;
}

3 references
public class NetworkTrafficPrediction {
    [ColumnName("PredictedLabel")]
    public bool PredictedLabel;

    public float Probability;
    public float Score;
}
```

Рис. 3.4. Вхідний та вихідний класи моделі

Вхідний клас містить поля, що відображають числові та текстові характеристики мережевого трафіку, наприклад, середній інтервал між пакетами, кількість байтів та пакетів, ентропію, IP-адреси і порти, мітку класу та інші. Другий клас містить поля, що відображають результат класифікації – передбачене значення, ймовірність належності до певного класу та зважений бал. Така структура забезпечує ефективне зчитування і передавання інформації між фазами завантаження даних, прогнозування та інтерпретації результатів.

Фагмент коду на рис. 3.5 реалізує обробку події натискання на кнопку відкриття тренувального файлу у графічному інтерфейсі Windows Forms. При натисканні створюється об’єкт діалогового вікна OpenFileDialog, який обмежує типи файлів для вибору – зокрема CSV-файли або всі файли загалом. Якщо користувач підтверджує вибір файлу, шлях до цього файлу зберігається у змінну _Path, а також відображається у текстовому полі FileNameTBox, що дозволяє користувачеві побачити, який саме файл було завантажено. Така функціональність забезпечує зручний механізм вибору файлу з даними для подальшої обробки та аналізу.

```
private void OpenBtn_Click(object sender, EventArgs e) {
  // Створення діалогового вікна для відкриття файлу
  OpenFileDialog openFileDialog = new OpenFileDialog {
    Filter = "Text files (*.csv)|*.csv|All files (*.*)|*.*",
    FilterIndex = 2,
    RestoreDirectory = true
  };
  if (openFileDialog.ShowDialog() == DialogResult.OK) {
    _Path = openFileDialog.FileName;
    FileNameTBox.Text = openFileDialog.FileName;
```

Рис. 3.5. Код для відкриття тренувального файлу

У наведеному на рис. 3.6 фрагменті коду виконується ініціалізація середовища машинного навчання за допомогою об’єкта MLContext, якому задається фіксоване зерно випадковості для забезпечення відтворюваності результатів. Далі здійснюється завантаження даних із вказаного CSV-файлу, шлях до якого зберігається у змінній _Path. Файл завантажується у вигляді структури IDataView, при цьому вказано, що перший рядок містить заголовки, а розділювачем колонок є кома. Після завершення завантаження у текстове поле RaportTBox виводиться повідомлення про успішне зчитування даних, а метод

Application.DoEvents дозволяє оновити інтерфейс у реальному часі для відображення цього повідомлення.

```
mlContext = new MLContext(seed: 1);
// Завантаження даних
dataView = mlContext.Data.LoadFromTextFile<NetworkTrafficData>(
    path: _Path,
    hasHeader: true,
    separatorChar: ',');
RaportTBox.Text = "Завантаження даних\r\n";
Application.DoEvents();
```

Рис. 3.6. Ініціалізація середовища машинного навчання

На рис. 3.7 наведено код, у якому визначається масив назв числових ознак, які будуть використовуватись для подальшої побудови моделі. До таких ознак належать інтервал між пакетами, кількість переданих байтів і пакетів, ентропія та тривалість з'єднання.

```
string[] numericColumns = {
            "avg_ipt", "bytes_in", "bytes_out", "entropy",
            "num_pkts_out", "num_pkts_in", "total_entropy", "duration"
        };
// Виведення кількості взірців для кожного класу
var labelCounts =
  mlContext.Data.CreateEnumerable<NetworkTrafficData>(dataView, reuseRowObject: false)
    .GroupBy(data => data.label)
    .Select(group => new { Label = group.Key, Count = group.Count() });
RaportTBox.Text += "Кількість взірців:\r\n";
foreach (var count in labelCounts) {
  RaportTBox.Text += ($"Клас: {count.Label}, Кількість: {count.Count}\r\n");
}
```

Рис. 3.7. Визначення масиву назв числових ознак та підрахунок кількості прикладів для кожного класу мітки

Після цього виконується підрахунок кількості прикладів для кожного класу мітки. Для цього всі записи зчитуються з IDataView у вигляді переліку об'єктів типу NetworkTrafficData, після чого групуються за полем label. Для кожної групи визначається кількість записів, і ці дані виводяться у текстове поле RaportTBox у вигляді переліку класів та відповідної кількості прикладів. Це дає змогу швидко оцінити розподіл даних за класами ще до етапу тренування моделі.

Фрагмент коду на рис. 3.8 реалізує побудову конвеєра попередньої обробки даних для задачі бінарної класифікації в ML.NET. На першому етапі відбувається перетворення текстових міток класу label у булевий формат, де клас benign інтерпретується як false, а outlier – як true, що відповідає формату

цільової змінної для класифікатора. Далі виконується кодування категоріальних ознак: для протоколу (proto) застосовується метод one-hot encoding, а для IP-адреси призначення (dest_ip) використано хеш-кодування з обмеженою кількістю бітів, що зменшує розмірність. Після цього всі числові ознаки об'єднуються в єдиний вектор ознак Features, який нормалізується за допомогою приведення до нульового середнього та одиничного стандартного відхилення. Така послідовність кроків дозволяє адаптувати сирі дані до формату, необхідного для ефективного навчання моделі.

```
var dataProcessPipeline = mlContext.Transforms.Conversion.MapValue(
        outputColumnName: "Label",
        inputColumnName: "label",
        keyValuePairs: new[] {
                    new KeyValuePair<string, bool>("benign", false),
                    new KeyValuePair<string, bool>("outlier", true)
        })
    .Append(mlContext.Transforms.Categorical.OneHotEncoding("ProtoEncoded", "proto"))
    .Append(mlContext.Transforms.Categorical.OneHotHashEncoding("DestIpEncoded",
    "dest_ip", numberOfBits: 10))
    .Append(mlContext.Transforms.Concatenate("Features", numericColumns))
    .Append(mlContext.Transforms.NormalizeMeanVariance("Features"));
```

Рис. 3.8. Побудова конвеєра попередньої обробки даних

У коді на рис. 3.9 виконується вибір алгоритму машинного навчання для задачі бінарної класифікації. В якості тренера використовується метод FastTree – швидка реалізація градієнтного бустингу на базі дерев рішень, який працює з колонками ознак Features та міткою Label. Далі конвеєр попередньої обробки, сформований раніше, об'єднується з вибраним тренером у загальний навчальний пайплайн. Після цього виконується розділення всього датасету на тренувальну та тестову вибірки у співвідношенні 80% до 20%, що дозволяє оцінити якість моделі на незалежних даних. Отримані підмножини даних використовуються для навчання моделі та її подальшої валідації.

```
var trainer = mlContext.BinaryClassification.Trainers.FastTree(
    labelColumnName: "Label",
    featureColumnName: "Features");

var trainingPipeline = dataProcessPipeline.Append(trainer);
// Розділення даних
var split = mlContext.Data.TrainTestSplit(dataView, testFraction: 0.2);
var trainData = split.TrainSet;
var testData = split.TestSet;
```

Рис. 3.9. Вибір алгоритму машинного навчання та розподіл даних

Фрагмент коду на рис. 3.10 відповідає за етап навчання та оцінювання моделі класифікації. На основі тренувального набору даних виконується навчання моделі за допомогою методу Fit, у результаті чого формується об’єкт trainedModel, який містить усі параметри моделі, адаптовані до вхідних даних. Далі модель застосовується до тестової вибірки, в результаті чого створюється новий набір з прогнозами. Отримані прогнози передаються до методу Evaluate, який обчислює ключові метрики якості класифікації, зокрема точність, повноту, F-міру, AUC тощо. Це дозволяє об’єктивно оцінити ефективність навченої моделі до її розгортання в системі.

```
trainedModel = trainingPipeline.Fit(trainData);

// Оцінювання моделі
var predictions = trainedModel.Transform(testData);
var metrics =
  mlContext.BinaryClassification.Evaluate(predictions, labelColumnName: "Label");
```

Рис. 3.10. Навчання та оцінювання моделі

Код на рис. 3.11 реалізує виведення результатів оцінювання навченої моделі у текстове поле.

```
RaportTBox.Text += "=== Матриця помилок ===\r\n";
var scoredData =
  mlContext.Data.CreateEnumerable<NetworkTrafficPrediction>(predictions,
  reuseRowObject: false).ToList();
var truePositives = scoredData.Count(p => p.PredictedLabel && p.Probability >= 0.5);
var trueNegatives = scoredData.Count(p => !p.PredictedLabel && p.Probability < 0.5);
var falsePositives = scoredData.Count(p => p.PredictedLabel && p.Probability < 0.5);
var falseNegatives = scoredData.Count(p => !p.PredictedLabel && p.Probability >= 0.5);
RaportTBox.Text += ($"True Positives: {truePositives}\r\n");
RaportTBox.Text += ($"True Negatives: {trueNegatives}\r\n");
RaportTBox.Text += ($"False Positives: {falsePositives}\r\n");
RaportTBox.Text += ($"False Negatives: {falseNegatives}\r\n");
// Виведення метрик для кожного класу
RaportTBox.Text += "=== Метрики для кожного класу ===\r\n";
RaportTBox.Text += "Позитивний клас (1):\r\n";
RaportTBox.Text += ($"Precision: {metrics.PositivePrecision:P2}\r\n");
RaportTBox.Text += ($"Recall: {metrics.PositiveRecall:P2}\r\n");
RaportTBox.Text += "Негативний клас (0):\r\n";
RaportTBox.Text += ($"Precision: {metrics.NegativePrecision:P2}\r\n");
RaportTBox.Text += ($"Recall: {metrics.NegativeRecall:P2}\r\n");
```

Рис. 3.11. Виведення результатів оцінювання навченої моделі

На початку відображаються загальні метрики, зокрема точність (accuracy), площа під ROC-кривою (AUC) та F1-метрика – усі у відсотковому форматі з точністю до двох знаків. Далі виконується формування та виведення матриці помилок, яка обчислюється вручну на основі списку передбачень: визначається кількість істинно позитивних, істинно негативних,

хибнопозитивних та хибнонегативних випадків відповідно до значення ймовірності. Завершується блок обрахунком та виведенням точності (precision) і повноти (recall) для кожного з класів окремо – позитивного та негативного. Такий підхід дозволяє здійснити комплексну оцінку моделі та проаналізувати її здатність розрізняти класи в задачі виявлення аномалій.

Метод відповідає за збереження натренованої моделі після перевірки правильності введених даних. Якщо всі вхідні дані коректні, формується унікальний шлях для збереження файлу моделі у форматі .zip, який міститиметься у підкаталозі teach локального каталогу проєкту.

```
private void SaveBtn_Click(object sender, EventArgs e) {
  if (IsDataEnteringCorrect()) {
    //Зберігання моделі
    string pathName = @"\teach\" + GenerateFileName() + ".zip";
    string localProj =
      System.IO.Path.GetDirectoryName(System.Reflection.Assembly.GetExecutingAssembly().Location);
    _ModelsProvider.InsertModels(ModelsNamesTBox.Text, pathName);
    mlContext.Model.Save(trainedModel, dataView.Schema, localProj + pathName);
    ClearAllData();
    _LogsProvider.InsertLogs(LoginForm.CurrentUser.UsersId,
      "Було навчено модель " +
      ModelsNamesTBox.Text, DateTime.Now);
    MessageBox.Show("Дані успішно збережено!");
  }
}
```

Рис. 3.12. Збереження натренованої моделі

Назва файлу генерується автоматично, що дозволяє уникнути конфліктів імен. Далі модель зберігається на диск разом зі схемою даних, а також додається відповідний запис до бази даних моделей із зазначенням назви моделі. Після цього всі поля очищуються, а до журналу подій вноситься інформація про виконання операції, включаючи час і користувача. У фіналі користувач отримує сповіщення про успішне збереження моделі у вигляді повідомлення у вікні.

У методі на рис. 3.13 реалізується завантаження збереженої моделі машинного навчання з локального диску та ініціалізація прогнозного механізму. На першому кроці формується повний шлях до файлу моделі, використовуючи кореневу директорію застосунку та відносний шлях, переданий у параметрі. Далі виконується завантаження моделі разом із її схемою ознак, яка автоматично визначається під час зчитування. Після цього

створюється об’єкт predictionEngine, що забезпечує можливість здійснювати одиничні передбачення на основі класу вхідних даних NetworkTrafficData та типу результату NetworkTrafficPrediction. Така логіка дозволяє інтегрувати збережену модель у процес аналізу нових даних без повторного навчання.

```
private void LoadData(string FilePath) {
  string localProj = Application.StartupPath + FilePath;
  // Визначення DataViewSchema для конвеєра
  // підготовки даних і навченої моделі
  DataViewSchema modelSchema;
  // Завантаження моделі
  ITransformer model = mlContext.Model.Load(localProj,
    out modelSchema);
  // Створення механізму прогнозування
  predictionEngine =
      mlContext.Model.CreatePredictionEngine<NetworkTrafficData,
      NetworkTrafficPrediction>(model);
}
```

Рис. 3.13. Завантаження збереженої моделі машинного навчання

Метод PredictBtn_Click призначений для перевірки роботи класифікаційної моделі в системі виявлення вторгнень шляхом прогнозування типу трафіку на основі даних, введених користувачем у графічній формі (рис. 3.14). Його виконання активується натисканням кнопки прогнозування, після чого модель аналізує введені значення і повертає відповідний результат.

```
private void PredictBtn_Click(object sender, EventArgs e) {
  // Перевірка коректності введених даних
  if (IsAllUserActivityDataCorrect() && IsModelExist()) {
    // Створення об'єкта NetworkTrafficData з даними з введених значень
    NetworkTrafficData testUserActivity = new NetworkTrafficData {
      avg_ipt = float.Parse(AvgIptTBox.Text),
      bytes_in = float.Parse(BytesInTBox.Text),
      bytes_out = float.Parse(BytesOutTBox.Text),
      dest_ip = DestIpTBox.Text,
      dest_port = DestPortTBox.Text,
      entropy = float.Parse(EntropyTBox.Text),
      num_pkts_out = float.Parse(NumPktsOutTBox.Text),
      num_pkts_in = float.Parse(NumPktsInTBox.Text),
      proto = ProtoTBox.Text,
      src_ip = SrcIpTBox.Text,
      src_port = SrcPortTBox.Text,
      label = "unknown", // Значення прогнозується моделлю
      total_entropy = float.Parse(TotalEntropyTBox.Text),
      duration = float.Parse(DurationTBox.Text)
    };
```

Рис. 3.14. Фрагмент методу для перевірки моделі

На початку методу перевіряється коректність заповнення всіх полів і наявність завантаженої моделі. Якщо ці умови виконуються, створюється екземпляр класу NetworkTrafficData, куди передаються значення із відповідних полів інтерфейсу, таких як кількість байтів, ентропія, кількість пакетів,

протоколи, адреси, порти та інші характеристики мережевого потоку. Поле мітки встановлюється як "unknown", оскільки передбачається, що саме модель визначить тип цього трафіку. Цей об’єкт слугує вхідними даними для функції прогнозування, яка працює на основі навченої моделі.

У фрагменті коду на рис. 3.15 здійснюється формування текстового повідомлення з усіма параметрами, які були введені користувачем для проведення прогнозу. Створюється об’єкт типу StringBuilder, до якого по черзі додаються рядки з назвами ознак та відповідними значеннями з об’єкта testUserActivity. Зокрема, виводяться показники мережевого трафіку, такі як середній інтервал між пакетами, кількість вхідних і вихідних байтів і пакетів, ентропійні характеристики, IP-адреси та порти джерела і призначення, протокол та тривалість з’єднання. Це забезпечує зручне представлення вхідних даних, що використовуються для аналізу моделі, та дозволяє користувачу перевірити їх перед здійсненням прогнозу.

```
var dataInfo = new StringBuilder();
dataInfo.AppendLine("Введені дані для прогнозування:");
dataInfo.AppendLine($"Середній інтервал: {testUserActivity.avg_ipt}");
dataInfo.AppendLine($"Вхідні байти: {testUserActivity.bytes_in}");
dataInfo.AppendLine($"Вихідні байти: {testUserActivity.bytes_out}");
dataInfo.AppendLine($"IP призначення: {testUserActivity.dest_ip}");
dataInfo.AppendLine($"Порт призначення: {testUserActivity.dest_port}");
dataInfo.AppendLine($"Ентропія: {testUserActivity.entropy}");
dataInfo.AppendLine($"Кількість вихідних пакетів: {testUserActivity.num_pkts_out}");
dataInfo.AppendLine($"Кількість вхідних пакетів: {testUserActivity.num_pkts_in}");
dataInfo.AppendLine($"Протокол: {testUserActivity.proto}");
dataInfo.AppendLine($"IP джерела: {testUserActivity.src_ip}");
dataInfo.AppendLine($"Порт джерела: {testUserActivity.src_port}");
dataInfo.AppendLine($"Загальна ентропія: {testUserActivity.total_entropy}");
dataInfo.AppendLine($"Тривалість: {testUserActivity.duration}");
```

Рис. 3.15. Формування текстового повідомлення з усіма параметрами

У цьому фрагменті коду на рис. 3.16 здійснюється процес прогнозування за допомогою попередньо навченої моделі. Об’єкт predictionEngine викликає метод Predict, якому передається сформований набір вхідних даних, і в результаті отримується об’єкт із передбаченням. Далі формується текстовий звіт, у якому зазначається, чи виявлено аномалію, а також відображається відповідна ймовірність її наявності у відсотковому форматі. Після цього в систему логування додається запис про те, що користувач провів операцію прогнозування для вибраної моделі, із фіксацією дати та часу. У фіналі

результат прогнозу додається до поля моніторингу інтерфейсу, що дозволяє користувачеві миттєво побачити результат аналізу введених параметрів.

```
var prediction = predictionEngine.Predict(testUserActivity);
// Формуємо результат прогнозування
var answer = new StringBuilder();
answer.AppendLine("\r\n--- Прогнозування ---");
answer.AppendLine($"\r\nЄ аномалія?: {(prediction.PredictedLabel ? "Так" : "Ні")}");
answer.AppendLine($"\r\nЙмовірність аномалії: {prediction.Probability:P2}");
// Логування та виведення результату
_LogsProvider.InsertLogs(LoginForm.CurrentUser.UsersId,
    "Було проведено прогнозування активності для моделі " + ModelsCBox.Text, DateTime.Now);
// Додавання результату прогнозування до текстового поля
MonitoringTBox.Text += answer.ToString();
```

Рис. 3.16. Процес прогнозування за допомогою попередньо навченої моделі

Розробка коду системи виявлення та попередження вторгнень охоплює повний цикл обробки даних – від завантаження і підготовки до прогнозування та виведення результатів у зручному для користувача вигляді. Структура реалізованих методів забезпечує тісну інтеграцію між навченою моделлю машинного навчання та графічним інтерфейсом, що дозволяє не лише автоматизувати аналіз мережевого трафіку, а й зробити процес прогнозування доступним для користувача без спеціальних знань у галузі штучного інтелекту. Застосування модульного підходу сприяє розширюваності системи, а прозорість відображення даних і результатів – покращенню взаємодії з кінцевим користувачем.

3.4 Тестування системи та її оцінка

Початковий етап тестування розробленої системи виявлення та попередження вторгнень передбачає оцінку її здатності класифікувати мережеву активність на основі попередньо навченої моделі машинного навчання. З цією метою було створено експериментальне середовище, що відтворює типові умови функціонування, включаючи різнорідний трафік, який містить як легітимні, так і потенційно небезпечні з'єднання. У якості апаратного забезпечення використано персональний комп’ютер із базовими технічними характеристиками, що дозволяє оцінити ефективність системи у звичайному користувацькому середовищі без застосування спеціалізованих серверних рішень.

Особливу увагу в межах цього етапу було приділено аналізу основних класифікаційних метрик, отриманих після завершення навчання моделі на основі попередньо підготовлених даних. Саме ці метрики дають змогу зробити попередню оцінку продуктивності системи, зокрема її точності, збалансованості та здатності відрізняти нормальний трафік від аномального. На рис. 3.17 подано результати тренування моделі, що відображають ключові показники її якості, такі як accuracy, F1-міра та AUC, які були обчислені під час тестування на відкладеній вибірці.

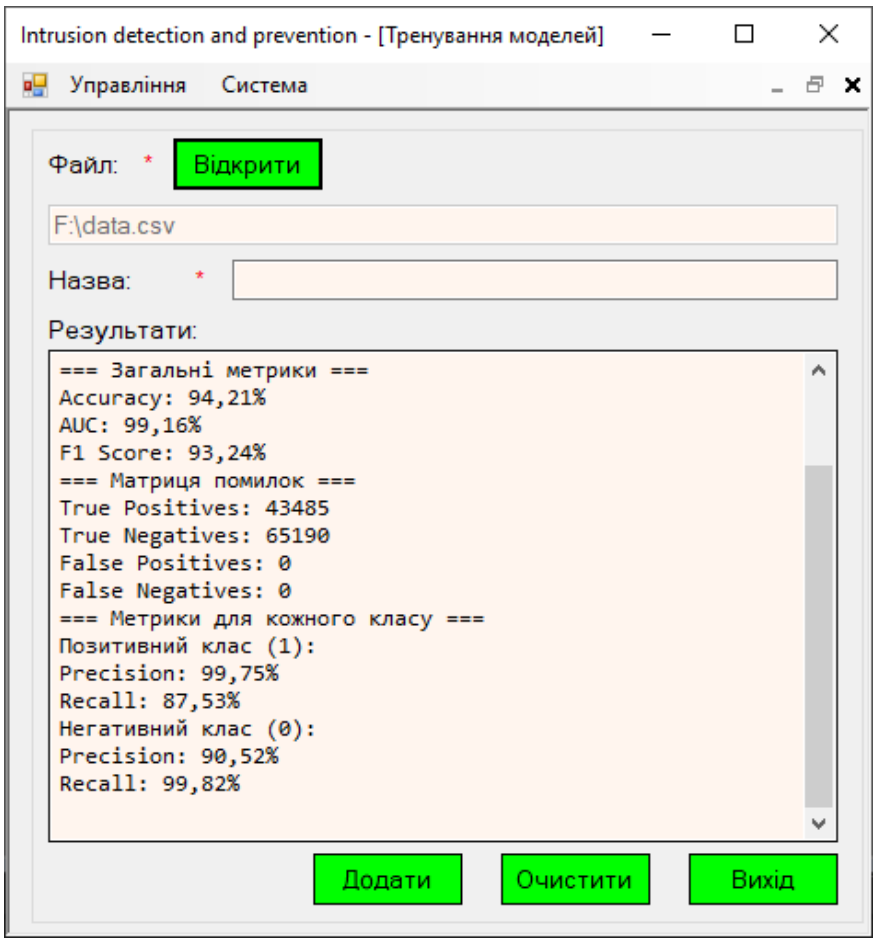

Рис. 3.17. Результат навчання моделі

Отримане значення accuracy склало 94,21%, що вказує на високий загальний рівень правильних класифікацій у задачі двокласового поділу. Значення F1-міри на рівні 93,24% демонструє добрий баланс між точністю та повнотою моделі, що є важливим у контексті виявлення як аномалій, так і нормального трафіку. Особливо варто відзначити AUC, яке досягло 99,16%, що свідчить про здатність моделі ефективно розділяти класи за широкого діапазону

порогів класифікації. Такий рівень результатів підтверджує надійність побудованої моделі в умовах тестового середовища.

Після попередньої оцінки загальних метрик точності, F1-міри та площі під ROC-кривою, доцільно проаналізувати специфічні характеристики моделі для кожного класу окремо. На рис. 3.18 подано значення точності (Precision) та повноти (Recall) для позитивного класу (аномалії) та негативного класу (нормальний трафік), що дає змогу глибше оцінити здатність системи до диференціації між безпечними та потенційно небезпечними з'єднаннями.

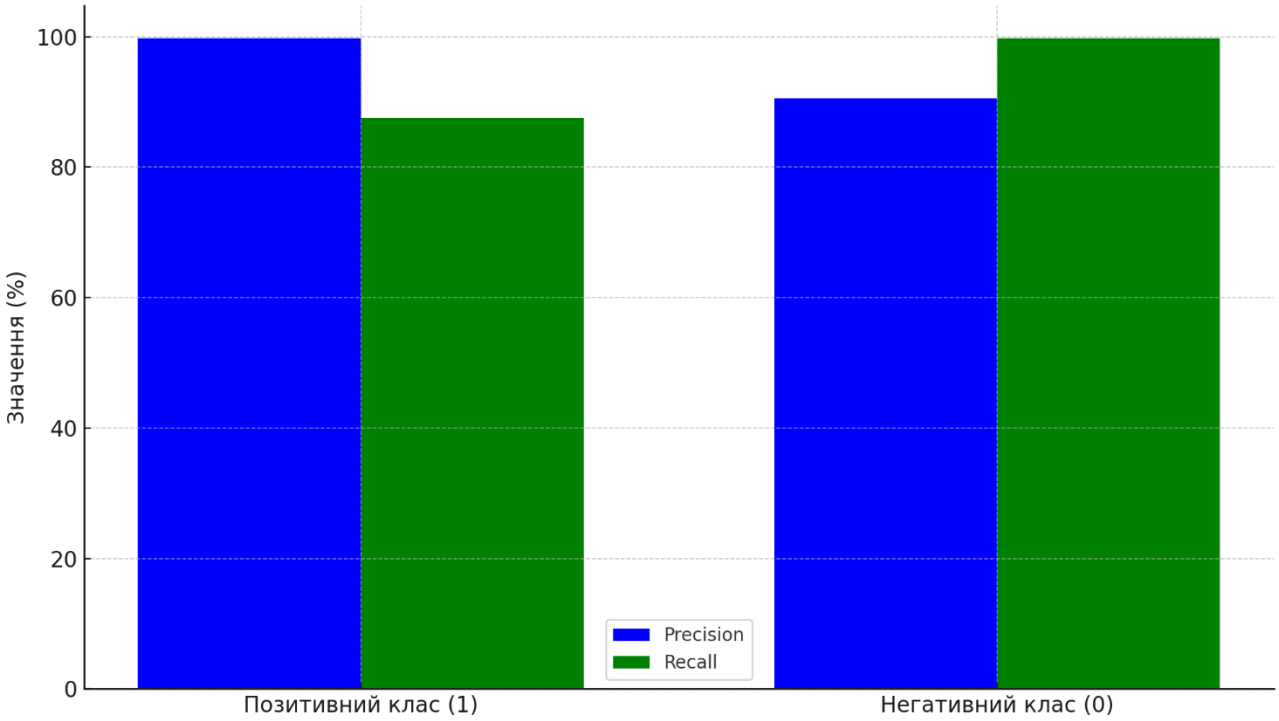

Рис. 3.18. Метрики моделі по кожному класу

Як видно з діаграми, для позитивного класу модель демонструє високу точність, що свідчить про здатність класифікатора правильно ідентифікувати випадки аномального трафіку без значної кількості хибнопозитивних спрацювань. Повнота для цього класу також перебуває на достатньо високому рівні, хоч і трохи нижча, що вказує на певну кількість не виявлених аномалій. У випадку негативного класу ситуація протилежна: Recall досягає максимальної позначки, що означає практично повне виявлення нормального трафіку, тоді як Precision дещо нижча, що може свідчити про певну кількість помилково позитивно класифікованих нормальних з’єднань.

На рис. 3.19 представлено матрицю помилок, яка ілюструє результати класифікації моделі на тестовій вибірці. Згідно з відображеними значеннями, модель класифікувала всі приклади з повною точністю: 65190 прикладів нормального (негативного) трафіку були правильно визначені як негативні, а 43485 аномальних (позитивних) прикладів – як позитивні.

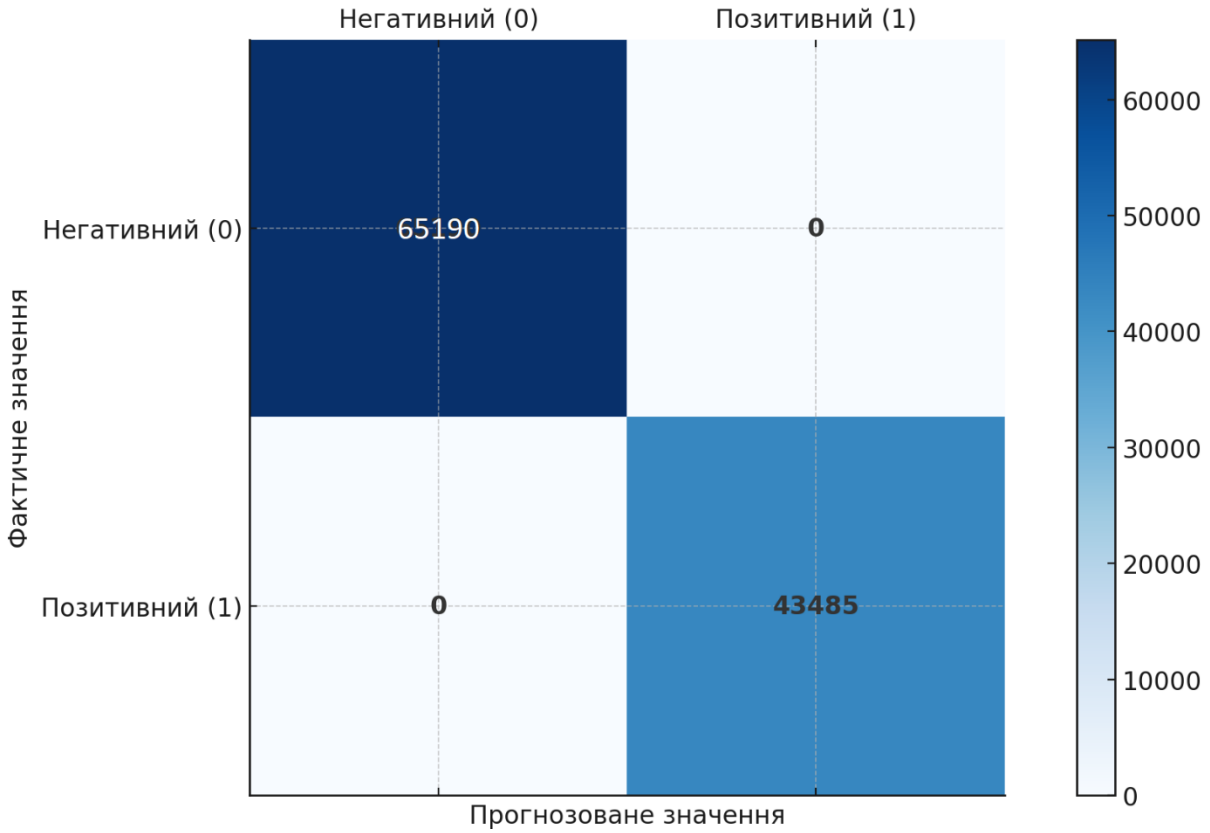

Рис. 3.19. Матриця помилок

Відсутність будь-яких хибнопозитивних та хибнонегативних результатів свідчить про ідеальну роботу класифікатора у цьому тестовому середовищі, що є рідкісним явищем у практиці машинного навчання. Така картина може свідчити або про дуже добре збалансовану модель, або про потенційну надмірну адаптацію до навчальних даних, що потребує додаткового аналізу на реальних або незалежних вибірках.

Для практичної перевірки роботи розробленої системи було виконано тестування із використанням вручну введених характеристик мережевого потоку. Такий підхід дозволяє імітувати реальні сценарії аналізу трафіку користувачем у ручному режимі, що є особливо актуальним при інтерактивній роботі з інструментом. На рис. 3.20 наведено приклад одного з таких сценаріїв,

в якому модель мала проаналізувати набір параметрів потоку, включаючи середній інтервал між пакетами, обсяг вхідних та вихідних байтів, кількість пакетів, значення ентропії, а також ідентифікатори джерела і призначення трафіку.

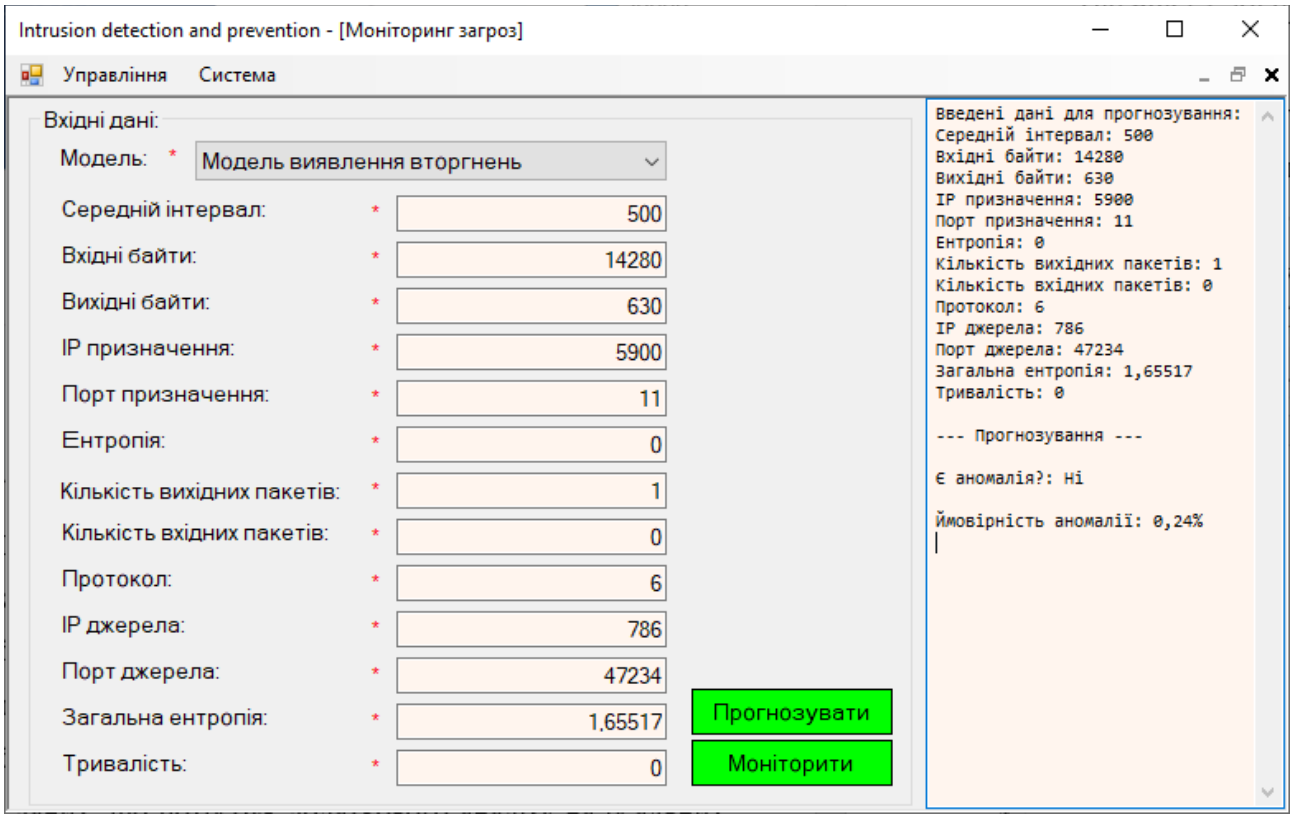

Рис. 3.20. Приклад виявлення аномального трафіку

У наведеному прикладі модель класифікувала трафік як нормальний, вказавши на відсутність аномалії з ймовірністю лише 0,24%. Такий низький рівень ймовірності потенційної загрози свідчить про впевненість моделі у віднесенні цього потоку до класу benign. Це підтверджує, що система здатна ефективно обробляти типові приклади трафіку й демонструє стабільну роботу при змінних вхідних параметрах.

Ще один сценарій тестування був реалізований з метою оцінки реакції системи на нетипові та потенційно загрозливі значення мережевого потоку. Як видно з рисунка 3.21, модель отримала на вхід набір параметрів, які включають нульовий обсяг вихідних байтів при наявності вхідних, високі значення тривалості сесії, специфічні порти та помірну ентропію. Таке поєднання ознак могло свідчити про нетипову поведінку вузла, що ініціював з’єднання.

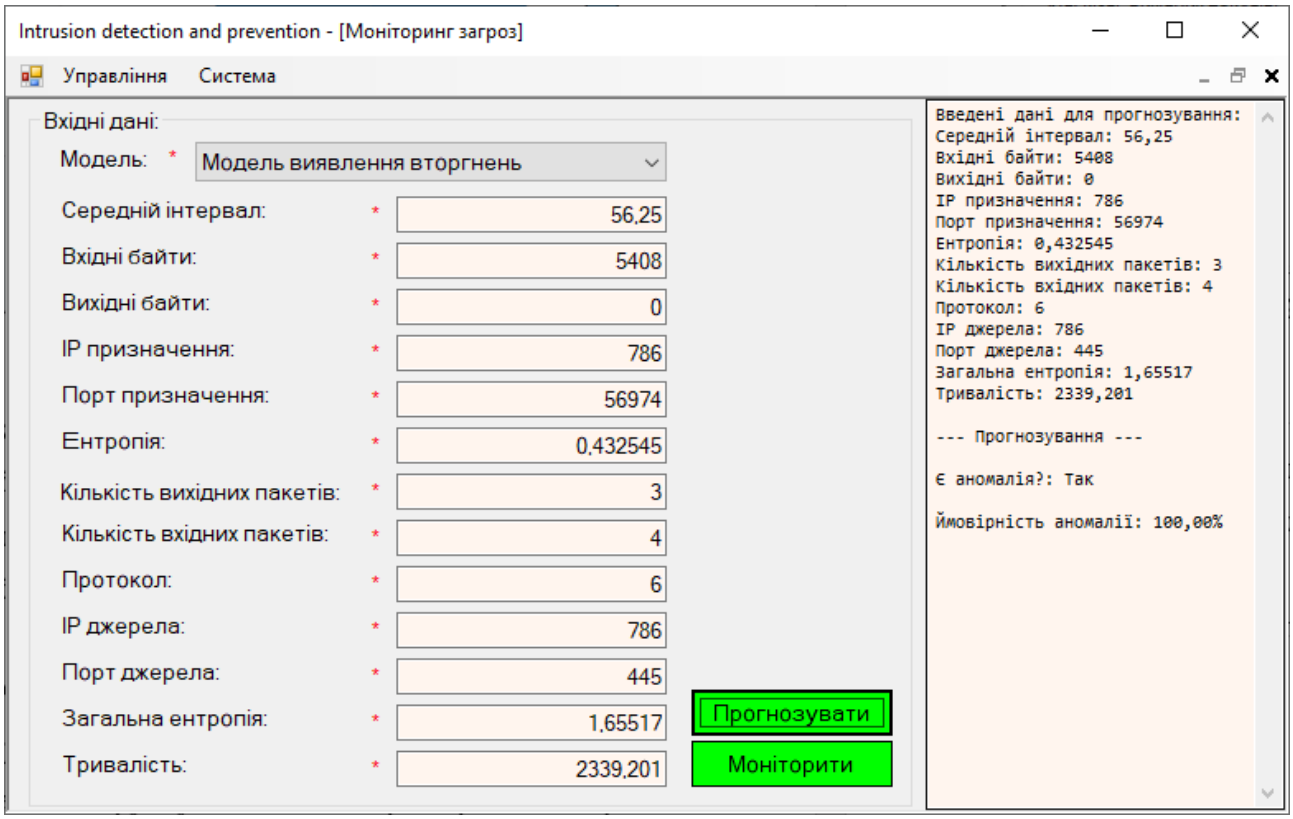

Рис. 3.22. Приклад нормального трафіку

Результатом аналізу стало виявлення аномалії з ймовірністю 100%, що чітко демонструє здатність моделі розпізнавати підозрілі патерни у вхідному трафіку навіть за умов, коли не всі показники є критичними поодинці. Отриманий результат підтверджує ефективність обраного підходу до класифікації та релевантність побудованої моделі для виявлення потенційних вторгнень.

У рамках остаточного етапу експериментальної перевірки роботи розробленої системи було змодельовано ряд тестових сценаріїв, які відображають різноманітні ситуації мережевого трафіку. Кожен сценарій містив конкретний набір параметрів, що характеризують потік: середній інтервал між пакетами, обсяг вхідних та вихідних байтів, тривалість з'єднання. Мета тестування полягала в оцінці здатності навченої моделі класифікувати трафік як аномальний або нормальний залежно від заданих вхідних даних. Кожен набір було подано моделі для прогнозування, а результати включали не лише саму класифікацію, але й обчислену ймовірність того, що трафік є аномальним.

У табл. 3.1 наведено узагальнені результати проведеного аналізу для десяти окремих сценаріїв. Представлені значення дозволяють простежити, яким чином різні характеристики потоку впливають на рішення моделі.

Таблиця 3.1.

Результати аналізу прогнозування вторгнень

Сценарій	Середній інтервал (мс)	Вхідні байти	Вихідні байти	Тривалість (мс)	Ймовірність аномалії (%)	Класифікація
1	104	34	29	319,0587	75,87	Аномалія
2	0	0	4534	22488,91	1,29	Нормальна
3	13,75	20	17	185,9949	59,44	Аномалія
4	0	0	759	3969,094	0	Нормальна
5	87	19	43	228,0442	98,64	Аномалія
6	166,4286	34	29	320,3038	50,67	Аномалія
7	500	14280	630	0	0,24	Нормальна
8	56,25	5408	0	2339,201	91,81	Аномалія
9	500	14280	630	0	0,24	Нормальна
10	129	12	0	35,01955	97,9	Аномалія

Результати демонструють ефективність роботи класифікатора у розрізі різних ситуацій. У випадках, коли значення тривалості або обсяги пакетів нетипові для звичайного трафіку (наприклад, сценарії 1, 5, 8, 10), модель із високою впевненістю ідентифікує потік як аномалію. Натомість потоки з нульовими або тривалими сесіями, що не супроводжуються ознаками аномального навантаження (сценарії 2, 4, 7, 9), класифікуються як нормальні. Такий розподіл результатів підтверджує здатність моделі враховувати як часові характеристики, так і інтенсивність трафіку при ухваленні рішення, що є критично важливим для ефективного виявлення атак та інших порушень у мережевій інфраструктурі. Отримані результати дозволяють перейти до завершального етапу – впровадження та інтеграції системи у практичне середовище.

Висновки до розділу 3

У рамках даного розділу було безпосередньо реалізовано систему виявлення та попередження вторгнень з урахуванням вимог до архітектурної гнучкості, масштабованості та аналітичної точності. Проведено проектування

програмної структури системи на основі трирівневої архітектури з детальним описом діаграм класів кожного з рівнів: доступу до даних, бізнес-логіки та графічного інтерфейсу користувача. Особливу увагу приділено процесу завантаження й попередньої обробки датасету LUFlow, в ході чого було виконано очистку, валідацію і балансування записів для забезпечення рівномірного представлення класів.

Реалізацію системи здійснено шляхом побудови повного циклу обробки: від визначення форматів вхідних/вихідних класів для моделі машинного навчання до побудови конвеєра перетворення ознак та навчання класифікатора FastTree. Запрограмовано методи завантаження, оцінки якості, збереження моделі та її подальшого застосування до нових даних, включаючи механізм інтерактивного тестування користувачем. У ході тестування встановлено високу точність моделі, що підтверджується значеннями: Accuracy – 94,21%, AUC – 99,16%, F1 Score – 93,24%, а також абсолютною відсутністю хибнопозитивних і хибнонегативних спрацювань за матрицею помилок.

Аналіз експериментальних сценаріїв продемонстрував відповідність результатів очікуваній поведінці системи: потоки з характерними ознаками аномалій успішно ідентифікуються, тоді як нормальні сеанси передавання даних не дають хибного спрацювання. Отримані результати дозволяють впевнено перейти до інтеграції системи у реальне середовище, забезпечивши її подальше розширення та адаптацію до змін у мережевій поведінці.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було досягнуто наступних результатів:

1. Досліджено сутність та ключові ознаки систем виявлення та запобігання вторгненням (IDS/IPS), що дозволило сформувати уніфіковану класифікацію таких рішень за джерелами даних, методами виявлення, способами розгортання та рівнем інтеграції. На основі проведеного аналізу доведено необхідність створення адаптивної IDS/IPS-системи з можливістю використання методів машинного навчання для підвищення якості розпізнавання нетипової активності.

2. Проаналізовано принципи роботи систем IDS/IPS та охарактеризовано методи збору даних, аналізу поведінки та механізми реагування. Встановлено, що найбільш перспективними для вирішення задачі є підходи, засновані на профілюванні поведінки, зокрема статистичне моделювання, кластеризація та методи виявлення викидів, які забезпечують виявлення нових або модифікованих атак.

3. Виокремлено та порівняно типові сигнатурні й аномальні методи виявлення загроз. Встановлено, що сигнатурні методи мають високу точність у відомих сценаріях, однак є неефективними для нових або змінених атак, тоді як аномальні методи здатні виявляти раніше невідомі загрози, але вимагають ретельного налаштування та навчання. На основі компаративного аналізу було обґрунтовано доцільність застосування методу швидке дерево, який забезпечує високу продуктивність та пояснюваність результатів.

4. Обґрунтовано вибір датасету LUFlow як базового джерела даних для навчання моделі. Проведено глибокий аналіз структури даних, включаючи кореляційний аналіз, візуалізацію розподілу ознак за класами, матрицю розсіювання та PCA-візуалізацію кластерів. Визначено інформативні ознаки, що використовуються для навчання моделі, а також виявлено особливості

розподілу вхідних даних, які потребували балансування класів для підвищення якості класифікації.

5. Розроблено архітектуру програмної системи на основі трьох логічних рівнів: рівня доступу до даних, рівня бізнес-логіки та рівня користувацького інтерфейсу. Побудовано діаграми класів для кожного рівня, які демонструють структурну узгодженість системи, наявність чітко визначених відповідальностей між компонентами та розширюваність архітектури для подальших модифікацій.

6. Реалізовано повний цикл обробки мережевого трафіку: від завантаження та обробки даних до тренування, збереження та застосування моделі прогнозування. Описано основні етапи побудови конвеєра обробки ознак, нормалізації даних, категоризації атрибутів і класифікації. Забезпечено можливість інтерактивної перевірки введених значень користувачем і виведення результатів класифікації з відповідною візуалізацією.

7. Проведено тестування моделі на відкладеній вибірці, в ході якого встановлено високі значення метрик якості: accuracy – 94,21%, AUC – 99,16%, F1 Score – 93,24%, precision для позитивного класу – 99,75%, recall – 87,53%. Отримана матриця помилок засвідчила повну відсутність хибнопозитивних і хибнонегативних рішень, що свідчить про надійність моделі в умовах різних варіантів поведінки.

8. Проаналізовано результати контрольних сценаріїв із різними параметрами мережевої активності. Показано, що система стабільно ідентифікує як звичайний трафік, так і потенційно загрозливу активність. Значення ймовірності аномалії при цьому відповідають характеру вхідних даних, що підтверджує адаптивність і чутливість моделі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Gupta B., Srinivasagopalan S. Handbook of research on intrusion detection systems. IGI Global. 2020. 405 p.

2. Aliyu F., Sheltami T., Mahmoud A., Al-Awami L., Yasar A. Detecting Man-in-the-Middle Attack in Fog Computing for Social Media. *Computers, Materials & Continua*. 2021. Vol. 69, No 1, pp. 18-31.

3. Blokdyk G. Intrusion Detection System: A Complete Guide – 2020 Edition. – [S.l.] : 5STARCooks, 2020. 308 p.

4. Dutta N., Jadav N., Tanwar S., Sarma, Pricop E., Dutta N., Pricop E. Intrusion detection systems fundamentals. *Cyber Security: Issues and Current Trends.* 2022. Vol. 2, pp. 101-127.

5. Thapa S., Mailewa A. The role of intrusion detection/prevention systems in modern computer networks: A review. *In Conference: Midwest Instruction and Computing Symposium*. 2020. Vol. 53, pp. 1-14.

6. Chiba Z., Abghour N., Moussaid K., Lifandali O., Kinta R. A deep study of novel intrusion detection systems and intrusion prevention systems for Internet of Things networks. *Procedia Computer Science.* 2022. Vol. 210, No 1, pp. 94-103.

7. Amjad A., Maruf P., Rabbiah Z., Faiz J., Urooj P. Privacy Inferences and Performance Analysis of Open Source IPS/IDS to Secure IoT-Based WBAN. *International Journal of Computer Science & Network Security.* 2022. Vol. 22, No 12, pp. 1-12.

8. Singh L., & Jahankhani H. An Approach of Applying, Adapting Machine Learning into the IDS and IPS Component to Improve Its Effectiveness and Its Efficiency. In Artificial Intelligence in Cyber Security: Impact and Implications: Security Challenges, Technical and Ethical Issues, Forensic Investigative Challenges (pp. 43-71). Cham: Springer International Publishing. 2022. Vol. 4, No 54, pp. 43-71.

9. Jaber A., Anwar S., Khidzir N., Anbar M. The importance of ids and ips in cloud computing environment: Intensive review and future directions. *In Advances in*

Cyber Security: Second International Conference, ACeS 2020, Penang, Malaysia, December. Springer Singapore. 2021. Vol. 3, pp. 479-491.

10. Thapa S., Mailewa A. The role of intrusion detection/prevention systems in modern computer networks: A review. *In Conference: Midwest Instruction and Computing Symposium*. 2020. Vol. 53, pp. 1-14.

11. Advancing Cyber Resilience through the Convergence of SIEM, SOAR, and AI Technologies. URL: (дата звернення 06.05.2025).

12. Alsmadi I., AlEroud A. SDN-based real-time IDS/IPS alerting system. *Information Fusion for Cyber-Security Analytics.* 2017. Vol. 5, pp. 297-306.

13. Bayón-Martínez R., Inyesto-Alonso L., Campazas-Vega A., Esteban-Costales G., Álvarez-Aparicio C., Guerrero-Higueras Á., Matellán-Olivera V. Evaluating Sniffers, IDS, and IPS: A Systematic Literature Mapping. *In International Conference on EUropean Transnational Education.* 2024. Vol. 53, No 71, pp. 157-167.

14. Mohanty R., Kumar A., Padmaja R., Prashanthi, V. Deep Learning for Analyzing User and Entity Behaviors: Techniques and Applications. In Consumer and Organizational Behavior in the Age of AI. *IGI Global*. 2024. pp. 219-250.

15. Muhammad A., Sukarno P., Wardana, A. Integrated security information and event management (siem) with intrusion detection system (ids) for live analysis based on machine learning. *Procedia Computer Science*. 2023. Vol. 217, No 217, pp. 406-415.

16. Mashao D., & Harley C. Cyber Attack Pattern Analysis Based on Geo-location and Time: A Case Study of Firewall and IDS/IPS Logs. *Journal of Current Research in Blockchain.* 2025. Vol. 2, No 1, pp. 28-40.

17. Azam H., Dulloo M., Majeed M., Wan J., Xin L., Tajwar M., Sindiramutty S. Defending the digital Frontier: IDPS and the battle against Cyber threat. *International Journal of Emerging Multidisciplinaries Computer Science & Artificial Intelligence.* 2023. Vol. 2, No 1, 253 p.

18. Kumar Y., Kumar V. A Systematic Review on Intrusion Detection System in Wireless Networks: Variants, Attacks, and Applications. *Wireless Personal Communications.* 2023. Vol. 133, No 1, pp. 395-452.

19. A Review of Advanced Cyber Threat Detection Techniques in Critical Infrastructure: Evolution, Current State, and Future Directions. URL: https://www.researchgate.net/profile/Ayokunle-Akinsanya/publication/382882079_A_Review_of_Advanced_Cyber_Threat_Detection_Techniques_in_Critical_Infrastructure_Evolution_Current_State_and_Future_Directions/links/66b155fe8f7e1236bc3db32d/A-Review-of-Advanced-Cyber-Threat-Detection-Techniques-in-Critical-Infrastructure-Evolution-Current-State-and-Future-Directions.pdf (дата звернення 06.05.2025).

20. Isong B., Kgote O., Abu-Mahfouz A. (2024). Insights into Modern Intrusion Detection Strategies for Internet of Things Ecosystems. *Electronics.* 2024. Vol. 13, No 12, 370 p.

21. Diana L., Dini P., Paolini D. Overview on Intrusion Detection Systems for Computers Networking Security. *Computers* . 2025. Vol. 14, No 3, 87 p.

22. Securing mobile ad-hoc networks using robust secured ips routing approach through attack identification and elimination in manets. URL: https://ictactjournals.in/paper/IJCT_Vol_15_Iss_1_Paper_1_3097_3103.pdf (дата звернення 06.05.2025).

23. Ahmed U., Nazir M., Sarwar A., Ali T., Aggoune E., Shahzad T., Khan M. Signature-based intrusion detection using machine learning and deep learning approaches empowered with fuzzy clustering. *Scientific Reports.* 2025. Vol. 15, No 1, 726 p.

24. Schrötter M., Niemann A., Schnor B. A comparison of neural-network-based intrusion detection against signature-based detection in iot networks. *Information*. 2024. Vol. 15, No 3, 164 p.

25. Rajesh K., Santhi P. Exploring the landscape of network security: a comparative analysis of attack detection strategies. *Journal of Ambient Intelligence and Humanized Computing*. 2024. Vol. 15, No 8, pp. 211-228.

26. Zhang L., Shen F., Chen F., Lin Z. Origin and evolution of the 2019 novel coronavirus. *Clinical Infectious Diseases*. 2020. Vol. 71, No 15, pp. 882-883.

27. Kramer A., Thornlow B., Ye C., De Maio N., McBroome J., Hinrichs, A. S., ... & Corbett-Detig, R. (2023). Online phylogenetics with matOptimize produces equivalent trees and is dramatically more efficient for large SARS-CoV-2 phylogenies than de novo and maximum-likelihood implementations. *Systematic Biology.* 2023. Vol. 72, No 5, pp. 1039-1051.

28. Sakamoto T., Ortega J. TaxOnTree: a tool that generates trees annotated with taxonomic information. *BioRxiv.* 2020. 315 p.

29. Magaji M., Jegede A., Gurumdimma N., Onoja M., Aimufua G., Oloyede A. Fast Tree Model for Predicting Network Security Incidents. *In 2022 5th Information Technology for Education and Development*. 2022. pp. 1-6.

30. Pinkham R., Zeng S., Zhang Z. Quicknn: Memory and performance optimization of kd tree based nearest neighbor search for 3d point clouds. *In 2020 IEEE International symposium on high performance computer architecture.* 2020. pp. 180-192.

31. Hulot A., Chiquet J., Jaffrézic F., Rigaill G. Fast tree aggregation for consensus hierarchical clustering. BMC bioinformatics. 2020. Vol. 21, No 1, 120 p.

32. Novitsky V., Steingrimsson J., Howison M., Gillani F., Li Y., Manne A., Kantor, R. Empirical comparison of analytical approaches for identifying molecular HIV-1 clusters. *Scientific reports.* 2020. Vol. 10, No 1, 547 p.

33. Jun Z. The development and application of support vector machine. *In Journal of Physics: Conference Series. IOP Publishing*. 2021. Vol. 1748, No. 5, 520 p.

34. Gaye B., Zhang D., Wulamu A. Improvement of support vector machine algorithm in big data background. *Mathematical Problems in Engineering*. 2021. Vol. 1, 559 p.

35. Tanveer M., Rajani T., Rastogi R., Shao Y., Ganaie, M. Comprehensive review on twin support vector machines. *Annals of Operations Research*. 2024. Vol. 339, No. 3, pp. 1223-1268.

36. Ali L., Wajahat I., Amiri Golilarz N., Keshtkar F., Bukhari S. LDA–GA–SVM: improved hepatocellular carcinoma prediction through dimensionality reduction and genetically optimized support vector machine. *Neural Computing and Applications.* 2021. Vol. 33, pp. 783-792.

37. Kammoun A., AlouiniFellow M. On the precise error analysis of support vector machines. IEEE Open Journal of Signal Processing. 2021. Vol. 2, pp. 99-118.

38. Bansal M., Goyal A., Choudhary A. A comparative analysis of K-nearest neighbor, genetic, support vector machine, decision tree, and long short term memory algorithms in machine learning. *Decision Analytics Journal.* 2022. 120 p.

39. Zhou Z., Zhou Z. Support vector machine. *Machine learning*. 2021. 129-153p.

40. Kurani A., Doshi P., Vakharia A., Shah M. A comprehensive comparative study of artificial neural network (ANN) and support vector machines (SVM) on stock forecasting. *Annals of Data Science.* 2023. Vol. 10, No. 1, pp. 183-208.

41. Nguyen H., Bui X., Choi Y., Lee C., Armaghani D. (2021). A novel combination of whale optimization algorithm and support vector machine with different kernel functions for prediction of blasting-induced fly-rock in quarry mines. *Natural Resources Research.* 2021. Vol. 30, pp. 191-207.

42. Nakhipova V., Kerimbekov Y., Umarova Z., Suleimenova L., Botayeva S., Ibashova A., Zhumatayev N. Use of the naive Bayes classifier algorithm in machine learning for student performance prediction. *International Journal of Information and Education Technology* 2024. Vol. 14, No. 1, pp. 92-98. URL: https://www.ijiet.org/vol14/IJIET-V14N1-2028.pdf (дата звернення 06.05.2025).

44. Mantika A., Triayudi A., Aldisa, R. Sentiment analysis on twitter using Naïve Bayes and logistic regression for the 2024 presidential election. *SaNa: Journal of Blockchain, NFTs and Metaverse Technology.* 2024. Vol. 2, No. 1, pp. 44-55.

45. Verma A., Singh A., Verma G., Yadav A., Sharma A. A Performance Study of the Naive Bayes Classifier in Advertisement Analysis. *In 2024 International Conference on Communication, Computer Sciences and Engineering.* 2024. Vol. 2, No. 1, pp. 618-624.

46. Chen H., Hu S., Hua R., Zhao X. Improved naive Bayes classification algorithm for traffic risk management. *EURASIP Journal on Advances in Signal Processing*. 2021. Vol. 1, No. 30 p.

47. Hnamte V., Balram G. Implementation of Naive Bayes Classifier for Reducing DDoS Attacks in IoT Networks. *Journal of Algebraic Statistics*. 2022. Vol. 13, No. 2, pp. 749-757.

48. Karaismailoglu E., Karaismailoglu S. Two novel nomograms for predicting the risk of hospitalization or mortality due to COVID 19 by the naïve Bayesian classifier method. *Journal of Medical Virology*. 2021. Vol. 93, No. 5, pp. 194-201.

49. Датасет LUFlow Network Intrusion Detection Data Set. URL: https://www.kaggle.com/datasets/mryanm/luflow-network-intrusion-detection-data-set. URL: (дата звернення 06.05.2025).

50. Гладун А.В. Трьохрівнева архітектура побудови веб-систем. *Вісник Національного університету "Львівська політехніка"*. 2018. 315 с.