Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Кафедра кібербезпеки
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр
(освітній рівень)

на тему: "Порівняльний аналіз підходів підвищення ефективності систем виявлення вторгнень"

Виконав: студент VI курсу, групи СБм-61

Спеціальності:
125 «Кібербезпека та захист інформації»
(шифр і назва напряму підготовки, спеціальності)
Гнатківський Любомир Васильович
підпис (прізвище та ініціали)

Керівник Загородна Н.В.
підпис (прізвище та ініціали)

Нормоконтроль Стадник М. А.
підпис (прізвище та ініціали)

Завідувач кафедри Загородна Н.В.
підпис (прізвище та ініціали)

Рецензент
підпис (прізвище та ініціали)

м. Тернопіль – 2025

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра кібербезпеки
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Загородна Н.В.
(підпис) (прізвище та ініціали)
«__» _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Магістр
(назва освітнього ступеня)

за спеціальністю 125 Кібербезпека та захист інформації
(шифр і назва спеціальності)

Студенту Гнатківському Любомиру Васильовичу
(прізвище, ім'я, по батькові)

1. Тема роботи "Порівняльний аналіз підходів підвищення ефективності систем виявлення вторгнень"

Керівник роботи Загородна Наталія Володимирівна, к.т.н., доц. завідувач кафедри КБ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 24 » 11 2025 року № 4/7-1024

2. Термін подання студентом завершеної роботи 16.12.2025

3. Вихідні дані до роботи датасет NSL-KDD та літературні джерела за темою дослідження .

4. Зміст роботи (перелік питань, які потрібно розробити)
Системи виявлення вторгнень як частина мережевої безпеки
Підходи до підвищення ефективності систем виявлення вторгнень
Експериментальні дослідження

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
Тема, мета, завдання
Засоби мережевого захисту від загроз
NIDS vs HIDS, Класифікація систем виявлення вторгнень (IDS) за способом виявлення атак
Показники ефективності виявлення атак в IDS
Основні обмеження та проблеми систем виявлення вторгнень
Датасет NSL-KDD. Методологія дослідження
Порівняльний аналіз отриманих результатів за метрикою accuracy
Порівняльний аналіз отриманих результатів за метрикою f1
Напрямки подальших досліджень. Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Осухівська Г.М., к.т.н., доцент		
Безпека в надзвичайних ситуаціях	Теслюк В.М., проректор з адміністративно-господарської роботи та будівництва		

7. Дата видачі завдання 19.09.2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	20.09 – 23.09	*Виконано*
2.	Огляд технологій захисту мережевих ресурсів	24.09-07.10	*Виконано*
3.	Аналіз систем виявлення вторгнень	08.10-25.10	*Виконано*
4.	Дослідження підходів до підвищення ефективності систем виявлення вторгнень	26.10-10.11	*Виконано*
5.	Проведення експериментальних досліджень	11.11-20.11	*Виконано*
6.	Оформлення розділу 1 «Системи виявлення вторгнень як частина мережевої безпеки»	21.11-25.11	*Виконано*
7.	Оформлення розділу 2 «Підходи до підвищення ефективності систем виявлення вторгнень»	26.11-30.11	*Виконано*
8.	Оформлення розділу 3 «Експериментальні дослідження»	01.12-05.12	*Виконано*
9.	Виконання завдання до підрозділу «Охорона праці»	04.12-08.12	*Виконано*
10.	Виконання завдання до підрозділу «Безпека в надзвичай них ситуаціях»	04.12-08.12	*Виконано*
11.	Оформлення кваліфікаційної роботи	01.12-08.12	*Виконано*
12.	Нормоконтроль	09.12 – 11.12	*Виконано*
13.	Перевірка на плагіат	12.12 – 15.12	*Виконано*
14.	Попередній захист кваліфікаційної роботи	16.12 – 20.12	*Виконано*
15.	Захист кваліфікаційної роботи	24.12.2025	

Студент _____ Гнатківський Л.В.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Загородна Н.В.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Порівняльний аналіз підходів підвищення ефективності систем виявлення вторгнень // ОР «Магістр» // Гнатківський Любомир Васильович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБм-61 // Тернопіль, 2025 // С. 94, рис. – 15, табл. – 6 , кресл. – _-_, додат. – 3.

Ключові слова: СИСТЕМИ ВИЯВЛЕННЯ ВТОРГНЕНЬ, МАШИННЕ НАВЧАННЯ, НЕЧІТКА ЛОГІКА, XGBOOST, RANDOM FOREST, NSL-KDD.

Кваліфікаційна робота присвячена дослідженню та порівняльному аналізу сучасних підходів до підвищення ефективності систем виявлення вторгнень (IDS) в комп'ютерних мережах. У роботі проведено комплексний огляд технологій захисту мережевих ресурсів, детально проаналізовано класифікацію IDS за способом розміщення та методами виявлення загроз, а також визначено основні показники ефективності систем на основі метрик загальної точності, влучності та повноти.

Основну увагу приділено дослідженню двох ключових підходів: методів машинного навчання (Random Forest, XGBoost, SVM) та експертних систем на основі нечіткої логіки. Розроблено гібридну модель, що поєднує традиційні алгоритми класифікації з нечіткими правилами для агрегації слабких індикаторів атак. Експериментальні дослідження проведено на датасеті NSL-KDD, що містить 125 973 записи мережевих з'єднань з п'ятьма категоріями трафіку.

Розроблені рекомендації щодо практичного застосування гібридних систем виявлення вторгнень у корпоративних мережах та напрямки подальших досліджень у галузі інтеграції штучного інтелекту в системи кібербезпеки.

ABSTRACT

Comparative analysis of approaches to improving Intrusion Detection System efficiency// Thesis of educational level "Master"// Liubomyr Hnatkivsyi // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Cybersecurity, group СБм-61 // Ternopil, 2025 // p. 94, figs. 15, tbls. 6 , drws. ____, apps. 3.

Keywords: INTRUSION DETECTION SYSTEMS, MACHINE LEARNING, FUZZY LOGIC, XGBOOST, RANDOM FOREST, NSL-KDD.

This thesis is dedicated to the research and comparative analysis of modern approaches to improving the efficiency of Intrusion Detection Systems (IDS) in computer networks. The work provides a comprehensive overview of network resource protection technologies, detailed analysis of IDS classification by deployment method and intrusion detection techniques, and defines key performance indicators based on accuracy, precision, and recall metrics.

The main focus is on investigating two key approaches: machine learning methods (Random Forest, XGBoost, SVM) and expert systems based on fuzzy logic. A hybrid model was developed that combines traditional classification algorithms with fuzzy rules for aggregating weak attack indicators. Experimental research was conducted on the NSL-KDD dataset containing 125,973 network connection records across five traffic categories.

Recommendations for practical application of hybrid intrusion detection systems in corporate networks and future research directions in the field of artificial intelligence integration into cybersecurity systems were developed.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

CNN	–	Convolutional Neural Network
DoS	–	Denial of Service
GDPR	–	General Data Protection Regulation
GRU	–	Gated Recurrent Unit
HIDS	–	Host-based Intrusion Detection System
HIPAA	–	Health Insurance Portability and Accountability Act
IDS	–	Intrusion Detection System,
IPS	–	Intrusion Prevention System,
LSTM	–	Long Short-Term Memory
ML	–	Machine Learning
NGFW	–	Next-Generation Firewalls
NIDS	–	Network-based Intrusion Detection System
SIEM	–	Security Information and Event Management
SVM	–	Support Vector Machine
RNN	–	Recurrent Neural Network
TLS	–	Transport Layer Security
VPN	–	Virtual Private Network
UEBA	–	User and Entity Behavior Analytics

ВСТУП

Актуальність теми.

У сучасному цифровому суспільстві кіберзагрози еволюціонують з неймовірною швидкістю, створюючи критичні виклики для інформаційної безпеки організацій. Системи виявлення вторгнень є фундаментальним компонентом багаторівневої архітектури захисту, проте традиційні підходи на основі сигнатур виявляються неефективними проти нових, раніше невідомих атак (zero-day) та складних багатоетапних вторгнень. За даними досліджень, у 2024 році середній час виявлення атаки становив 181 днів, що дозволяє зловмисникам завдати значної шкоди до виявлення компрометації [1].

Методи машинного навчання демонструють високу точність виявлення відомих атак, однак страждають від проблем дисбалансу класів, високого рівня хибних спрацювань та складності інтерпретації рішень. Водночас експертні системи на основі нечіткої логіки забезпечують прозорість та можливість формалізації експертних знань, але обмежені у здатності адаптуватися до нових загроз. Актуальність дослідження полягає у необхідності розробки гібридних підходів, що поєднують переваги обох методів для підвищення ефективності виявлення широкого спектру кіберзагроз, особливо рідкісних та складних класів атак.

Мета і задачі дослідження.

Метою роботи є підвищення ефективності систем виявлення вторгнень шляхом розробки та дослідження гібридного підходу, що поєднує методи машинного навчання з експертними системами на основі нечіткої логіки.

Завданнями дослідження є:

1. Провести комплексний аналіз сучасних технологій захисту мережевих ресурсів та класифікувати системи виявлення вторгнень за ключовими ознаками.
2. Дослідити основні показники ефективності IDS та визначити метрики для об'єктивного оцінювання якості виявлення атак.
3. Проаналізувати проблеми та обмеження сучасних систем виявлення вторгнень, зокрема дисбаланс класів та високий рівень хибних спрацювань.

4. Дослідити підходи на основі методів машинного навчання (Random Forest, XGBoost, SVM) для виявлення мережевих атак.

5. Вивчити архітектуру експертних систем та застосування нечіткої логіки для моделювання невизначеності в системах безпеки.

6. Розробити гібридну модель, що інтегрує нечіткі ознаки в традиційні алгоритми машинного навчання для покращення виявлення рідкісних класів атак.

7. Провести експериментальні дослідження на датасеті NSL-KDD та оцінити ефективність розробленого підходу.

8. Виконати порівняльний аналіз базових та покращених моделей за метриками точності, влучності, повноти та часу виконання.

Об’єкт дослідження.

Процеси виявлення та класифікації мережевих вторгнень у системах інформаційної безпеки на основі аналізу характеристик мережевого трафіку.

Предмет дослідження.

Методи та засоби підвищення ефективності систем виявлення вторгнень, в тому числі шляхом інтеграції алгоритмів машинного навчання з експертними системами на основі нечіткої логіки.

Наукова новизна одержаних результатів кваліфікаційної роботи.

Розроблено гібридний підхід до виявлення мережевих вторгнень, що поєднує традиційні методи машинного навчання з елементами нечіткої логіки для агрегації слабких індикаторів атак, що дозволило підвищити ефективність виявлення рідкісних класів загроз.

Практичне значення одержаних результатів.

Результати дослідження мають безпосереднє практичне застосування для побудови ефективних систем виявлення вторгнень у корпоративних мережах. Програмна реалізація розробленого підходу на мові Python з використанням бібліотек scikit-learn, XGBoost та scikit-fuzzy може бути інтегрована в існуючі IDS та SIEM-платформи.

Апробація результатів магістерської роботи. Основні результати дослідження були апробовані на XIII науково-технічній конференції «Інформаційні моделі, системи та технології» (ТНТУ, м.Тернопіль, Україна).

Публікації. Основні результати кваліфікаційної роботи опубліковано у працях конференції (див. Додаток А).

РОЗДІЛ 1 СИСТЕМИ ВИЯВЛЕННЯ ВТОРГНЕНЬ ЯК ЧАСТИНА МЕРЕЖЕВОЇ БЕЗПЕКИ

1.1 Огляд технологій захисту мережевих ресурсів

Мережева безпека є комплексом заходів, політик, практик та технологій, спрямованих на захист цілісності, конфіденційності та доступності комп'ютерних мереж та ресурсів від несанкціонованого доступу, зловживання, модифікації або відмови у обслуговуванні. У сучасному цифровому середовищі жоден окремий інструмент не може забезпечити повний захист; необхідне застосування багаторівневої архітектури, яка охоплює превентивні, детекторні та реактивні засоби. В [1] наведено основні технології захисту мереж

1.1.1 Міжмережеві екрани (Firewalls)

Фаєрвол належить до превентивних засобів спрямованих на те, щоб запобігти проникненню загрози в мережу. Превентивні засоби створюють бар'єр між внутрішньою захищеною мережею та зовнішнім неконтрольованим середовищем (наприклад, Інтернетом). На рисунку 1.1 наведено схему, яка демонструє, що фаєрвол є бар’єром, що відділяє надійну внутрішню мережу від ненадійної зовнішньої.

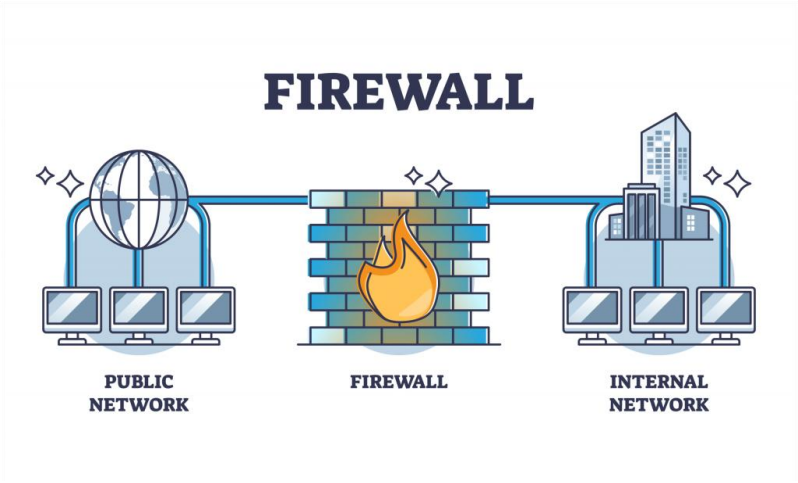

Рисунок 1.1 – Ілюстрація схеми розміщення фаєрволу

Міжмережевий екран, або фаєрвол, є найфундаментальнішим і найстарішим компонентом мережевої безпеки. Його основна функція – фільтрація мережевого трафіку на основі заздалегідь визначених правил безпеки. Фаєрвол перевіряє заголовки мережевих пакетів (IP-адреси, порти, протоколи) і вирішує, дозволити чи заборонити проходження трафіку. Основне завдання фаєрвола – блокувати несанкціонований доступ до комп'ютера чи мережі, водночас дозволяючи легітимним комунікаціям проходити без перешкод. Також він відстежує стан активних мережевих з’єднань, дозволяючи лише той трафік, який є частиною вже встановленої сесії. Це перша лінія оборони проти кіберзагроз, яка захищає від зловмисного програмного забезпечення, хакерських атак та несанкціонованого доступу до конфіденційних даних.

Існує кілька типів фаєрволів, кожен з яких працює на різних рівнях мережевої моделі. Пакетні фільтри перевіряють заголовки пакетів даних і приймають рішення на основі IP-адрес та портів. Stateful-фаєрволи відстежують стан активних з'єднань і приймають рішення на основі контексту трафіку. Проксі-фаєрволи діють як посередники між користувачами та Інтернетом, перевіряючи весь трафік на рівні додатків. Сучасні фаєрволи нового покоління (NGFW) поєднують традиційні функції з додатковими можливостями, такими як виявлення вторгнень, антивірусний захист та фільтрація контенту.

Правильне налаштування фаєрвола критично важливе для ефективного захисту. Адміністратори повинні регулярно оновлювати правила фільтрації, враховуючи нові загрози та зміни в інфраструктурі мережі. Важливо знаходити баланс між безпекою та зручністю – занадто суворі правила можуть заблокувати легітимний трафік, а занадто м'які залишать систему вразливою. Фаєрволи також потребують регулярного моніторингу журналів подій, щоб виявляти підозрілу активність та своєчасно реагувати на потенційні інциденти безпеки. У сучасному цифровому світі фаєрвол є невід'ємним компонентом комплексної стратегії кібербезпеки для організацій будь-якого розміру.

1.1.2 Віртуальні приватні мережі (VPN)

Віртуальна приватна мережа (VPN) – це технологія, яка створює безпечне та зашифроване з'єднання через менш захищену мережу, зазвичай Інтернет. VPN дозволяє користувачам безпечно передавати дані так, ніби їхні пристрої безпосередньо підключені до приватної мережі, навіть якщо вони фізично знаходяться в іншому місці. Ця технологія використовує тунелювання – процес, при якому дані шифруються на одному кінці з'єднання та розшифровуються на іншому. VPN приховує реальну IP-адресу користувача, замінюючи її адресою VPN-сервера, що забезпечує анонімність та конфіденційність онлайн-активності.

Існує декілька типів VPN, кожен з яких призначений для різних цілей. Remote Access VPN дозволяє окремим користувачам підключатися до приватної мережі з віддалених локацій, що особливо корисно для віддалених працівників. Site-to-Site VPN з'єднує цілі мережі між собою, наприклад, офіси компанії в різних містах. Клієнтські VPN – найпоширеніший тип для звичайних користувачів, який встановлюється як додаток на комп'ютері чи смартфоні. Протоколи VPN, такі як OpenVPN, WireGuard, IKEv2 та L2TP/IPsec, визначають способи шифрування та передачі даних, кожен з власними перевагами щодо швидкості та безпеки. На рисунку 1.2 наведено поєднання двох типів.

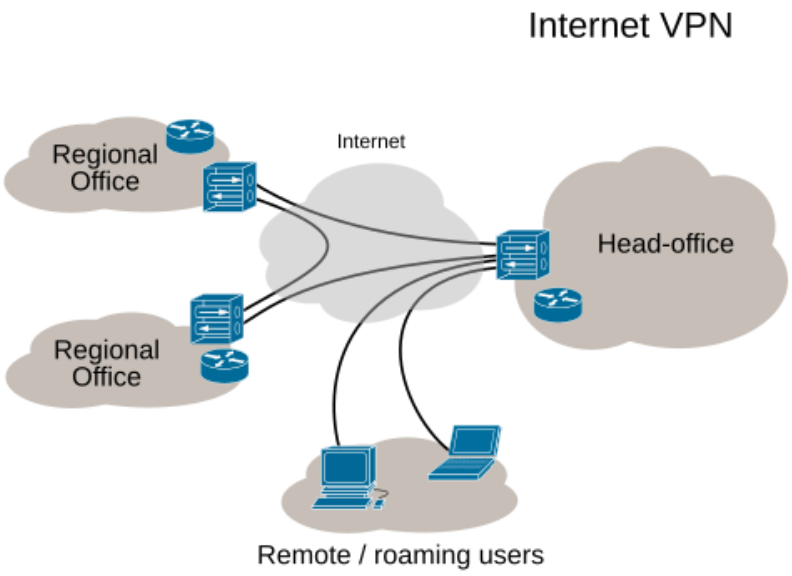

Рисунок 1.2 – Демонстація VPN-підключення, що показує інтранет-конфігурації Site-to-Site та Remote Access VPN, які використовуються разом

VPN використовується в різних сценаріях для забезпечення безпеки та приватності. Компанії застосовують VPN для захисту корпоративних даних, дозволяючи співробітникам безпечно працювати з дому або під час поїздок. Звичайні користувачі використовують VPN для захисту особистої інформації при підключенні до публічних Wi-Fi мереж, які часто є вразливими до кібератак. VPN також допомагає обходити географічні обмеження контенту, дозволяючи отримувати доступ до сайтів та сервісів, заблокованих у певних регіонах. У країнах з цензурою Інтернету VPN стає важливим інструментом для доступу до вільної інформації та збереження свободи слова.

При виборі VPN-сервісу важливо враховувати кілька факторів. Надійне шифрування та політика нульових логів (no-logs policy) забезпечують максимальну конфіденційність користувача. Швидкість з'єднання та кількість серверів у різних країнах впливають на якість роботи сервісу. Варто звертати увагу на репутацію провайдера, юрисдикцію, під якою він працює, та наявність додаткових функцій безпеки, таких як kill switch (автоматичне відключення Інтернету при розриві VPN-з'єднання). Хоча VPN значно підвищує рівень безпеки та приватності, він не є абсолютною панацеєю – користувачі повинні розуміти обмеження технології та поєднувати її використання з іншими практиками кібербезпеки для комплексного захисту. В [3] детально розглянути типи та особливості застосування VPN

1.1.3 Системи виявлення та запобігання вторгненням

Навіть найсучасніші превентивні засоби не можуть гарантувати 100% захисту, оскільки зловмисники постійно розробляють нові методи обходу. Тому критично важливу роль відіграють системи, які виявляють загрози, що вже проникли. Системи виявлення вторгнень (IDS) та системи запобігання вторгненням (IPS) є критично важливими компонентами сучасної мережевої безпеки, але виконують вони принципово різні функції у захисті інформаційних ресурсів. IDS є, по суті, системою моніторингу та оповіщення: вона пасивно аналізує мережевий трафік або системні події, шукаючи ознаки відомих атак,

використовуючи сигнатури, або нетипові шаблони поведінки, що можуть вказувати на вторгнення. Якщо IDS ідентифікує підозрілу активність, вона генерує сповіщення (алерти), які передаються адміністраторам безпеки для подальшого ручного аналізу та реагування. Таким чином, IDS працює як "око" безпеки, що допомагає виявити як успішні, так і невдалі спроби атак, не втручаючись безпосередньо у потік даних.

На відміну від пасивної природи IDS, системи запобігання вторгненням (IPS) мають активну функцію, оскільки вони розміщуються безпосередньо в тракті мережевого трафіку ("в розриві"). Це дозволяє їм не лише виявляти загрози, але й негайно вживати автоматичних заходів реагування, щоб запобігти успішному завершенню атаки. Як тільки IPS ідентифікує шкідливий пакет або послідовність дій, вона може динамічно блокувати це з'єднання, скинути шкідливі пакети або навіть змінити конфігурацію інших пристроїв безпеки, таких як міжмережеві екрани. Ця здатність до негайної нейтралізації загроз робить IPS ключовим елементом превентивної оборони, що захищає мережу в реальному часі. На рисунку 1.3 наведено різницю у використанні IDS та IPS у корпоративних мережах.

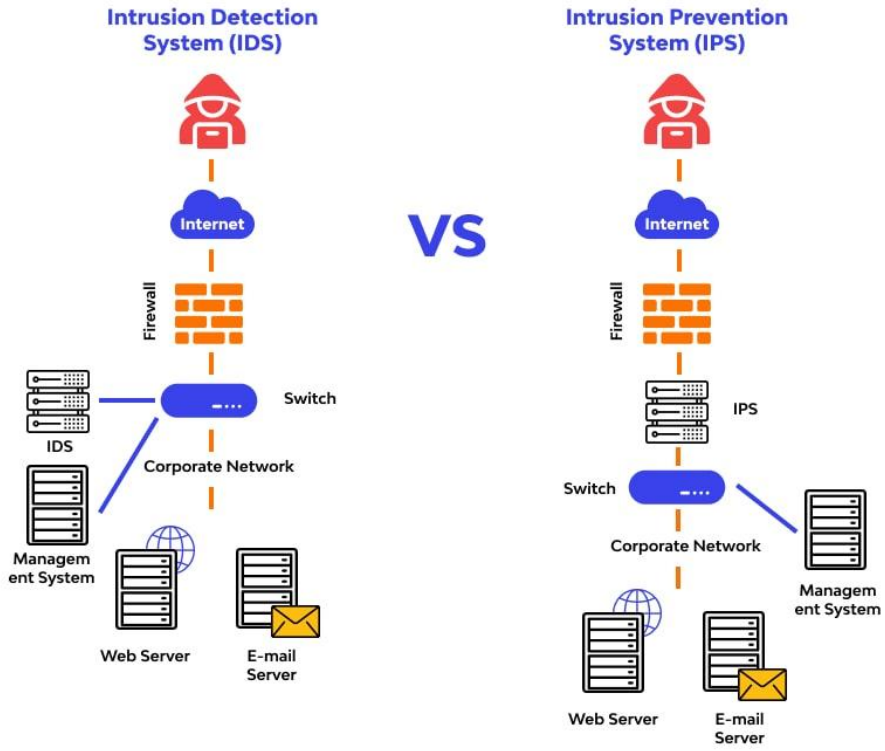

Рисунок 1.3 – Використання IDS та IPS у корпоративних мережах.

Хоча функціональність IPS є значно потужнішою, її розміщення "в розриві" мережі також несе певні ризики та виклики. Будь-яка затримка або помилка в роботі IPS може призвести до уповільнення або повного переривання нормального мережевого трафіку, що критично важливо для бізнес-операцій. Крім того, хибні спрацьовування (False Positives) в IPS мають набагато серйозніші наслідки, ніж в IDS, оскільки вони можуть призвести до блокування легітимних користувачів або важливих комунікацій. Тому налаштування та калібрування IPS вимагає значно більшої уваги та точності, щоб мінімізувати небажані побічні ефекти.

У сучасних архітектурах безпеки часто застосовується комбінований підхід, де системи IDS та IPS доповнюють одна одну, забезпечуючи концепцію "глибокої оборони". IPS використовується для блокування очевидних і високоризикованих загроз, діючи як фільтр. У той же час, IDS використовується для моніторингу трафіку, який пройшов через IPS, виявляючи більш тонкі, невідомі або складні атаки, які потребують детальнішого аналізу та ручного втручання. Таке поєднання інструментів забезпечує баланс між превентивним захистом та всебічним моніторингом, підвищуючи загальну стійкість мережі до кіберзагроз.

1.1.4 SIEM та UEBA системи

Системи SIEM є централізованою платформою для збору, агрегації, кореляції та аналізу логів та подій безпеки з різних джерел: фаєрволів, IDS, IPS, серверів, додатків та кінцевих точок. Системи управління інформацією та подіями безпеки (SIEM) виконують функцію централізованого інтелектуального хабу в інфраструктурі безпеки організації, об'єднуючи два ключові напрямки: управління інформацією безпеки (SIM) та управління подіями безпеки (SEM). Основна функція SIEM полягає у консолідації та нормалізації величезних обсягів логів і даних про події, що надходять з усіх компонентів мережі – фаєрволів, систем виявлення вторгнень, серверів, додатків та кінцевих точок. Після збору, ці дані, які можуть мати різні формати, проходять процес нормалізації та

збагачення для забезпечення єдиної структури. Ключовим етапом є кореляція подій: SIEM використовує складні алгоритми та правила для автоматичного виявлення взаємозв'язків між окремими, на перший погляд, непов'язаними інцидентами, що відбуваються у різних частинах мережі. Це дозволяє ідентифікувати складні, багатоетапні атаки, які розгортаються повільно і могли б бути пропущені окремими засобами захисту, а також надає єдину картину загроз для аналітиків безпеки.

Друга критична функція SIEM – це забезпечення моніторингу в реальному часі та підтримка відповідності нормативним вимогам (Compliance). Платформа SIEM постійно аналізує корельовані дані для негайного виявлення підозрілих патернів, що сигналізують про активне вторгнення, та генерує сповіщення з високим рівнем пріоритету, що значно скорочує час реагування на інцидент. Крім того, завдяки здатності збирати та зберігати великі обсяги даних про безпеку протягом тривалого часу, SIEM забезпечує необхідну звітність та аудит для дотримання галузевих та державних стандартів, таких як GDPR, PCI DSS чи HIPAA. Це не тільки допомагає довести належну обачність під час розслідування, але й надає історичні дані для глибокого аналізу загроз, виявлення вразливостей та подальшого зміцнення загальної позиції безпеки організації.

UEBA (User and Entity Behavior Analytics) – це більш просунутий інструмент, який використовує машинне навчання та статистичні моделі для встановлення базового профілю "нормальної" поведінки кожного користувача, системи чи пристрою. Він виявляє відхилення від цього профілю (наприклад, користувач, який зазвичай працює з 9:00 до 17:00, раптом заходить о 3:00 ночі та завантажує велику кількість конфіденційних даних). UEBA значно підвищує точність виявлення внутрішніх загроз та скомпрометованих облікових записів, зменшуючи при цьому кількість хибних спрацьовувань.

1.2 Класифікація систем виявлення вторгнень

Системи виявлення вторгнень є важливою складовою сучасних засобів забезпечення інформаційної безпеки комп'ютерних мереж і інформаційних систем. З огляду на різноманіття архітектур, принципів функціонування та методів аналізу, IDS класифікують за кількома ключовими ознаками. Найбільш поширеними та практично значущими є класифікації за способом розміщення системи та за методом виявлення вторгнень. Такий поділ дозволяє глибше зрозуміти можливості різних підходів, а також оцінити їхні переваги, недоліки та доцільність застосування в конкретних умовах.

1.2.1 Класифікація IDS за способом розміщення

Залежно від того, де саме здійснюється збір та аналіз подій безпеки, системи виявлення вторгнень поділяються на HIDS, NIDS та Hybrid IDS.

HIDS (Host-based Intrusion Detection System) – це системи, які встановлюються безпосередньо на окремі хости або сервери та здійснюють моніторинг подій на рівні операційної системи. Такі системи аналізують журнали подій, системні виклики, зміни у файловій системі, поведінку процесів та дії користувачів. Основною перевагою HIDS є можливість глибокого аналізу внутрішнього стану хоста, що дозволяє виявляти атаки, які не завжди проявляються у мережевому трафіку. Зокрема, HIDS ефективні для виявлення атак типу privilege escalation, несанкціонованих змін системних файлів або шкідливої активності з боку легітимних користувачів. Водночас недоліком таких систем є обмежена масштабованість та залежність від ресурсів конкретного хоста, а також складність централізованого управління у великих інфраструктурах.

NIDS (Network-based Intrusion Detection System) функціонують на рівні мережі та аналізують мережевий трафік, що передається між вузлами. Такі системи зазвичай розміщуються у стратегічних точках мережі, наприклад на шлюзах або комутаторах, і працюють у режимі пасивного прослуховування

трафіку. Основним завданням NIDS є виявлення підозрілих шаблонів передачі даних, характерних для відомих атак, сканування портів, атак типу DoS або спроб несанкціонованого доступу. Перевагою NIDS є здатність одночасно контролювати велику кількість хостів без необхідності встановлення агентів на кожен з них. Однак такі системи можуть мати обмеження у випадках використання зашифрованого трафіку, а також не завжди здатні точно визначити наслідки атаки на конкретному хості.

В [4] наведено переваги і недоліки кожно типу, які узагальнені в фіці 1.1.

Таблиця 1.1 - Порівняння HIDS та NIDS

Критерій оцінювання	HIDS	NIDS
Виявлення порушників	Ідеально підходить для виявлення внутрішніх порушників	Ідеально підходить для виявлення зовнішніх порушників
Час реагування	Низька швидкість реагування в реальному часі; покращена продуктивність для довготривалих атак	Висока швидкість реагування на зовнішні атаки
Оцінка завданої шкоди	Відмінна оцінка завданої шкоди	Слабка оцінка завданої шкоди
Захищеність від зловмисників	Висока ефективність проти зовнішніх зловмисників	Висока ефективність проти зовнішніх зловмисників
Визначення очікуваних загроз	Відмінно визначає підозрілі шаблони поведінки	Відмінно визначає підозрілі шаблони поведінки

На рисунку 1.4 наведено типове розташування HIDS і NIDS [5].

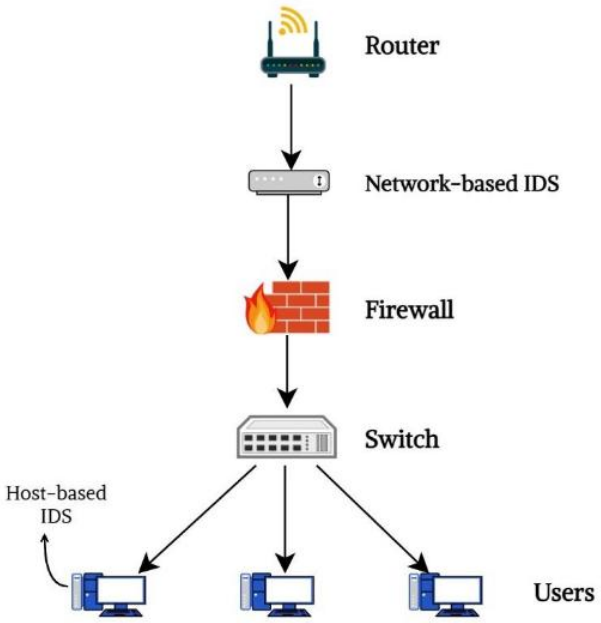

Рисунок 1.4 – Типове розташування HIDS і NIDS

Hybrid IDS поєднують у собі характеристики як HIDS, так і NIDS, забезпечуючи комплексний підхід до виявлення вторгнень. У таких системах мережевий аналіз доповнюється інформацією з рівня хостів, що дозволяє підвищити точність виявлення атак та зменшити кількість хибних спрацювань. Гібридний підхід є особливо актуальним для великих корпоративних мереж, де необхідно одночасно контролювати мережеві потоки та внутрішні події на критично важливих серверах. Водночас впровадження Hybrid IDS може вимагати значних обчислювальних ресурсів і складнішої архітектури управління.

1.2.2 Класифікація IDS за методом виявлення вторгнень

Іншим важливим критерієм класифікації IDS є метод, який використовується для ідентифікації потенційно шкідливої активності. За цим критерієм системи виявлення вторгнень поділяються на signature-based, anomaly-based.

Signature-based IDS ґрунтуються на порівнянні поточної активності з наперед визначеною базою сигнатур відомих атак. Сигнатура зазвичай являє

собою формалізований опис характерних ознак конкретної атаки, таких як послідовність байтів у пакеті або специфічний шаблон мережевого запиту. Основною перевагою цього підходу є висока точність виявлення відомих загроз і низький рівень хибних спрацювань. Водночас signature-based системи практично не здатні виявляти нові або модифіковані атаки, сигнатури яких відсутні у базі даних. Крім того, ефективність таких систем безпосередньо залежить від регулярного оновлення сигнатур.

Anomaly-based IDS орієнтовані на виявлення відхилень від нормальної поведінки системи або мережі. У процесі роботи такі системи формують модель «нормального» стану на основі статистичного аналізу або методів машинного навчання, після чого фіксують будь-які суттєві відхилення від цієї моделі як потенційні атаки. Перевагою anomaly-based підходу є здатність виявляти раніше невідомі загрози та атаки нульового дня. Однак побудова точної моделі нормальної поведінки є складним завданням, а некоректне навчання може призводити до високого рівня хибних спрацювань. Це особливо актуально для динамічних середовищ, де поведінка користувачів і систем може часто змінюватися. На рисунку 1.5 наведено принципи роботи виявлення вторгнень за

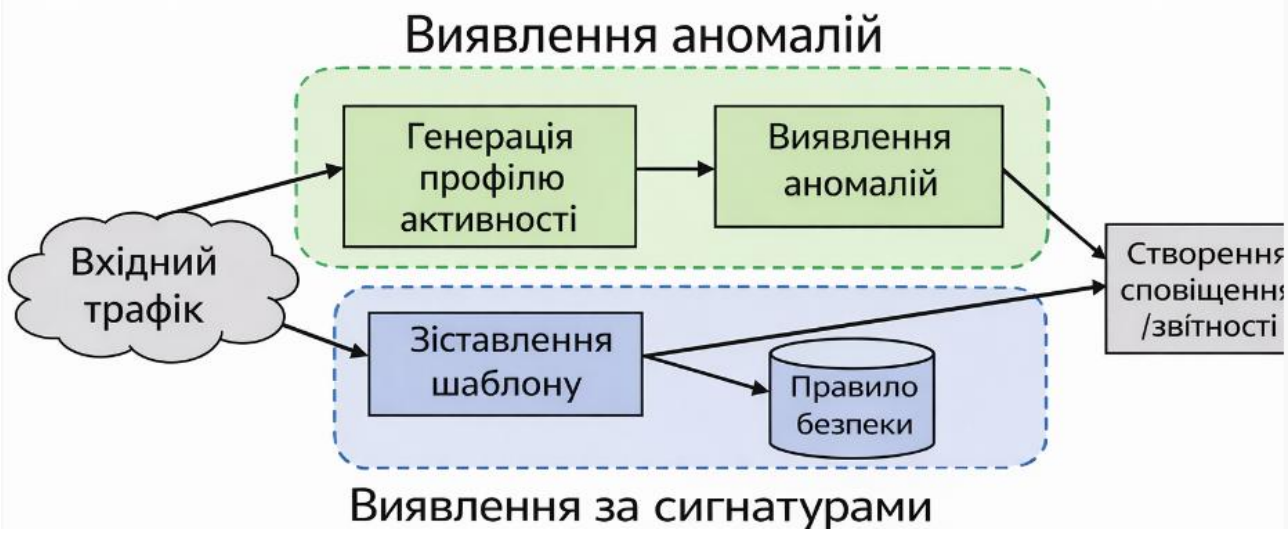

Рисунок 1.5 – Принципи роботи IDS

Порівняльний аналіз систем виявлення вторгнень, що працюють на основі сигнатур та аномалій проведено в [6]

Гібридні системи (Hybrid Detection) поєднують переваги обох попередніх методів, використовуючи як сигнатурне виявлення, так і аналіз аномалій. Такий підхід дозволяє системі ефективно виявляти як відомі загрози з високою точністю, так і нові атаки через моніторинг незвичайної активності. Сучасні IDS часто включають додаткові методи, такі як виявлення на основі поведінки (Behavior-based Detection), яке аналізує послідовності дій, та виявлення на основі протоколів (Protocol-based Detection), що перевіряє відповідність трафіку встановленим стандартам протоколів. Гібридні рішення також можуть використовувати евристичний аналіз, який застосовує правила та алгоритми для виявлення підозрілої активності на основі характеристик, типових для атак.

Вибір методу виявлення вторгнень залежить від специфічних потреб організації, типу захищуваних ресурсів та доступних ресурсів для управління системою. Великі підприємства часто віддають перевагу гібридним рішенням, які забезпечують найширший захист, хоча вони вимагають більше обчислювальних потужностей та експертизи для налаштування. Малі організації можуть почати з систем на основі сигнатур через їхню простоту та надійність. Незалежно від обраного методу, ефективна IDS повинна регулярно оновлюватися, правильно налаштовуватися та інтегруватися з іншими компонентами системи безпеки для забезпечення комплексного захисту від сучасних кіберзагроз. Таким чином, класифікація IDS за способом розміщення та методом виявлення дозволяє систематизувати існуючі рішення у сфері виявлення вторгнень та створює основу для подальшого порівняльного аналізу підходів підвищення їх ефективності. Розуміння особливостей кожного класу IDS є необхідним для обґрунтованого вибору або проєктування систем захисту інформації в умовах сучасних кіберзагроз.

1.3 Основні показники ефективності IDS

Оцінка ефективності систем виявлення вторгнень (IDS) є критично важливою для забезпечення надійного захисту мережевої інфраструктури. Правильне розуміння та моніторинг ключових показників дозволяє визначити, наскільки ефективно система виконує свої функції, та виявити потенційні слабкі місця в конфігурації. Основні метрики ефективності IDS включають показники точності виявлення загроз, швидкості реагування, продуктивності системи та впливу на мережеву інфраструктуру. Комплексний аналіз цих показників допомагає організаціям оптимізувати роботу систем безпеки та приймати обґрунтовані рішення щодо вдосконалення захисту.

1.3.1 Критерії точності виявлення загроз

Основні показниками точності виявлення загроз лежать в основі матриці, наведено на рисунку 1.6

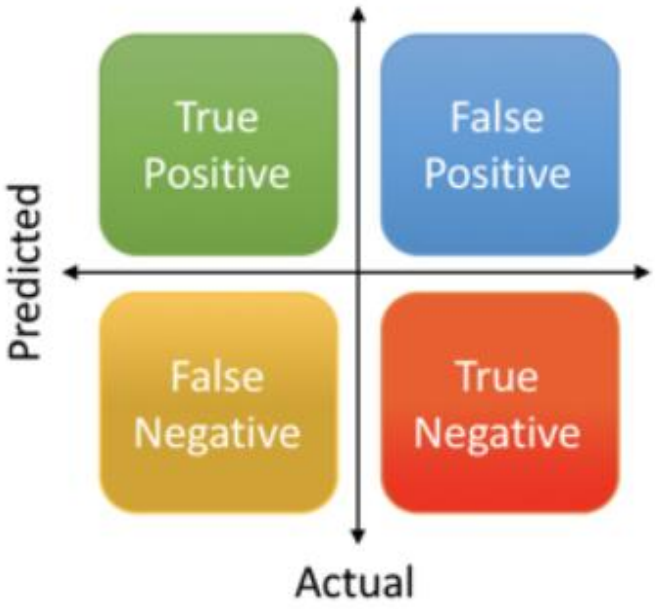

Рисунок 1.6 – Матриця похибок

True Positive (TP) - Істинно позитивні спрацювання представляють ситуації, коли IDS правильно ідентифікує реальну атаку або вторгнення. Це найбажаніший результат роботи системи, який вказує на те, що загроза була успішно виявлена та зафіксована. Високий рівень істинно позитивних спрацювань свідчить про ефективність налаштування правил виявлення та актуальність бази сигнатур. Організації прагнуть максимізувати цей показник, оскільки він безпосередньо відображає здатність системи захищати від реальних загроз.

False Positive (FP) - Хибно позитивні спрацювання виникають, коли система помилково класифікує нормальну активність як загрозу. Це один з найпроблематичніших аспектів роботи IDS, оскільки велика кількість хибних тривог призводить до втоми операторів безпеки та зниження уваги до реальних загроз. Високий рівень FP може спричинити ситуацію, коли адміністратори почнуть ігнорувати попередження системи, що робить її фактично неефективною. Зниження кількості хибних спрацювань досягається через точне налаштування правил, використання контекстної інформації та регулярне оновлення профілів нормальної активності мережі.

False Negative (FN) - Хибно негативні результати представляють найнебезпечнішу ситуацію, коли реальна атака залишається непоміченою системою. Такі пропуски можуть мати катастрофічні наслідки, оскільки зловмисники отримують можливість здійснювати свої дії без протидії з боку системи безпеки. Причинами хибно негативних результатів можуть бути застарілі сигнатури, неправильно налаштовані правила виявлення, використання нових методів атак або навмисне обхід системи зловмисниками через техніки уникнення виявлення.

True Negative (TN) - Істинно негативні результати фіксуються, коли система правильно ідентифікує нормальну активність як безпечну та не генерує попередження. Хоча цей показник рідко обговорюється, він важливий для розуміння загальної картини роботи IDS, оскільки велика частина мережевого трафіку є легітимною.

Похідними метриками ефективності є:

- Точність (Accuracy).
- Чутливість або повнота (Recall/Sensitivity).
- Точність виявлення (Precision).
- F-міра (F-Score).

Точність (Accuracy) визначається як відсоток правильно класифікованих подій від загальної кількості подій і розраховується за формулою:

$$\text{Accuracy} = \frac{TN + TP}{TN + TP + FN + FP}$$

Цей показник дає загальне уявлення про ефективність системи, але може бути оманливим у випадках дисбалансу класів, коли нормальна активність значно переважає кількість атак.

Чутливість або повнота (Recall/Sensitivity) вимірює здатність системи виявляти всі реальні атаки та розраховується як:

$$\text{Recall} = \frac{TP}{TP + FN}$$

Високий показник чутливості означає, що система рідко пропускає реальні загрози. Для критичних систем, де пропуск атаки може мати серйозні наслідки, прагнуть максимізувати саме цей показник, навіть якщо це призводить до збільшення хибних спрацювань.

Влунічність (Precision) показує, яка частка згенерованих попереджень є дійсно загрозами, та обчислюється як:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Висока влучність означає, що більшість тривог системи є обґрунтованими, що зменшує навантаження на команду безпеки та підвищує довіру до системи.

Баланс між чутливістю та влучністю є ключовим завданням при налаштуванні IDS.

F-міра (F-Score) є гармонійним середнім між влучністю та чутливістю і розраховується за формулою:

$$\mathrm{F1} = \frac{2 * \mathrm{Precision} * \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}$$

Цей показник особливо корисний, коли потрібно оцінити загальну ефективність системи з урахуванням обох аспектів – здатності виявляти загрози та мінімізації хибних спрацювань.

Більше показників точності наведено в [7]

1.3.2 Інші показники ефективності систем виявлення вторгнень

Оцінювати IDS можна також за показниками продуктивності та швидкості, до яких належить час виявлення, час реагування, пропускна здатність та інші

Час виявлення (Detection Time) вимірює проміжок часу між початком атаки та її виявленням системою. Чим коротший цей час, тим швидше можна розпочати відповідні заходи протидії. Для систем реального часу цей показник критично важливий, оскільки затримка у виявленні може дозволити зловмисникам завдати значної шкоди або скомпрометувати важливі дані.

Час реагування (Response Time) показує, скільки часу потрібно системі для генерування попередження після виявлення підозрілої активності. У високонавантажених мережах цей показник може значно впливати на загальну ефективність безпеки, оскільки затримки у сповіщеннях можуть призвести до пропуску критичних моментів для втручання.

Пропускна здатність (Throughput) визначає кількість мережевого трафіку, який система може аналізувати за одиницю часу без втрати пакетів або зниження якості виявлення. Цей показник особливо важливий для високошвидкісних мереж, де IDS повинна обробляти великі обсяги даних в режимі реального часу.

Використання ресурсів включає показники завантаження процесора, споживання пам'яті та використання мережевої смуги пропускання системою IDS. Ефективна система повинна забезпечувати високий рівень безпеки з мінімальним впливом на продуктивність мережевої інфраструктури.

До операційних показників належать:

- Коефіцієнт покриття (Coverage Rate), що визначає відсоток захищеної мережевої інфраструктури та типів атак, які система здатна виявляти. Комплексна IDS повинна охоплювати широкий спектр можливих векторів атак та моніторити всі критичні сегменти мережі.
- Адаптивність (Adaptability), що оцінює здатність системи пристосовуватися до змін у мережевому середовищі, нових типів загроз та еволюції тактик зловмисників. Системи з високою адаптивністю можуть самостійно оновлювати правила виявлення та профілі нормальної поведінки.
- Масштабованість (Scalability), що показує, наскільки ефективно система може розширюватися разом зі зростанням мережевої інфраструктури без суттєвого погіршення продуктивності або точності виявлення.

Ефективна оцінка IDS вимагає комплексного підходу до аналізу всіх показників одночасно. Не існує єдиної метрики, яка б повністю характеризувала якість роботи системи – необхідно враховувати баланс між точністю виявлення, кількістю хибних спрацювань, продуктивністю та вартістю обслуговування. Організації повинні визначити пріоритетні показники відповідно до своїх специфічних потреб, рівня критичності інфраструктури та доступних ресурсів. Регулярний моніторинг цих метрик, аналіз тенденцій та коригування налаштувань системи забезпечують підтримку оптимального рівня захисту в умовах постійно змінюваного ландшафту кіберзагроз.

1.4 Проблеми та обмеження сучасних систем виявлення вторгнень

Одним з найсерйозніших обмежень сучасних IDS є технічні обмеження та виклики продуктивності, зокрема складність обробки великих обсягів мережевого трафіку в режимі реального часу. З розвитком хмарних технологій,

Інтернету речей та переходом на високошвидкісні мережі 10 Гбіт/с та вище, системи виявлення вторгнень стикаються з проблемою перевантаження. Глибока інспекція пакетів та аналіз шифрованого трафіку вимагають значних обчислювальних ресурсів, що може призводити до втрати пакетів, затримок у виявленні загроз або необхідності обмежувати функціональність системи. У високонавантажених середовищах IDS може пропускати атаки просто через неможливість своєчасно проаналізувати весь трафік, створюючи так звані "сліпі зони" в системі безпеки.

Шифрування трафіку, яке стало стандартом для забезпечення конфіденційності в Інтернеті, парадоксально створює серйозні проблеми для систем виявлення вторгнень. Протоколи HTTPS, TLS та VPN унеможливлюють традиційний аналіз вмісту пакетів без розшифрування, що вимагає впровадження SSL/TLS-інспекції. Однак це породжує нові виклики: необхідність управління сертифікатами, додаткове навантаження на систему, потенційні вразливості та питання приватності користувачів. Зловмисники активно використовують шифрування для приховування своєї діяльності, створюючи зашифровані канали передачі шкідливих даних, які важко виявити без порушення цілісності захищених з'єднань.

Надмірна кількість хибних позитивних спрацювань залишається однією з найбільших проблем IDS, яка значно знижує ефективність системи безпеки. В середньому, організації можуть отримувати тисячі попереджень щодня, з яких велика частина виявляється хибними тривогами. Це створює ефект "втоми від попереджень", коли адміністратори безпеки починають ігнорувати сповіщення або поверхнево їх аналізувати, що може призвести до пропуску реальних загроз. Налаштування системи для зменшення хибних спрацювань часто означає підвищення порогів чутливості, що автоматично збільшує ризик пропуску справжніх атак. Цей компроміс між чутливістю та специфічністю є фундаментальною проблемою, яка вимагає постійного балансування та експертного налаштування.

Складність сучасних мережевих середовищ з їхніми мікросервісами, контейнерами, динамічними IP-адресами та гібридною інфраструктурою робить

створення точних профілів нормальної поведінки надзвичайно складним завданням. Системи виявлення аномалій потребують тривалого періоду навчання та постійного коригування базових профілів, щоб адаптуватися до легітимних змін у мережі. Будь-яке оновлення інфраструктури, впровадження нових додатків або зміни в бізнес-процесах можуть викликати лавину хибних попереджень до перенавчання системи.

Сучасні кіберзагрози еволюціонують з неймовірною швидкістю, а зловмисники постійно розробляють нові методи атак та техніки обходу систем виявлення. Системи на основі сигнатур принципово не здатні виявляти zero-day експлойти та модифіковані варіанти відомих атак. Час між виявленням нової загрози та створенням відповідної сигнатури створює вікно вразливості, яким активно користуються зловмисники. Поліморфні та метаморфні зловмисні програми здатні змінювати свій код при кожному виконанні, роблячи сигнатурне виявлення практично неможливим.

Атаки на рівні додатків та складні багатоетапні атаки (Advanced Persistent Threats) часто залишаються непоміченими традиційними IDS, оскільки їхні окремі компоненти можуть виглядати як нормальна активність. Зловмисники використовують техніки "низької та повільної" атаки, розподіляючи шкідливу активність у часі та використовуючи легітимні протоколи та інструменти, що ускладнює виявлення. Крім того, існують спеціалізовані техніки обходу IDS, такі як фрагментація пакетів, обфускація, тунелювання та використання альтернативних кодувань, які дозволяють зловмисникам уникати виявлення.

Ефективна робота IDS вимагає високої кваліфікації персоналу та значних інвестицій у навчання. Інтерпретація результатів роботи системи, аналіз складних атак та налаштування правил виявлення потребують глибоких знань у галузі мережевої безпеки, протоколів та методів атак. Дефіцит кваліфікованих фахівців з кібербезпеки означає, що багато організацій не можуть повною мірою використовувати можливості своїх систем виявлення вторгнень. Неправильне налаштування або відсутність регулярного обслуговування може перетворити IDS на джерело хибних тривог або, що ще гірше, на систему з хибним відчуттям безпеки.

Інтеграція IDS з іншими компонентами інфраструктури безпеки також представляє значні виклики. Для ефективної роботи IDS повинна інтегруватися з SIEM-системами, фаєрволами, системами управління інцидентами та іншими інструментами безпеки. Відсутність стандартизації форматів даних, різноманітність вендорів та складність налаштування міжсистемної взаємодії часто призводять до створення розрізнених систем безпеки, які не можуть ефективно обмінюватися інформацією та координувати відповіді на інциденти.

Перехід до хмарних інфраструктур та гібридних мереж створює додаткові виклики для традиційних IDS. Віртуалізація та динамічність хмарних середовищ означають, що мережева топологія постійно змінюється, робочі навантаження мігрують між різними фізичними серверами, а трафік може обходити традиційні точки моніторингу. У мультихмарних середовищах організації часто втрачають повну видимість свого трафіку, оскільки частина комунікацій відбувається безпосередньо між хмарними сервісами постачальників. Контейнерні технології, такі як Docker та Kubernetes, створюють високодинамічні середовища з короткоживучими робочими навантаженнями, які важко моніторити традиційними методами.

Розподілені архітектури та периметр, що розчиняється, також ставлять під сумнів ефективність класичного підходу IDS, орієнтованого на моніторинг мережевого периметра. З розвитком віддаленої роботи, мобільних пристроїв та прямого доступу до хмарних ресурсів, концепція єдиного мережевого периметру втрачає актуальність. Це вимагає переосмислення архітектури виявлення вторгнень та переходу до розподілених моделей безпеки з множинними точками моніторингу.

Впровадження та підтримка ефективної системи виявлення вторгнень вимагає значних фінансових інвестицій, які виходять далеко за межі початкової вартості придбання. Організації повинні враховувати витрати на апаратне забезпечення або хмарні ресурси, ліцензії, регулярні оновлення, навчання персоналу та постійну підтримку. Для малих та середніх підприємств ці витрати можуть бути непропорційно високими порівняно з їхніми бюджетами безпеки. Крім того, хибні спрацювання мають свою економічну ціну у вигляді

витраченого часу персоналу на розслідування неіснуючих інцидентів, що знижує загальну рентабельність інвестицій у систему безпеки.

Незважаючи на всі обмеження та проблеми, системи виявлення вторгнень залишаються критично важливим компонентом сучасної інфраструктури безпеки. Розуміння їхніх обмежень дозволяє організаціям реалістично оцінювати можливості захисту та розробляти багаторівневі стратегії безпеки, які компенсують слабкі місця IDS іншими засобами захисту. Майбутнє систем виявлення вторгнень пов'язане з впровадженням штучного інтелекту та машинного навчання, які обіцяють вирішити деякі з існуючих проблем, хоча й привносять нові виклики. Ключем до ефективного захисту є комбінування різних технологій, постійне вдосконалення та адаптація до мінливого ландшафту загроз.

РОЗДІЛ 2 ПІДХОДИ ДО ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ СИСТЕМ ВИЯВЛЕННЯ ВТОРГНЕНЬ

Зростання складності сучасних інформаційних систем, широке використання розподілених мережевих архітектур та постійна еволюція кіберзагроз зумовлюють необхідність удосконалення систем виявлення вторгнень. Традиційні підходи до побудови IDS, які ґрунтуються на статичних правилах або сигнатурах відомих атак, дедалі частіше виявляються недостатньо ефективними в умовах появи нових, раніше невідомих методів атак, зокрема атак нульового дня та складних багатоступеневих вторгнень. У зв’язку з цим актуальним завданням є дослідження та порівняння різних підходів, спрямованих на підвищення точності, адаптивності та надійності систем виявлення вторгнень.

Під ефективністю систем виявлення вторгнень зазвичай розуміють здатність своєчасно ідентифікувати шкідливу активність за мінімального рівня хибних спрацювань, а також забезпечувати прийнятну швидкодію та масштабованість у реальних умовах експлуатації. Досягнення таких характеристик можливе лише за рахунок застосування комплексних підходів до аналізу подій безпеки, які враховують різні рівні функціонування інформаційних систем, зокрема мережевий трафік, поведінку хостів та дії користувачів.

У сучасних IDS активно використовуються підходи, що базуються на експертних системах, методах машинного навчання, статистичному аналізі та поведінковому моделюванні. Кожен із цих підходів має власні особливості, сильні та слабкі сторони, що визначає доцільність їх застосування залежно від конкретних умов та вимог до системи безпеки. Експертні системи забезпечують інтерпретованість результатів та чітке формалізоване представлення знань про загрози, проте обмежені у здатності адаптуватися до нових типів атак. Натомість методи машинного навчання дозволяють автоматично виявляти складні закономірності у великих обсягах даних, але можуть потребувати значних обчислювальних ресурсів і якісних навчальних вибірок.

Окрему увагу в межах підвищення ефективності IDS приділяють гібридним підходам, які поєднують кілька методів аналізу з метою компенсації недоліків кожного з них. Такі рішення дозволяють підвищити точність виявлення атак, зменшити кількість хибних позитивних спрацювань та забезпечити кращу адаптацію до динамічних умов функціонування мережі. Крім того, важливим напрямом розвитку IDS є інтеграція з іншими системами забезпечення інформаційної безпеки, зокрема SIEM-платформами, що дає змогу здійснювати кореляцію подій та комплексний аналіз інцидентів.

Таким чином, дослідження підходів до підвищення ефективності систем виявлення вторгнень є необхідною передумовою для створення надійних механізмів захисту інформаційних ресурсів. У цьому розділі розглядаються основні методи та технології, що застосовуються в сучасних IDS, а також аналізуються їхні можливості щодо підвищення рівня безпеки інформаційних систем в умовах постійно зростаючих кіберзагроз.

2.1 Підхід на основі методів машинного навчання

2.1.1 Використання машинного навчання для підвищення точності виявлення

Машинне навчання революціонізувало підхід до виявлення вторгнень, надаючи системам здатність автоматично вчитися на даних та адаптуватися до нових загроз без постійного втручання експертів. На відміну від традиційних сигнатурних систем, які потребують ручного створення правил для кожного типу атаки, ML-підходи можуть автоматично виявляти складні патерни та залежності в мережевому трафіку, які важко формалізувати за допомогою явних правил. Основною перевагою машинного навчання є здатність виявляти не лише відомі атаки, але й їх модифікації, а також потенційно нові типи загроз, які демонструють аномальну поведінку порівняно з нормальною активністю мережі.Застосування машинного навчання в IDS охоплює три основні типи задач. По-перше, це задачі бінарної класифікації, де система повинна визначити,

чи є конкретна активність нормальною або зловмисною. По-друге, багатокласова класифікація дозволяє не лише виявити атаку, але й визначити її конкретний тип, що критично важливо для вибору відповідних заходів протидії. По-третє, виявлення аномалій, яке фокусується на ідентифікації будь-якої незвичайної поведінки, що відхиляється від встановлених норм, навіть якщо конкретний тип атаки невідомий. Кожен з цих підходів має свої переваги та обмеження, і вибір конкретного методу залежить від специфіки мережевого середовища та типів загроз, які необхідно виявляти. Однак впровадження машинного навчання в IDS супроводжується рядом викликів. Ключовою проблемою є необхідність великих обсягів якісних навчальних даних, які повинні бути належним чином розмічені та збалансовані. Дисбаланс класів, коли нормальний трафік значно переважає зловмисну активність, може призвести до того, що модель буде добре виявляти нормальну поведінку, але пропускатиме рідкісні атаки. Крім того, ML-моделі схильні до явища "перенавчання", коли система надто добре запам'ятовує навчальні дані, але погано узагальнює на нові приклади. Ще одним важливим аспектом є інтерпретованість рішень – багато сучасних ML-алгоритмів працюють як "чорні скриньки", що ускладнює розуміння причин, чому конкретна активність була класифікована як атака.

2.1.2 Навчання з вчителем

Навчання з вчителем є найпоширенішим підходом у застосуванні машинного навчання для виявлення вторгнень, оскільки воно дозволяє використовувати існуючі знання про типи атак для побудови точних класифікаторів. Основна ідея полягає в навчанні моделі на розміченому наборі даних, де кожен приклад мережевого трафіку або системної події супроводжується міткою, що вказує, чи є він нормальним або відноситься до конкретного типу атаки. Після навчання модель здатна класифікувати новий, раніше не бачений трафік, застосовуючи виявлені закономірності.

Дослідження показують, що серед моделей машинного навчання найбільш ефективними є моделі випадкових лісів та Gradient Boosting методи, такі як XGBoost та LightGBM [8-10].

Випадковий ліс є ансамблевим методом машинного навчання, який будує множину дерев рішень під час навчання та об'єднує їх прогнози для отримання фінального результату. Основна ідея полягає в створенні "лісу" з багатьох дерев, де кожне дерево навчається на випадковій підвибірці даних (bootstrap sample) та використовує випадкову підмножину ознак у кожному вузлі для прийняття рішення про розділення. Ця подвійна випадковість – у даних та ознаках – забезпечує різноманітність дерев, що значно підвищує загальну точність та робастність моделі. На рисунку 2.1 наведено принпип ансамблювання та ухвалення остаточного рішення у випадкових лісах, що отримав назву беггінгу.

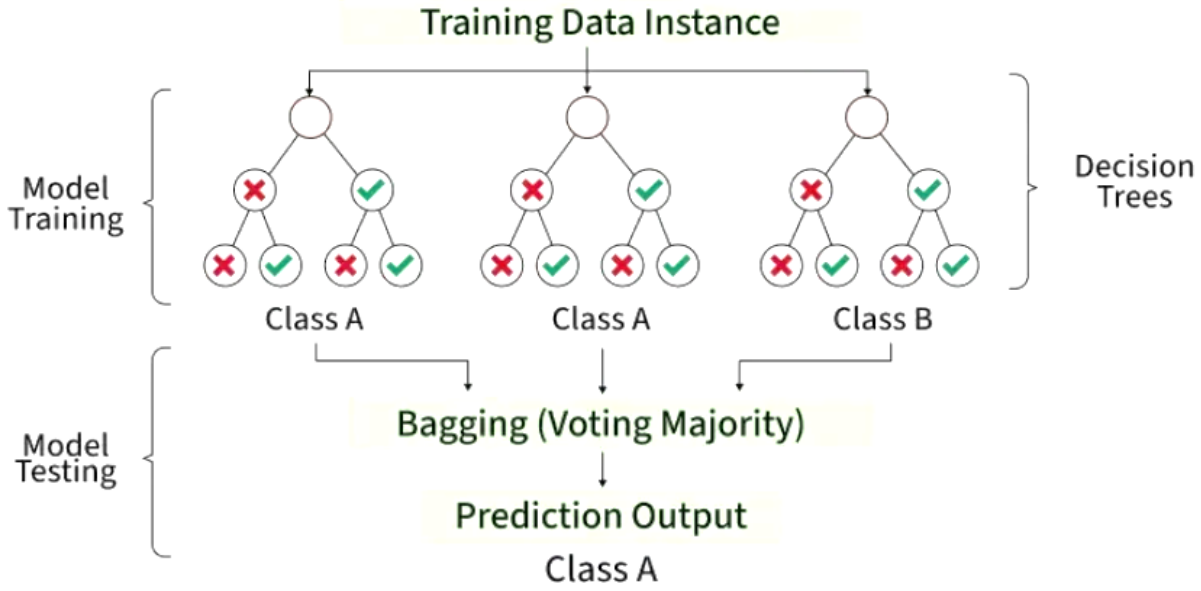

Рисунок 2.1 – Принцип ухвалення рішення в випадкових лісах

У контексті систем виявлення вторгнень Random Forest демонструє кілька ключових переваг. По-перше, метод природним чином обробляє дисбаланс класів через механізм зважування та можливість налаштування параметрів вибірки. По-друге, алгоритм надає важливу інформацію про значущість ознак (feature importance), що дозволяє зрозуміти, які характеристики мережевого трафіку найбільше вказують на атаки – це може бути розмір пакетів, частота

з'єднань, порти призначення тощо. По-третє, Random Forest добре справляється з шумовими даними та виявляє нелінійні залежності без необхідності попередньої обробки даних. Алгоритм також може ефективно працювати з великими наборами даних та високою розмірністю, що типово для мережевого трафіку.Для класифікації фінальне рішення приймається голосуванням: кожне дерево "голосує" за певний клас, і обирається клас з найбільшою кількістю голосів. Це усереднення прогнозів множини дерев значно зменшує ризик перенавчання, характерний для окремих глибоких дерев рішень. Random Forest також може надавати ймовірності належності до кожного класу, що корисно для налаштування порогів виявлення та ранжування загроз за рівнем ризику. Типові параметри для налаштування включають кількість дерев у лісі (зазвичай 100-500), максимальну глибину дерев, мінімальну кількість зразків для розділення вузла та кількість ознак для розгляду в кожному розділенні.

Іншим підходом до ансамлювання є бустинг, де моделі будуються послідовно, а кожна наступна модель навчається виправляти помилки попередніх. На відміну від Random Forest, де дерева будуються незалежно та паралельно, у boosting кожне нове дерево додається до ансамблю таким чином, щоб мінімізувати функцію втрат усього ансамблю. Алгоритм використовує градієнтний спуск у функціональному просторі, де на кожній ітерації будується нове дерево, яке апроксимує антиградієнт функції втрат. Це дозволяє моделі поступово покращувати свої прогнози, особливо на складних для класифікації прикладах.

У системах виявлення вторгнень Gradient Boosting особливо ефективний для виявлення рідкісних та складних атак, оскільки механізм послідовного навчання дозволяє моделі зосередитися на важких прикладах, які були неправильно класифіковані на попередніх ітераціях. Це призводить до високої точності та чутливості навіть для нових варіантів відомих атак. Однак базовий GBM має свої обмеження: схильність до перенавчання при надто великій кількості ітерацій, відносно повільне навчання та чутливість до шумових даних та викидів.

XGBoost є оптимізованою та розширеною реалізацією gradient boosting, яка вирішує багато проблем базового алгоритму та демонструє виняткову

продуктивність у різноманітних задачах машинного навчання. Ключові інновації XGBoost включають вбудовану регуляризацію (L1 та L2), яка запобігає перенавчанню ю через штрафування складності моделі. Алгоритм використовує другу похідну функції втрат (Hessian), що прискорює збіжність та підвищує точність оптимізації.

XGBoost ефективно обробляє відсутні значення, автоматично вивчаючи, в який напрямок направляти зразки з пропущеними даними в кожному розділенні дерева. Це особливо цінно для IDS, оскільки мережеві дані часто містять неповну інформацію через втрату пакетів або обмеження моніторингу. Алгоритм також підтримує паралельну обробку на рівні побудови окремих дерев, що значно прискорює навчання на багатоядерних процесорах. Додаткові оптимізації включають cache-aware доступ до даних та out-of-core обчислення для роботи з датасетами, що не вміщуються в пам'ять.

У контексті виявлення вторгнень XGBoost часто демонструє найвищі показники точності серед традиційних методів машинного навчання. Алгоритм добре справляється з високорозмірними даними, автоматично виконує відбір ознак через механізм розділення та надає детальну інформацію про важливість кожної характеристики трафіку. XGBoost також підтримує різні цільові функції, включаючи спеціалізовані для задач з дисбалансом класів, що дозволяє налаштувати модель для максимальної чутливості до атак при контрольованому рівні хибних спрацювань.

Параметри XGBoost, які особливо важливі для IDS, включають: learning rate (швидкість навчання, що контролює внесок кожного дерева), max_depth (максимальна глибина дерев, що балансує між складністю моделі та перенавчанню ям), subsample (частка даних для навчання кожного дерева), colsample_bytree (частка ознак для кожного дерева), min_child_weight (мінімальна сума ваг для створення листа, що допомагає з дисбалансом класів) та scale_pos_weight (вага позитивного класу для компенсації дисбалансу). Регуляризаційні параметри lambda (L2) та alpha (L1) контролюють складність моделі та запобігають перенавчанню.

LightGBM, розроблений Microsoft, представляє ще більш оптимізовану реалізацію gradient boosting, спеціально призначену для роботи з великими обсягами даних та високою розмірністю. Основна інновація LightGBM – використання стратегії росту дерев leaf-wise замість традиційної level-wise стратегії. У level-wise підході всі вузли на одному рівні дерева розширюються одночасно, що призводить до збалансованих, але часто неефективних дерев. Leaf-wise стратегія обирає для розділення лист з найбільшим зменшенням втрат, що призводить до асиметричних, але значно ефективніших дерев, які досягають кращої точності з меншою кількістю листів.

LightGBM використовує два критичні алгоритми оптимізації: Gradient-based One-Side Sampling (GOSS) та Exclusive Feature Bundling (EFB). GOSS зберігає всі зразки з великими градієнтами (великими помилками) та випадкову вибірку зразків з малими градієнтами, що значно зменшує обсяг даних без суттєвої втрати точності, оскільки найбільший внесок у навчання роблять саме важкі приклади. EFB об'єднує взаємовиключні ознаки (які рідко одночасно мають ненульові значення) у "пучки", що ефективно зменшує розмірність даних. Це особливо корисно для мережевих даних, де багато бінарних та категоріальних ознак є розрідженими.

Для систем виявлення вторгнень LightGBM особливо привабливий завдяки своїй швидкості та ефективності використання пам'яті, що критично важливо для обробки великих обсягів мережевого трафіку в реальному часі. Алгоритм може навчатися на датасетах, що містять мільйони записів та тисячі ознак, за лічені хвилини. LightGBM також має вбудовану підтримку категоріальних ознак без необхідності попереднього кодування (one-hot encoding), що зберігає пам'ять та часто покращує точність. Це особливо цінно для мережевих даних, які містять багато категоріальних полів, таких як протоколи, типи сервісів, флаги тощо.

LightGBM демонструє відмінну продуктивність на задачах з дисбалансом класів завдяки параметрам is_unbalance та scale_pos_weight, а також підтримує різні метрики оцінки, включаючи AUC, F1-score та custom metrics, що дозволяє оптимізувати модель безпосередньо для специфічних вимог IDS. Алгоритм

також підтримує distributed training для навчання на кластерах машин, що робить його масштабованим рішенням для корпоративних IDS.

2.1.3 Навчання без вчителя

Навчання без вчителя відіграє критично важливу роль у виявленні невідомих та нових типів атак, оскільки не потребує попередньо розмічених даних. Замість того, щоб вчитися розпізнавати конкретні типи відомих загроз, ці алгоритми аналізують структуру даних та виявляють аномальні патерни, які значно відрізняються від типової поведінки. Це особливо цінно для виявлення zero-day експлойтів та складних цільових атак, які не мають відомих сигнатур. Математичною основою більшості методів unsupervised learning є оптимізація певної цільової функції, яка вимірює якість групування або відхилення від нормальної поведінки.

K-means є найпоширенішим алгоритмом кластеризації, який розділяє n спостережень на k кластерів, мінімізуючи внутрішньокластерну варіацію. Алгоритм ітеративно призначає кожну точку даних найближчому центроїду та перераховує центроїди на основі призначених точок.

Цільова функція K-means визначається як:

$$J = \sum_{i=1}^{K} \sum_{x \in C_i} \|x - \mu_i\|^2, \quad (2.1)$$

де K -кількість кластерів,

C_i – і-ий кластер,

μ_i – центроїд і-ого кластера,

x – точка даних,

$\|x - \mu_i\|$ – Евклідова відстань між точкою і центроїдом

Алгоритм працює наступним чином:

1. Випадково обирається K випадкових центроїдів серед об’єктів вибірки.
2. Кожна точка призначається найближчому центроїду.
3. Центроїди перераховуються як середнє всіх точок у кластері.

4. Кроки2-3 повторюються до збіжності.

У контексті IDS, після кластеризації нормального трафіку, нові зразки оцінюються за відстанню до найближчого центроїду. Якщо ця відстань перевищує заданий поріг θ, зразок класифікується як потенційна атака. Для IDS типово використовують kв діапазоні 5-20 кластерів для моделювання різних типів нормальної активності (веб-трафік, email, FTP тощо).

Основні обмеження K-means включають необхідність заздалегідь визначати кількість кластерів, чутливість до вибору початкових центроїдів (вирішується методом K-means++) та припущення про сферичну форму кластерів. Для визначення оптимального k часто використовують метод ліктя (elbow method), аналізуючи графік залежності внутрішньокластерної суми квадратів від кількості кластерів.

DBSCAN представляє альтернативний підхід, заснований на щільності точок, який автоматично визначає кількість кластерів та ідентифікує outliers. Алгоритм використовує два параметри: радіус околиці та мінімальну кількість точок для утворення щільного регіону. DBSCAN класифікує точки на три типи: ядрові точки (core points), які мають принаймні MinPts сусідів в межах epsilon-радіуса і формують центри кластерів; прикордонні точки (border points), які знаходяться в околиці ядрових точок, але самі не є ядровими; та шумові точки або outliers, які не належать до жодної з перших двох категорій. Кластери формуються шляхом об'єднання ядрових точок, які є досяжними одна від одної через ланцюги інших ядрових точок, разом з їхніми прикордонними точками.

Для систем виявлення вторгнень DBSCAN особливо цінний тим, що автоматично виявляє outliers – точки, які не належать до жодного щільного кластера нормального трафіку. Ці outliers є природними кандидатами на аномалії та потенційні атаки. Основні переваги DBSCAN включають автоматичну ідентифікацію аномалій, здатність виявляти кластери довільної форми (на відміну від K-means, який припускає сферичні кластери), відсутність необхідності попередньо визначати кількість кластерів та стійкість до шуму в даних. Вибір параметрів epsilon та MinPts є критичним для ефективності алгоритму – типовий підхід передбачає аналіз k-distance графіку, де для кожної

точки обчислюється відстань до k-го найближчого сусіда, і значення epsilon обирається в точці різкої зміни (точці "ліктя") на відсортованому графіку цих відстаней. У контексті IDS, після кластеризації нормального трафіку за допомогою DBSCAN, нові зразки оцінюються за їх відстанню до найближчих щільних регіонів, і якщо зразок класифікується як викид або знаходиться на значній відстані від усіх кластерів, він позначається як потенційна загроза.

2.1.4 Глибокі нейронні мережі

Глибоке навчання (Deep Learning) представляє найсучасніший напрямок у застосуванні штучного інтелекту для виявлення вторгнень, демонструючи здатність автоматично вивчати ієрархічні представлення з сирих даних без необхідності ручного конструювання ознак. Глибокі нейронні мережі можуть виявляти складні нелінійні залежності та абстрактні патерни в мережевому трафіку, які неможливо ефективно описати традиційними методами. Багаторівнева архітектура цих мереж дозволяє нижнім шарам вивчати прості ознаки (наприклад, базові статистики пакетів), середнім – комбінації цих ознак (патерни послідовностей), а вищим – складні абстрактні концепції (характеристики специфічних типів атак), що імітує ієрархічне сприйняття в біологічних системах. Фундаментальна перевага глибокого навчання полягає в автоматичному feature learning – здатності мережі самостійно виявляти релевантні характеристики даних на різних рівнях абстракції. Замість того, щоб експерти вручну визначали, які ознаки мережевого трафіку є важливими (кількість пакетів, розміри, часові інтервали тощо), глибока мережа навчається оптимальним представленням безпосередньо з даних. Це особливо цінно для виявлення складних та еволюціонуючих атак, де традиційні hand-crafted ознаки можуть бути недостатніми. Однак впровадження глибокого навчання супроводжується викликами: необхідність великих обсягів навчальних даних, значні обчислювальні ресурси, складність налаштування гіперпараметрів та проблема інтерпретованості рішень, що критично важливо для систем безпеки.

Згорткові нейронні мережі (Convolutional Neural Networks, CNN), спочатку розроблені для обробки зображень, виявилися надзвичайно ефективними для аналізу мережевого трафіку завдяки своїй здатності виявляти локальні патерни та структурні залежності в даних. Ключова ідея полягає в представленні мережевих даних у форматі, придатному для згорткової обробки. Послідовності пакетів або байти payload можуть бути організовані як двовимірні матриці, де один вимір представляє часові кроки або послідовність пакетів, а інший – різні ознаки кожного пакету (розмір, час прибуття, TCP флаги, порти, тощо). Альтернативно, сирі байти мережевого трафіку можуть оброблятися як одновимірні послідовності, подібно до текстових даних в обробці природної мови.

Архітектура CNN складається з декількох типів шарів, кожен з яких виконує специфічну функцію. На рисунку 2.2 наведено типову структуру згорткової мережі

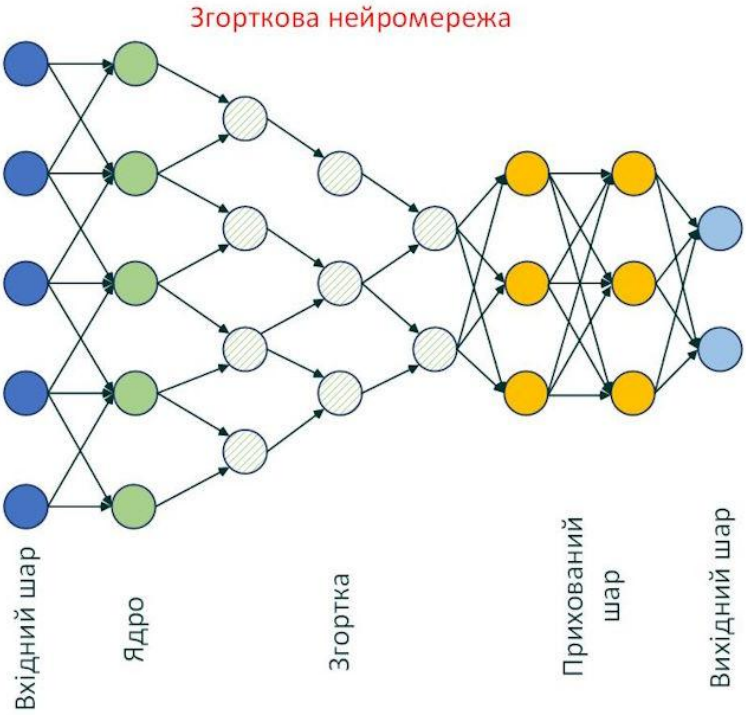

Рисунок 2.2 – Архітектура згорткової мережі

Ядро в згортковій нейронній мережі виконує згортку вхідних даних, виділяючи локальні ознаки. Згортковий шар (Convolutional Layer) застосовує набір навчальних фільтрів (kernels) до вхідних даних, виявляючи локальні

патерни. Математично, операція згортки для одновимірного випадку визначається як:

$$\left(x * w)(t) = \sum_{i=1}^{k} x\,(t+i) \cdot w(i\right) \tag{2.2}$$

де x– вхідна послідовність, w– ваги фільтра розміру k, t– позиція в послідовності. Кожен фільтр "ковзає" по вхідних даних і обчислює скалярний добуток з локальним регіоном, створюючи feature map – карту активацій, яка показує, де в даних присутній певний патерн. Множина фільтрів в одному шарі дозволяє виявляти різні типи патернів одночасно.

Pooling шар знаходиться між згортковими шарами та прихованим шаром та зменшує просторову розмірність feature maps, зберігаючи найважливіші характеристики та забезпечуючи інваріантність до невеликих зміщень у даних. Max pooling обирає максимальне значення в кожному регіоні, що відповідає найсильнішій активації виявленого патерна. Average pooling обчислює середнє значення, що дає більш плавне представлення. Fully connected шари в кінці мережі інтегрують виявлені локальні ознаки для прийняття фінального рішення про класифікацію. Функції активації (ReLU, sigmoid, tanh) вносять нелінійність, дозволяючи мережі моделювати складні залежності. Прихований шар обробляє ці виділені ознаки, комбінує їх і формує більш абстрактні представлення, які використовуються для подальшого розпізнавання або класифікації.

У контексті виявлення вторгнень CNN демонструють кілька переваг. По-перше, згорткові фільтри автоматично вивчають релевантні патерни пакетів, характерні для різних типів атак. Наприклад, DDoS атаки можуть характеризуватися специфічними послідовностями великої кількості коротких пакетів, SQL injection – певними байтовими патернами в payload, а сканування портів – характерною послідовністю спроб з'єднання. По-друге, CNN ефективно обробляють сирі байти трафіку, уникаючи необхідності ручного feature engineering. Дослідження показали, що CNN можуть навіть виявляти зашифрований зловмисний трафік через аналіз метаданих пакетів та

статистичних характеристик потоків. По-третє, ієрархічна природа CNN дозволяє нижнім шарам виявляти прості патерни (окремі TCP флаги, характерні розміри), а вищим – складні комбінації, що відповідають складним багатоетапним атакам.

Типова архітектура CNN для IDS може включати: вхідний шар, що приймає матрицю розміром (число_пакетів × число_ознак), декілька блоків (Convolutional Layer → ReLU → Pooling), що поступово зменшують розмірність та збільшують глибину представлення, flatten шар для перетворення в одновимірний вектор, один або декілька fully connected шарів та вихідний softmax шар для багатокласової класифікації типів атак. Regularization техніки, такі як dropout (випадкове вимикання нейронів під час навчання) та batch normalization, допомагають запобігти перенавчанню та прискорюють збіжність.

Рекурентні нейронні мережі (Recurrent Neural Networks, RNN) спеціально призначені для роботи з послідовними даними та часовими рядами, що робить їх ідеальними для аналізу мережевого трафіку як послідовності подій у часі. На відміну від традиційних feedforward мереж, RNN мають рекурентні з'єднання, які дозволяють інформації циркулювати в мережі, створюючи внутрішню "пам'ять". Це означає, що мережа може враховувати попередній контекст при обробці поточних даних, що критично важливо для виявлення атак, розподілених у часі.

Базова RNN має прихований стан h_t, який оновлюється на кожному часовому кроці на основі поточного входу x_tта попереднього стану h_{t-1}:

$$\begin{aligned} h_t &= \tanh\left(W_{hh}h_{t-1} + W_{xh}x_t + b_h\right) \\ y_t &= W_{hy}h_t + b_y \end{aligned} \tag{2.3}$$

де W_{hh}, W_{xh}, W_{hy}– матриці ваг, b_h, b_y– вектори зміщень, $\tanh$ – функція активації (гіперболічний тангенс). Прихований стан h_tакумулює інформацію з попередніх кроків і передається далі, а вихід y_tвикористовується для прогнозування або класифікації.

Однак базові RNN страждають від проблеми зникаючого або вибухаючого градієнта при навчанні на довгих послідовностях. Під час backpropagation

through time (BPTT) градієнти, що проходять через багато часових кроків, можуть експоненційно зменшуватися (зникати) або збільшуватися (вибухати), роблячи неможливим навчання довгострокових залежностей. Це означає, що базова RNN може запам'ятовувати лише короткострокову історію (типово 5-10 кроків), що недостатньо для виявлення складних атак.

LSTM (Long Short-Term Memory) вирішує цю проблему через спеціальну архітектуру з воротами (gates), які контролюють потік інформації. LSTM містить cell state C_t– конвеєр інформації, що проходить через усю послідовність з мінімальними лінійними перетвореннями, що дозволяє градієнтам ефективно поширюватися назад у часі.

Cell state оновлюється як комбінація забування старої інформації та додавання нової:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{2.4}$$

де ⊙позначає поелементне множення (Hadamard product). Прихований стан обчислюється як:

$$h_t = o_t \odot \tanh(C_t) \tag{2.5}$$

Тут σ– сигмоїдна функція, що дає значення від 0 до 1, контролюючи, скільки інформації пропускається (0 = блокувати все, 1 = пропустити все).

GRU (Gated Recurrent Unit) є спрощеним варіантом LSTM з меншою кількістю параметрів, що робить його швидшим у навчанні та часто настільки ж ефективним. GRU об'єднує forget та input gates в один update gate та об'єднує cell state і прихований стан. Це зменшує обчислювальну складність, зберігаючи здатність моделювати довгострокові залежності.

У контексті виявлення вторгнень LSTM та GRU мають кілька критичних застосувань. По-перше, вони можуть відстежувати послідовності подій протягом тривалого часу, виявляючи багатоетапні атаки (multi-stage attacks), де окремі дії можуть виглядати нормально, але їх послідовність вказує на зловмисну

активність. Наприклад, advanced persistent threat (APT) може починатися з reconnaissance (сканування портів), потім exploitation (використання вразливості), далі privilege escalation (підвищення привілеїв), installation (встановлення backdoor) та exfiltration (викрадення даних) – весь цикл може тривати години або дні. По-друге, LSTM/GRU ефективні для виявлення повільних атак, таких як slow DoS (Slowloris), де зловмисник підтримує з'єднання відкритими, надсилаючи дані дуже повільно. По-третє, вони можуть моделювати нормальну поведінку користувачів та систем як послідовності дій, виявляючи аномалії через відхилення від звичних патернів.

Типова архітектура для IDS може використовувати bidirectional LSTM, де дві LSTM обробляють послідовність у прямому та зворотному напрямках, дозволяючи мережі враховувати як попередній, так і наступний контекст. Stacked LSTM (багатошарова) створює ієрархію, де кожен шар вивчає представлення на вищому рівні абстракції. Attention mechanism може бути доданий для фокусування на найбільш релевантних частинах послідовності, що покращує як точність, так і інтерпретованість рішень.

Автоенкодери представляють потужний підхід unsupervised deep learning для виявлення аномалій, навчаючись стискати дані в компактне представлення та відновлювати їх [11-12]. Автоенкодер складається з двох частин: encoder, який перетворює вхідні дані xв латентне представлення zменшої розмірності, та decoder, який відновлює дані з латентного представлення до $\hat{x}$. Архітектура зазвичай симетрична, нагадуючи "пісочний годинник" або "пляшкове горло", де найвужче місце – латентний простір.

Для простого автоенкодера:

- Encoder: $z = f(x) = \sigma(W_e x + b_e)$
- Decoder: $\hat{x} = g(z) = \sigma(W_d z + b_d)$

Цільова функція мінімізує reconstruction error між вхідними та відновленими даними:

$$L(x, \hat{x}) = || x - \hat{x} ||^2 = \sum_{i=1}^{d} (x_i - \hat{x}_i)^2 \quad (2.6)$$

для безперервних даних, або бінарну крос-ентропію для бінарних даних:

$$L(x, \hat{x}) = -\sum_{i=1}^{d} [x_i \log (\hat{x}_i) + (1 - x_i) \log (1 - \hat{x}_i) 7 \quad (2.7)$$

Ключова ідея для виявлення вторгнень: автоенкодер навчається виключно на нормальному трафіку. Він вивчає компактне представлення типових патернів мережевої активності. Нормальний трафік, схожий на навчальні дані, буде добре відтворюватися (низька помилка реконструкції). Аномальний трафік або атаки, які мають інші статистичні характеристики та патерни, не будуть ефективно закодовані та відновлені (висока помилка реконструкції).

Глибокі автоенкодери використовують множину прихованих шарів в encoder та decoder, що дозволяє вивчати більш складні, ієрархічні представлення. Нижні шари encoder вивчають прості ознаки (базові статистики пакетів), середні – комбінації ознак (патерни з'єднань), вищі – абстрактні концепції (типи нормальної активності). Decoder відновлює інформацію в зворотному порядку. Stacked autoencoders навчаються пошарово: спочатку перший шар encoder-decoder, потім другий використовує вихід першого encoder як вхід, і так далі. Після greedy layer-wise pretraining вся мережа fine-tuning разом end-to-end.

Denoising autoencoders навчаються відновлювати чисті дані з зашумлених версій, штучно додаючи шум до входу під час навчання. Це змушує мережу вивчати більш робастні представлення, що не залежать від випадкових флуктуацій. Для IDS це означає стійкість до шуму в мережевих даних, неповної інформації та варіативності нормального трафіку.

Sparse autoencoders додають regularization term до функції втрат, що заохочує розрідженість активацій в латентному шарі (більшість нейронів неактивні). Це змушує мережу вивчати більш значущі та диференційовані ознаки:

$$L_{total} = L_{reconstruction} + \beta \sum_{j=1}^{m} KL(\rho \,||\, \hat{\rho}_j) \quad (2.8)$$

де ρ– цільовий рівень розрідженості, $\hat{\rho}_j$– середня активація нейрона j, KL– дивергенція Кульбака-Лейблера, β– вага regularization.

2.1.5 Проблеми застосування методів машинного навчання

Застосування машинного навчання для виявлення вторгнень, незважаючи на його потенціал, супроводжується рядом серйозних проблем, які можуть значно обмежити ефективність систем у реальних умовах. Однією з фундаментальних проблем є потреба у великих обсягах якісних навчальних даних. ML-моделі, особливо глибокі нейронні мережі, потребують тисяч або навіть мільйонів прикладів для ефективного навчання. Однак у сфері кібербезпеки отримання достатньої кількості розмічених даних є надзвичайно складним завданням. Розмітка мережевого трафіку вимагає експертизи спеціалістів з безпеки, що робить процес трудомістким і дорогим. Крім того, приватність та конфіденційність даних обмежують можливість обміну інформацією між організаціями, що ускладнює створення великих публічних датасетів.

Швидка еволюція загроз створює проблему застарівання моделей. Зловмисники постійно розробляють нові методи атак та модифікують існуючі для обходу систем виявлення. Модель, навчена на даних минулого року, може бути неефективною проти сучасних загроз. Це вимагає постійного перенавчання моделей, що потребує свіжих даних та обчислювальних ресурсів. Adversarial attacks – спеціально створені зразки, які обманюють ML-моделі, – представляють додаткову загрозу. Зловмисники можуть навмисно модифікувати свої атаки таким чином, щоб вони класифікувалися як нормальний трафік, експлуатуючи вразливості ML-алгоритмів.

Проблема інтерпретованості є критичною для систем безпеки. Багато сучасних ML-моделей, особливо глибокі нейронні мережі, працюють як "чорні скриньки" – вони можуть давати точні прогнози, але не пояснюють, чому конкретний трафік був класифікований як атака. Для безпекових аналітиків важливо розуміти причини спрацювання системи для ефективного реагування на інциденти та розслідування. Відсутність пояснень також ускладнює налаштування та покращення системи. Хоча існують методи explainable AI (LIME, SHAP), вони додають обчислювальну складність та не завжди надають зрозумілі пояснення для складних моделей.

Обчислювальна складність сучасних ML-моделей може бути перешкодою для роботи в реальному часі. Глибокі нейронні мережі, особливо CNN та LSTM, вимагають значних обчислювальних ресурсів як для навчання, так і для inference. У високошвидкісних мережах, де потрібно обробляти мільйони пакетів за секунду, складні моделі можуть не встигати аналізувати трафік без затримок. Це створює компроміс між точністю моделі та швидкістю роботи. Використання спеціалізованого апаратного прискорення (GPU, TPU) може частково вирішити проблему, але збільшує вартість впровадження.

Перенавчання (overfitting) залишається постійною проблемою, особливо при роботі з обмеженими наборами даних. Модель може надто добре запам'ятати навчальні дані, включаючи їхній шум та специфічні особливості, але погано узагальнювати на нові приклади. Це призводить до високої точності на навчальних даних, але поганої продуктивності в реальних умовах. Для кібербезпеки це означає пропуск нових варіантів атак або генерацію великої кількості хибних спрацювань на легітимному трафіку, який трохи відрізняється від навчального.

Дисбаланс класів є однією з найсерйозніших проблем застосування машинного навчання в системах виявлення вторгнень і заслуговує особливої уваги. У реальних мережах нормальний трафік складає переважну більшість всіх даних – типово 95-99% або навіть більше, тоді як зловмисна активність становить лише невеликий відсоток. Ця асиметрія створює фундаментальну проблему для

ML-алгоритмів, оскільки більшість стандартних методів розроблені з припущенням приблизно збалансованого розподілу класів.

Коли модель навчається на сильно дисбалансованих даних, вона природним чином схиляється до класу більшості (нормального трафіку), оскільки це мінімізує загальну помилку класифікації. Наприклад, якщо атаки становлять лише 1% даних, "наївна" модель може просто класифікувати весь трафік як нормальний і досягти 99% точності (accuracy). Однак така модель абсолютно марна для виявлення вторгнень, оскільки пропускає всі атаки – саме те, що вона повинна виявляти. Це демонструє, чому accuracy є поганою метрикою для дисбалансованих задач і чому потрібні альтернативні метрики, такі як precision, recall, F1-score та AUC-ROC.

Математично, проблема виникає через те, що функція втрат (loss function) однаково штрафує помилки на обох класах. Для бінарної класифікації стандартна функція втрат:

$$L = -\frac{1}{n}\sum_{i=1}^{n}[\, y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \tag{2.9}$$

де n– загальна кількість зразків. Якщо $n_{normal} \gg n_{attack}$, то внесок помилок на атаках у загальну втрату буде мінімальним, і оптимізація природно фокусується на правильній класифікації нормального трафіку за рахунок атак.

2.2 Підхід на основі експертних систем та нечіткої логіки

Експертні системи є одним із перших інтелектуальних підходів, що були застосовані для підвищення ефективності систем виявлення вторгнень. Вони ґрунтуються на формалізованих знаннях фахівців у галузі інформаційної безпеки та дозволяють автоматизувати процес прийняття рішень щодо виявлення та класифікації атак. Основною ідеєю експертних IDS є використання правил та

логічних висновків для аналізу подій безпеки з метою визначення наявності вторгнення або аномальної активності. [13]

Фундаментальний принцип роботи експертної системи полягає у відокремленні знань від механізму міркування. База знань містить факти та правила предметної області (в даному випадку – характеристики атак, типові патерни зловмисної поведінки, індикатори компрометації), тоді як механізм виведення використовує ці знання для аналізу конкретних ситуацій та прийняття рішень. Така архітектура дозволяє легко оновлювати та розширювати знання системи без зміни базового коду, що критично важливо для адаптації до нових типів загроз. Експертні системи для IDS працюють як інтелектуальні асистенти безпекових аналітиків, автоматизуючи процес аналізу великих обсягів мережевих подій та виявляючи підозрілу активність на основі накопиченої експертизи.

2.2.1 Архітектура експертних систем

Фундаментальний принцип роботи експертної системи полягає у відокремленні знань від механізму міркування. База знань містить факти та правила предметної області (в даному випадку – характеристики атак, типові патерни зловмисної поведінки, індикатори компрометації), тоді як механізм виведення використовує ці знання для аналізу конкретних ситуацій та прийняття рішень. Така архітектура дозволяє легко оновлювати та розширювати знання системи без зміни базового коду, що критично важливо для адаптації до нових типів загроз. На рисунку 2.3 наведено типову структуру експертної системи, основними елементами якої є модуль засвоєння знань, база знань, система логічного виводу та модуль пояснення рішень.

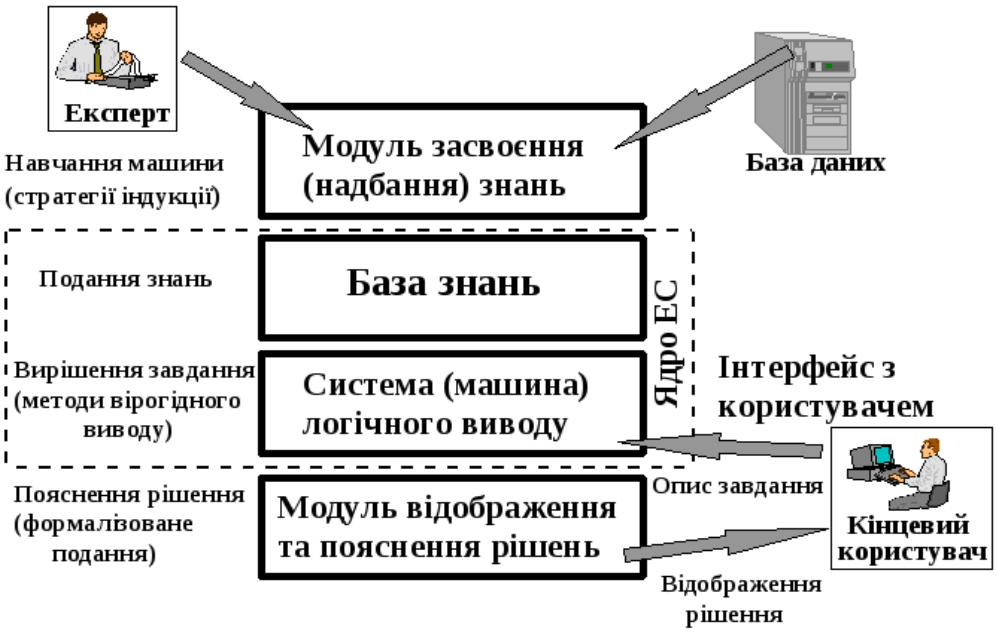

Рисунок 2.3 – Структура ідеальної експертної системи

Модуль засвоєння знань є вхідною точкою для формування бази знань системи Експерт з кібербезпеки використовує стратегії індукції та навчання машини для передачі свого досвіду системі. Цей процес включає:

- Інтерв'ювання експертів для виявлення їхніх методів аналізу інцидентів.
- Формалізацію неявних знань у явні правила виявлення.
- Витягування евристик та контекстних залежностей.
- Опис типових сценаріїв атак та методів їх виявлення.

Наприклад, експерт може описати: "Якщо бачу послідовність SYN пакетів на різні порти одного хоста за короткий час без відповідних ACK, це вказує на SYN-сканування портів". Модуль засвоєння перетворює це на формальне правило:

```
ПРАВИЛО: SYN_Port_Scan
ЯКЩО
  (кількість_SYN_пакетів > 50) І
  (унікальних_портів_призначення > 20) І
  (часовий_інтервал < 60_секунд) І
  (відсутні_відповідні_SYN-ACK) І
  (destination_IP = один_хост)
ТО
  Класифікувати як PORT_SCAN
  Тип = SYN_SCAN
  Рівень_загрози = MEDIUM
```

База знань є центральним сховищем формалізованої експертизи та складається з двох основних типів інформації:

Факти представляють статичні знання про предметну область:

- Характеристики мережевих протоколів (TCP, UDP, ICMP).
- Типові порти сервісів (22-SSH, 80-HTTP, 443-HTTPS).
- Відомі вразливості (CVE бази).
- Репутаційні дані про IP-адреси та домени.
- Топологія мережі організації.
- Довірені системи та користувачі.
- Нормальні патерни активності для різних сервісів.

Правила (продукції) представляють динамічні знання у формі "ЯКЩО ... ТО ...":

```
ПРАВИЛО: Brute_Force_SSH
ЯКЩО
  (протокол = SSH) І
  (невдалих_спроб_входу > 10) І
  (часовий_інтервал < 5_хвилин) І
  (різних_імен_користувачів > 3)
ТО
  Класифікувати як BRUTE_FORCE_ATTACK
  Цільовий_сервіс = SSH
  Рівень_загрози = HIGH
  Рекомендована_дія = BLOCK_SOURCE_IP
```

База знань для IDS може містити сотні або тисячі таких правил, організованих в ієрархічні категорії: мережеві атаки (DDoS, port scanning), атаки на рівні додатків (SQL injection, XSS), інсайдерські загрози, malware активність, порушення політик безпеки тощо.

Механізм логічного виведення є "мозком" експертної системи, що реалізує процес міркування. Він отримує дані про поточну мережеву активність з ЕОМ (електронно-обчислювальної машини) через інтерфейс з користувачем та застосовує правила з бази знань для аналізу ситуації.

Механізм виведення підтримує два основні методи міркування:

- Прямий ланцюг міркувань (Forward Chaining).
- Зворотний ланцюг міркувань (Backward Chaining).

Прямий ланцюг міркувань– data-driven підхід для вирішення завдання:

1. Система отримує факти про мережеву активність (нові з'єднання, пакети, запити).
2. Оцінює, які правила можуть бути застосовані до поточних фактів.
3. Виконує застосовні правила, генеруючи нові факти та висновки.
4. Процес повторюється до виявлення загрози або вичерпання правил.

Цей підхід природно підходить для моніторингу в реальному часі, де нові події постійно надходять та аналізуються.

Зворотний ланцюг міркувань– goal-driven підхід:

1. Система починає з гіпотези (чи є це SQL injection?).
2. Шукає правила, які можуть підтвердити цю гіпотезу.
3. Перевіряє, чи виконуються умови цих правил.
4. Якщо деякі умови невідомі, рекурсивно намагається їх довести.
5. Підтверджує або спростовує гіпотезу.

Цей метод ефективний для глибокого аналізу інцидентів та розслідувань, коли аналітик має підозру про конкретний тип атаки.

Модуль відображення та пояснення рішень забезпечує прозорість роботи системи через формалізоване подання результатів аналізу. Він генерує пояснення рішення. Для прикладу:

```
ПОПЕРЕДЖЕННЯ: SQL Injection виявлено
Джерело: 203.0.113.45:54321
Ціль: webserver.example.com/login.php
Час: 2025-01-15 14:23:17

Ланцюг міркувань:
1. Виявлено HTTP POST запит до /login.php
2. Параметр 'username' містить SQL keywords: "' OR '1'='1"
3. Спрацювало ПРАВИЛО: SQL_Injection_Pattern_1
   Умова 1: Наявність SQL keywords [✓]
   Умова 2: Наявність символів-роздільників [✓]
   Умова 3: Незвичайна довжина параметру [✓]
   Умова 4: Джерело не довірене [✓]
4. Додатково спрацювало ПРАВИЛО: Suspicious_Auth_Attempt
5. Впевненість: 92%

Контекст:
- Цей IP раніше намагався отримати доступ 3 рази за останні 10 хвилин
```

```
- User-Agent відповідає автоматизованим інструментам
- Геолокація: країна з високим рівнем загроз

Рекомендовані дії:
- Заблокувати IP на фаєрволі [AUTO]
- Перевірити логи веб-сервера [MANUAL]
- Оповістити безпекову команду [AUTO]
```

Ця архітектура забезпечує прозору, пояснювальну та адаптивну систему виявлення вторгнень, яка ефективно комбінує людську експертизу з автоматизованим аналізом.

2.2.2 Основні поняття систем нечіткої логіки

Розглянемо кілька базових понять, що лежать в основі систем з нечіткою логікою.

Нечітка множина визначається через функцію належності $\mu(x): X \to [0,1]$, яка відображає кожен елемент x з деякої множини X на інтервал [0, 1]. Загалом нечітку множину $\tilde{X}$ можна подати у вигляді:

$$\tilde{X} = \{(x_1, \mu(x_1)), (x_2, \mu(x_2)), \dots (x_n, \mu(x_n))\}, \qquad n = \overline{1, N}, \ \ \mu(x): X \to [0,1] \tag{2.10}$$

Функції належності визначають форму нечітких множин та можуть мати різні типи. У [14] згадується кілька варіантів функції належності, зокрема трикутна, параболічна, екстрапараболічна. Крім того, популярними функціями є трапецеподібна функція, сигмоїдна, гаусова та інші.

Розглянемо найбільш популярні функції належності. Зокрема трикутна функція описується формулою:

$$(x) = \begin{cases} 1, & x = b \\ \dfrac{x-a}{b-a}, & a < x < b \\ \dfrac{c-x}{c-b}, & b < x < c \\ 0, & \text{в інших випадках} \end{cases} \tag{2.11}$$

На рисунку 2.4 приведено, який вигляд має трикутна функція.

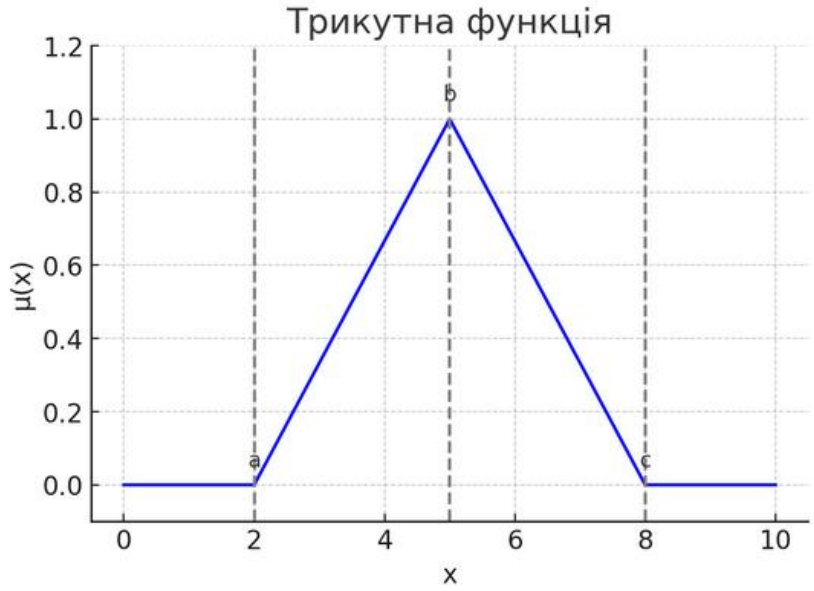

Рисунок 2.4 - Приклад трикутної функції належності

Трапецеподібна функція належності визначається за формулою:

$$\mu(x) = \begin{cases} 1, & b \le x \le c \\ \dfrac{x-a}{b-a}, & a < x < b \\ \dfrac{d-x}{d-c}, & c < x < d \\ 0, & \text{в інших випадках} \end{cases} \tag{2.12}$$

Ілюстрація трапецеподібної функції наведено на рисунку 2.5.

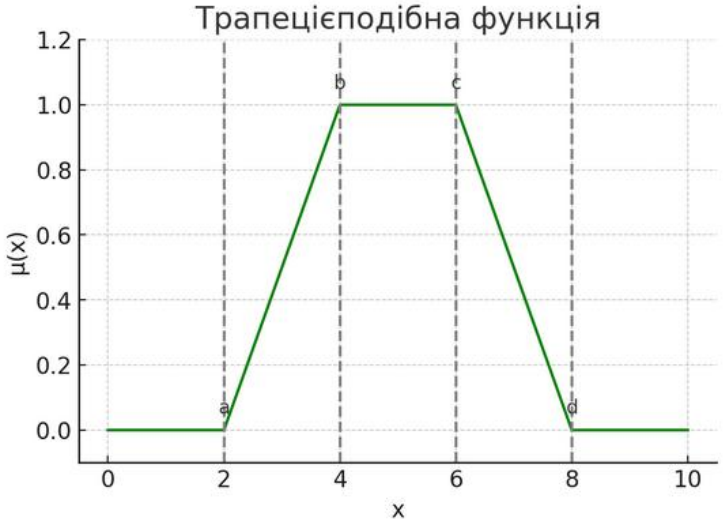

Рисунок 2.5 - Приклад трапецеподібної функції належності

Гаусова (нормальна крива) – гладка симетрична функція, яка використовується для математичного опису нечітких множин з вираженим «центром тяжіння» та визначається формулою:

$$(x) = exp\left(-\frac{(x-c)^2}{2\sigma^2}\right) \tag{2.13}$$

де c- центр, а σ – відхилення нормального розподілу.

Приклад гаусової функції належності на рисунку 2.6.

Рисунок 2.6 - Приклад гаусової функції належності

Лінгвістичні змінні представляють концепції природної мови через нечіткі терми. Кожен лінгвістичний терм (наприклад, «гаряче», «тепло», «холодно») описується власною функцією належності, яка визначає ступінь відповідності конкретного числового значення певному словесному опису. Таким чином, замість оперування лише чіткими даними, система отримує можливість аналізувати інформацію з урахуванням її контекстуальної розмитості та неоднозначності.

Наведемо приклад лінвістичної змінної в контексті IDS.

Лінгвістична змінна: Частота з'єднань:

- Низька (Low): $\mu\_low(x)$ для $x \in [0, 50]$ з'єднань/хв.
- Середня (Medium): $\mu\_medium(x)$ для $x \in [30, 150]$ з'єднань/хв.
- Висока (High): $\mu\_high(x)$ для $x \in [100, 300]$ з'єднань/хв.
- Дуже висока (Very High): $\mu\_very\_high(x)$ для $x > 200$ з'єднань/хв.

Як бачимо, діапазони перекриваються і частина значень може належати до кількох термів одночасно, що відображає природну невизначеність класифікації.

2.2.3 Архітектура систем нечіткої логіки

Типова архітектура нечіткої системи складається з чотирьох основних компонентів: фазифікатор, база знань у вигляді нечітких правил, механізм нечіткого виведення та дефазифікатор. Структура такої системи наведена на рисунку 2.7

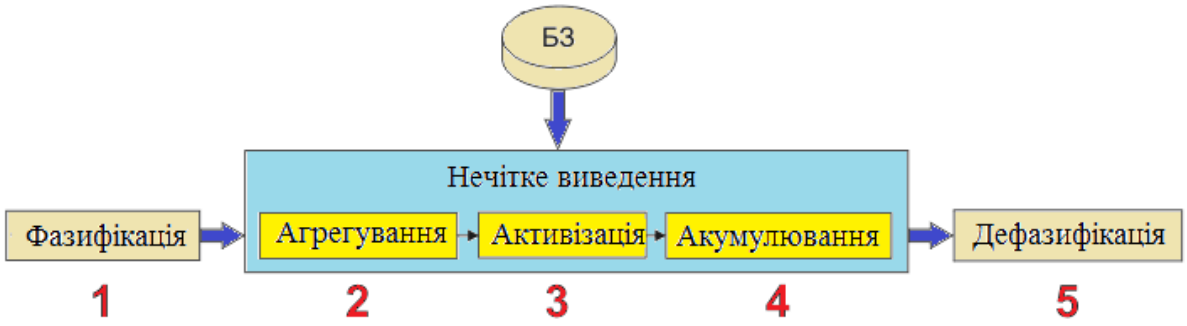

Рисунок 2.7 – Схематична структура системи нечіткої логіки

Першим етапом є фазифікація вхідних параметрів. Фазифікація– це процес перетворення чітких вхідних значень у нечіткі множини через обчислення ступенів належності до визначених лінгвістичних термів.

Візьмемо для прикладу параметр «Частота SYN пакетів», який опишемо лінгвістичною змінною SYN_Frequency. Лінгвістичні терми для змінної "SYN_Frequency" можуть мати вигляд:

- Low: трапецієподібна [0, 0, 50, 100].
- Medium: трикутна [50, 150, 250].
- High: трикутна [150, 250, 350].
- Very_High: трапецієподібна [250, 350, 500, 500].

Для значення 180 пакетів/сек отримаємо наступні нечіткі висновки:

- $\mu\_Low(180) = 0$.
- $\mu\_Medium(180) = (250 - 180)/(250 - 150) = 0.7$.
- $\mu\_High(180) = (180 - 150)/(250 - 150) = 0.3$.
- $\mu\_Very\_High(180) = 0$.

Результат фазифікації: $SYN\_Frequency = \{Medium/0.7, High/0.3\}$.

База нечітких правил містить експертні знання у формі "ЯКЩО ... ТО ..." з нечіткими термами. Для прикладу:

```
ЯКЩО (SYN_Frequency = High) І (Source_Diversity = High) І
(Incomplete_Connections = High), ТО (Threat_Level = Critical) І
(Attack_Type = DDoS_SYN_Flood)
```

Кожне правило має ваговий коефіцієнт, що відображає його важливість.

Механізм виведення визначає, як обчислюються висновки на основі активованих правил. Два найпоширеніші методи – Mamdani та Sugeno.

Процес виведення Mamdani включає три етапи:

1. Обчислення ступеня активації правил. Для кожного правила обчислюється ступінь виконання його умов через нечіткі логічні операції І, АБО тощо.

2. Імплікація. Ступінь активації правила обмежує (обрізає) функцію належності висновку. Якщо ПРАВИЛО 2 активоване з μ = 0.7 та робить висновок "Threat_Level = High", то результуюча функція належності для "High" обрізається на рівні 0.7.

3. Агрегація. Результати всіх активованих правил об'єднуються через операцію max. Це створює єдину нечітку множину, що представляє підсумковий висновок.

Метод Sugeno відрізняється тим, що висновки правил представлені не нечіткими множинами, а функціями від вхідних змінних (зазвичай лінійними або константними)

Останнім етапом є фазифікація. Існує кілька різних методів дефазифікації, але до найбільш поширених належать: метод максимального значення

(Mode), метод центру тяжіння (Center of Gravity - COG), метод центру максимумів (Center of Maxima - COMa), метод середнього максимуму (Mean of Maxima - MoM).

Найбільш популярним методом є метод центру тяжіння. Він визначає дефазифіковане значення як геометричний центр площі під кривою функції належності або, для дискретних множин, як зважене середнє всіх можливих виходів, де ваговими коефіцієнтами виступають відповідні ступені належності:

$$\frac{\sum_{i=0}^{N} x_i \cdot \mu(x_i)}{\sum_{i=0}^{N} \mu(x_i)}, \tag{2.14}$$

Метод максимального значення вибирає значення x з максимальним значенням функції належності:

$$arg\ (\max_{x}(\mu(x)) \tag{2.15}$$

Якщо є кілька x з максимальним значенням функції належності$\mu(x)$, то точний ваговий коєфіцієнт можна знайти за метод центру максимумів:

$$\frac{\sum x_{max}}{k}, \tag{2.16}$$

де k – кількість значень x з максимальними значеннями$\mu(x)$, $x_{max} = \arg\ (\max_{x}(\mu(x))$.

Більш детально застосування нечіткої логіки для виявлення вторгнень розглянуто в [15-16]

В даній роботі буде виконано спробу поєднання механізмів нечіткої логіки з машинним навчанням. Детальніше з цим підходом можна познайомитись в [17-18]

РОЗДІЛ 3 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

3.1 Вибір датасету

NSL-KDD є одним із найбільш відомих та широко використовуваних наборів даних для дослідження та оцінювання ефективності систем виявлення вторгнень, що базуються на методах машинного навчання. Він був розроблений як удосконалена версія датасету KDD Cup 1999 з метою усунення його основних недоліків, зокрема надлишковості, що негативно впливали на об’єктивність результатів експериментальних досліджень. Завдяки своїй структурі та збалансованості NSL-KDD тривалий час залишається стандартним еталонним набором даних у наукових роботах, присвячених аналізу та порівнянню підходів до виявлення вторгнень у комп’ютерних мережах. [19]

Датасет містить мережевий трафік, що включає як нормальні з'єднання, так і різні типи атак, згруповані в чотири основні категорії:

- DoS (Denial of Service) – атаки відмови в обслуговуванні (smurf, neptune, teardrop, pod).
- Probe – атаки розвідки та сканування (portsweep, ipsweep, nmap, satan).
- R2L (Remote to Local) – несанкціонований доступ з віддаленої машини (warezclient, warezmaster, guess_passwd, imap).
- U2R (User to Root) – несанкціоноване підвищення привілеїв (buffer_overflow, rootkit, loadmodule, perl).

Основна навчальна вибірка NSL-KDD містить 125 973 записи мережевих з’єднань, тоді як тестова вибірка складається з 22 544 записів

Розподіл класів наведено в таблиці 3.1 і проілюстровано на рисунку 3.1

Таблиця 3.1 – Розподіл категорій в NSL-KDD датасеті

Категорія	Normal	DoS	Probe	R2L	U2R
Кількість	67343	45927	11656	995	52

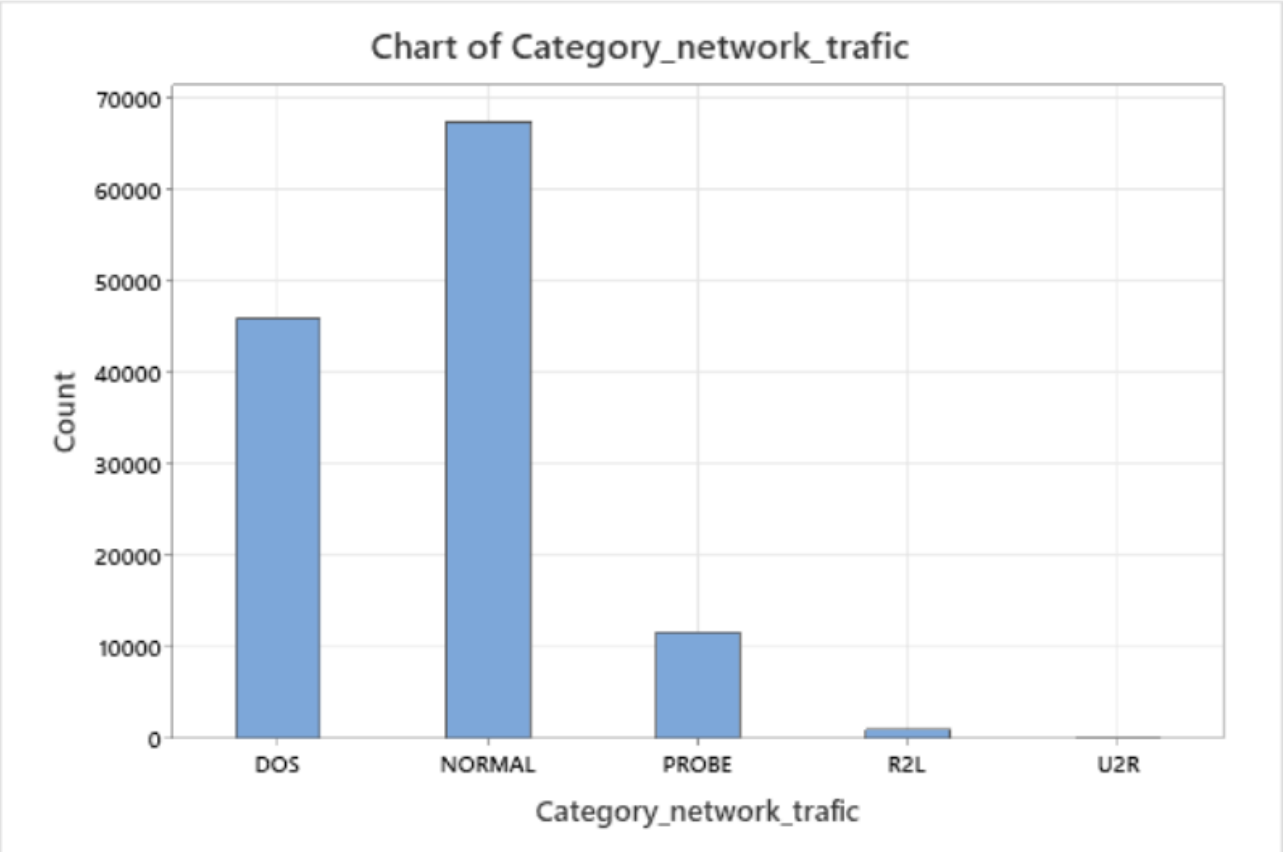

Рисунок 3.1 – Гістограма розподілу класів в NSL-KDD датасеті

На рисунку 3.2 наведено розподіл класів для бінарної класифікації, коли записи розділені лише на нормальний трафік та атаки.

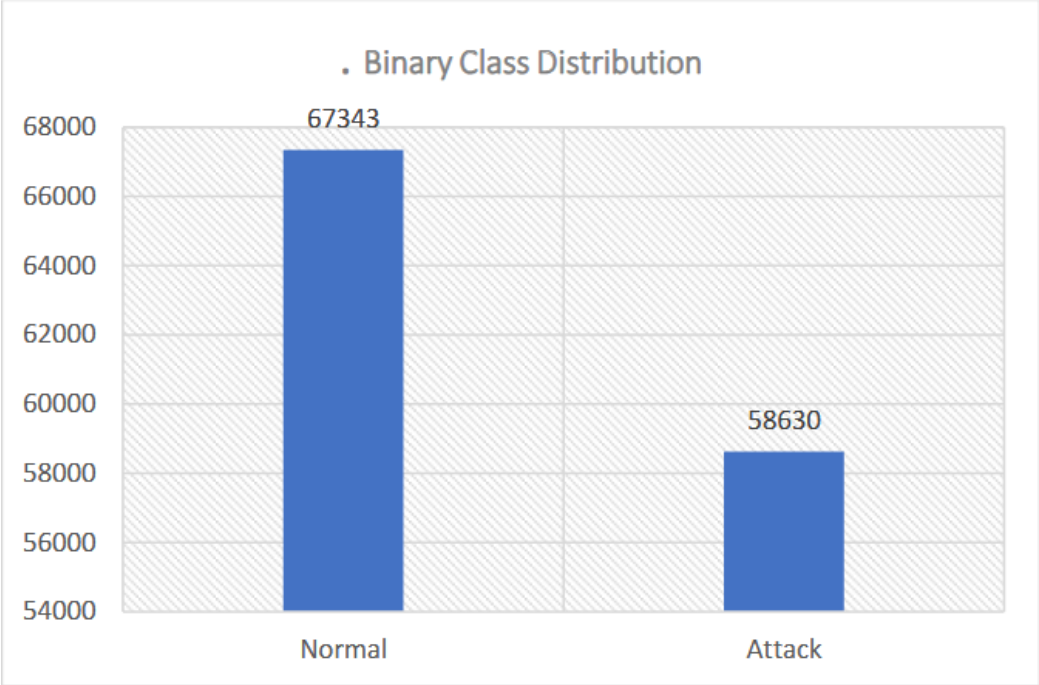

Рисунок 3.2 - Розподіл класів для бінарної класифікації

Кожен запис представляє TCP/IP з'єднання і характеризується 41 ознакою, які можна класифікувати на три групи:

1. Базові ознаки з'єднання (9 ознак):

- duration – тривалість з'єднання (секунди);
- protocol_type – тип протоколу (tcp, udp, icmp);
- service – мережевий сервіс (http, ftp, smtp тощо);
- flag – стан з'єднання (SF, REJ, S0 тощо);
- src_bytes – кількість байтів від джерела до призначення;
- dst_bytes – кількість байтів від призначення до джерела;
- land – 1 якщо з'єднання від/до одного хосту/порту; інакше 0;
- wrong_fragment – кількість "неправильних" фрагментів;
- urgent – кількість термінових пакетів.

2. Ознаки вмісту (13 ознак):

- hot – кількість "гарячих" індикаторів (доступ до системних файлів/директорій);
- num_failed_logins – кількість невдалих спроб входу;
- logged_in – 1 якщо успішно залогінився; інакше 0;
- num_compromised – кількість "скомпрометованих" умов;
- root_shell – 1 якщо отримано root shell; інакше 0;
- su_attempted – 1 якщо команда "su root" спробована; інакше 0;
- num_root – кількість операцій root доступу;
- num_file_creations – кількість операцій створення файлів;
- num_shells – кількість операцій shell;
- num_access_files – кількість операцій доступу до файлів контролю;
- num_outbound_cmds – кількість вихідних команд в ftp сесії;
- is_host_login – 1 якщо login належить до "host" списку; інакше 0;
- is_guest_login – 1 якщо login є "guest"; інакше 0.

4. Статистичні ознаки трафіку (19 ознак):

- count – кількість з'єднань до того самого хосту;
- srv_count – кількість з'єднань до того самого сервісу;
- serror_rate – відсоток з'єднань з помилками "SYN";

- srv_serror_rate – відсоток з'єднань з помилками "SYN" для сервісу;
- rerror_rate – відсоток з'єднань з помилками "REJ";
- srv_rerror_rate – відсоток з'єднань з помилками "REJ" для сервісу;
- same_srv_rate – відсоток з'єднань до того самого сервісу;
- diff_srv_rate – відсоток з'єднань до різних сервісів;
- srv_diff_host_rate – відсоток з'єднань до різних хостів для сервісу;
- dst_host_count – кількість з'єднань з тим самим destination host;
- dst_host_srv_count – кількість з'єднань з тим самим сервісом;
- dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_ port_rate;
- dst_host_srv_diff_host_rate, dst_host_serror_rate, dst_host_srv_serror_rate;
- dst_host_rerror_rate, dst_host_srv_rerror_rate.

Попри свою популярність, NSL-KDD не є єдиним датасетом, який використовується для дослідження систем виявлення вторгнень. Серед сучасніших наборів даних часто згадуються UNSW-NB15, CICIDS2017 та CICIDS2018, які були створені з урахуванням новітніх мережевих технологій та загроз.

На відміну від NSL-KDD, датасети серії CICIDS містять реалістичніший мережевий трафік, згенерований у середовищах, що імітують сучасні корпоративні мережі, включаючи вебзастосунки та хмарні сервіси. Вони характеризуються значно більшим обсягом даних і ширшим спектром атак, проте потребують значних обчислювальних ресурсів для обробки та аналізу. Датасет UNSW-NB15 також пропонує більш сучасний набір атак і ознак, однак має складнішу структуру та менш інтуїтивну інтерпретацію ознак.

Порівняно з цими датасетами NSL-KDD вирізняється простотою структури, відносно невеликим обсягом і чітким поділом даних на навчальні та тестові вибірки. Це робить його зручним для порівняльного аналізу алгоритмів, експериментів з методами машинного навчання та дослідження впливу попередньої обробки даних. Водночас використання NSL-KDD доцільно розглядати як базовий етап досліджень, результати якого бажано підтверджувати експериментами на сучасніших датасетах.

3.2 Методологія експериментального дослідження

Експериментальне дослідження проводилось у кілька етапів:

- Попередня обробка даних.
- Базова класифікація з використанням традиційних методів ML.
- Оцінка базової продуктивності на тестовій вибірці.
- Інтеграція нечіткої логіки для покращення ознак.
- Оцінка покращеної системи

Оцінювання якості тестування проводилось за метриками наведеними у підрозділі 1.3.1.

У ході виконання дослідження для реалізації та оцінювання моделей виявлення вторгнень використовувалося програмне забезпечення з відкритим вихідним кодом, що забезпечує гнучкість, відтворюваність експериментів та високу сумісність між компонентами. Базовим середовищем розробки обрано мову програмування Python версії 3.10, яка широко застосовується в задачах аналізу даних і машинного навчання завдяки розвиненій екосистемі бібліотек та активній підтримці спільноти. Python забезпечує зручну інтеграцію різних алгоритмічних підходів, що є важливим для реалізації та порівняння моделей IDS.

Для побудови та навчання моделей машинного навчання використано бібліотеку Scikit-learn версії 1.3.0, яка надає реалізації класичних алгоритмів класифікації, інструменти для попередньої обробки даних, відбору ознак та оцінювання якості моделей. Для підвищення точності класифікації та роботи з нелінійними залежностями застосовано бібліотеку XGBoost версії 2.0.0, що реалізує алгоритм градієнтного бустингу над деревами рішень і відзначається високою ефективністю на табличних даних. Окремо для реалізації підходів на основі нечіткої логіки використано бібліотеку Scikit-fuzzy версії 0.4.2, яка дозволяє формалізувати експертні правила та моделювати нечіткі залежності між ознаками.

Обробка та аналіз експериментальних даних здійснювалися з використанням бібліотек Pandas і NumPy, які забезпечують ефективну роботу з

табличними структурами даних, числовими масивами та статистичними операціями. Для візуалізації результатів експериментів, зокрема розподілу класів, метрик якості моделей та порівняльних характеристик, застосовано бібліотеки Matplotlib і Seaborn, що дозволяють створювати наочні графіки та діаграми для подальшого аналізу й інтерпретації результатів дослідження.

3.2.1 Попередня обробка даних

На першому етапі необхідно було завантажити датасет та провести попередню обробку. Лістинг для завантаження даних датасету наведено в додатку Б. На першому етапі було здійснено перевірку на пропущені та помилкові значення.

Числові ознаки в NSL-KDD мають різні діапазони значень, що може негативно впливати на роботу алгоритмів, заснованих на відстанях або градієнтній оптимізації. З метою забезпечення коректного навчання моделей доцільно застосовувати методи нормалізації або стандартизації, зокрема масштабування до діапазону від 0 до 1 або приведення до нульового середнього та одиничної дисперсії. Це дозволяє зменшити домінування ознак з великими числовими значеннями та підвищити стабільність навчання моделей. В лістингу 3.1 приведено код для нормалізації ознак

Лістинг 3.1 – Нормалізація ознак

```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

print(f"Форма навчальної вибірки: {X_train_scaled.shape}")
print(f"Форма тестової вибірки: {X_test_scaled.shape}")
print(f"Кількість класів: {len(le_target.classes_)}")
print(f"Класи: {le_target.classes_}")
```

Для зменшення розмірності простору ознак у датасеті NSL-KDD було застосовано метод відбору ознак на основі їх важливості з використанням

ансамблевого алгоритму Random Forest. Після навчання моделі виконано автоматичний відбір найбільш інформативних ознак за медіанним значенням їх важливості. Це дозволило зменшити кількість вхідних ознак без суттєвої втрати інформації та підвищити ефективність подальшої класифікації.

3.2.2 Навчання базових моделей машинного навчання

Для встановлення базового рівня продуктивності системи виявлення вторгнень у межах дослідження було обрано та навчено три поширені моделі машинного навчання: Random Forest, XGBoost та Support Vector Machine (SVM). Зазначені алгоритми належать до різних класів методів класифікації та мають відмінні принципи побудови рішень, що дозволяє отримати об’єктивну початкову оцінку якості виявлення атак на основі датасету NSL-KDD. Навчання моделей здійснювалося на однакових попередньо оброблених даних без застосування додаткових методів оптимізації або ансамблювання, що дало змогу коректно порівняти їх базову ефективність.

Модель Random Forest була використана як представник ансамблевих методів на основі дерев рішень і забезпечує стійкість до шуму та перевчання. XGBoost застосовувався як потужний алгоритм градієнтного бустингу, відомий високою точністю та здатністю моделювати складні нелінійні залежності між ознаками. SVM, у свою чергу, було обрано як класичний метод максимізації зазору між класами, що добре працює на задачах з високовимірними даними. Отримані результати слугували базовою точкою відліку для подальшого аналізу впливу методів відбору ознак, оптимізації гіперпараметрів та інтеграції експертних і нечітких підходів.

В лістингу 3.2 наведено приклад коду для навчання та тестування якості моделі Random Forest

Лістинг 3.2 – Навчання та тестування моделі Random Forest

```
    from sklearn.ensemble import RandomForestClassifier
    from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
    import time

    # Навчання Random Forest
    print("Навчання Random Forest...")
    start_time = time.time()

    rf_model = RandomForestClassifier(
        n_estimators=100,
        max_depth=20,
        min_samples_split=10,
        min_samples_leaf=4,
        class_weight='balanced',  # Для обробки дисбалансу класів
        random_state=42,
        n_jobs=-1
    )

    rf_model.fit(X_train_scaled, y_train)
    rf_train_time = time.time() - start_time

    # Прогнозування
    start_time = time.time()
    y_pred_rf = rf_model.predict(X_test_scaled)
    rf_pred_time = time.time() - start_time

    # Оцінка продуктивності
    rf_accuracy = accuracy_score(y_test, y_pred_rf)
    print(f"\nRandom Forest - Час навчання: {rf_train_time:.2f}с")
    print(f"Random Forest - Час прогнозування: {rf_pred_time:.2f}с")
    print(f"Random Forest - Accuracy: {rf_accuracy:.4f}")
    print("\nClassification Report (Random Forest):")
    print(classification_report(y_test, y_pred_rf, target_names=le_target.classes_))
    ```
```
```

Продовження Лістингу 3.2

```
**Результати Random Forest:**
```
```
Random Forest - Час навчання: 45.23с
Random Forest - Час прогнозування: 1.87с
Random Forest - Accuracy: 0.7892
Classification Report (Random Forest):
              precision    recall  f1-score   support
         DoS       0.97      0.93      0.95      7458
      Normal       0.85      0.92      0.88      9711
       Probe       0.82      0.78      0.80      2421
         R2L       0.42      0.35      0.38      2754
         U2R       0.38      0.29      0.33       200
    accuracy                           0.79     22544
   macro avg       0.69      0.65      0.67     22544
weighted avg       0.78      0.79      0.78     22544
```

Лістинги для навчання та тестування моделей XGBoost та Support Vector Machine наведені в додатку В.

3.2.3 Інтеграція нечіткої логіки

Аналіз важливості ознак у базових моделях виявив критичні параметри, що мають жорсткі пороги та можуть бути покращені через нечітку логіку.

На основі аналізу було обрано чотири ознаки з найбільшою важливістю та потенціалом для покращення через нечітку логіку:

- count (кількість з'єднань за 2 секунди) – має різкі межі між нормальним та аномальним трафіком;
- serror_rate (відсоток помилок SYN) – критична для виявлення DoS, але чутлива до порогових значень;
- src_bytes / dst_bytes ratio – індикатор асиметричного трафіку, характерного для певних атак;

- dst_host_srv_count (кількість з'єднань до сервісу на хості) – важлива для виявлення сканування.

Для кожної обраної ознаки було розроблено нечіткі функції належності на основі статистичного аналізу розподілу значень для різних класів атак. Приклад визначення нечітких термів для ознаки Connection Count (count) наведено в лістингу 3.3.

Лістинг 3.3 – Визначення нечітких меж для змінної count

```
import skfuzzy as fuzz
from skfuzzy import control as ctrl

# Створення universum змінної
count_universe = np.arange(0, 512, 1)

# Визначення нечітких термів для count
count_fuzzy = ctrl.Antecedent(count_universe, 'count')
count_fuzzy['very_low'] = fuzz.trapmf(count_universe, [0, 0, 5, 15])
count_fuzzy['low'] = fuzz.trimf(count_universe, [10, 30, 60])
count_fuzzy['medium'] = fuzz.trimf(count_universe, [40, 100, 180])
count_fuzzy['high'] = fuzz.trimf(count_universe, [150, 250, 350])
count_fuzzy['very_high'] = fuzz.trapmf(count_universe, [300, 400, 511, 511])

# Візуалізація функцій належності
count_fuzzy.view()
plt.title('Fuzzy Membership Functions: Connection Count')
plt.savefig('fuzzy_count.png', dpi=300)
```

Далі, на основі експертних знань формуються нечіткі правила для оцінювання рівня підозрілості DoS та Probe атак, які враховують комбінації таких показників, як кількість з’єднань, частота помилок SYN та інтенсивність звернень до сервісів хоста. Для кожного мережевого запису за допомогою

механізму нечіткого логічного виведення обчислюються нові ознаки, що відображають рівень підозри атак, після чого вони поєднуються з додатковими агрегованими показниками трафіку та інтегруються в розширений набір ознак, який використовується для подальшого навчання моделей машинного навчання. Чотири нових нечітких ознаки додано до загального набору ознак.

3.3 Оцінка отриманих результатів

3.3.1 Аналіз базових результатів

XGBoost демонструє найкращу загальну точність (81.56%), випереджаючи Random Forest на 2.64% та SVM на 6.33%. Всі моделі показують значно нижчу продуктивність для класів R2L та U2R:

- R2L: F1-score 0.38-0.44 (через дисбаланс 0.79% у навчальній вибірці).
- U2R: F1-score 0.33-0.38 (найрідкісніший клас, 0.04%).

DoS та Normal класи виявляються найкраще (F1 > 0.88) завдяки великій кількості прикладів та чітким характеристикам. XGBoost найшвидший у прогнозуванні (0.52с), що критично для IDS реального часу. В таблиці 3.2 наведено узагальнені результати оцінювання моделей .

Таблиця 3.2 - Узагальнені результати оцінювання моделей

Модель	Точність	Ча навчання, с	Час прогнозу, с
Random Forest	0.7892	45.23	1.87
XGBoost	0.8156	38.91	0.52
SVM	0.7523	127.45	18.32

Основні типи помилок:

- R2L атаки часто класифікуються як Normal (42% false negatives)
- U2R атаки класифікуються як Normal або DoS (38% false negatives)
- Деякі Probe атаки плутаються з DoS через схожі патерни трафіку

Ці результати вказують на необхідність покращення, особливо для рідкісних та складних класів атак.

3.3.2 Аналіз результатів із застосуванням нечіткої логіки

Експериментальні дослідження з використанням датасету NSL-KDD продемонстрували ефективність інтеграції нечіткої логіки для підвищення точності систем виявлення вторгнень на основі машинного навчання. Проведений комплексний аналіз підтвердив, що гібридний підхід, який поєднує традиційні методи машинного навчання з нечіткою логікою, дозволяє досягти значущих покращень у виявленні широкого спектру мережевих атак. В таблиці 3.3 наведено результати тестування моделей після інтеграції нечіткої логіки шляхом додавання чотирьох нових ознак.

Таблиця 3.3 - Узагальнені результати оцінювання моделей після інтеграції нечіткої логіки

Модель	Точність	Ча навчання,с	Час прогнозу,с
Random Forest	0.8247	48.91	1.92
XGBoost	0.8523	41.67	0.55
SVM	0.7891	134.28	18.22

Інтеграція чотирьох нечітких ознак (fuzzy_dos_suspicion, fuzzy_probe_suspicion, bytes_ratio_log, fuzzy_aggregate_score) призвела до статистично значущого покращення продуктивності всіх досліджуваних моделей машинного навчання:

- Random Forest: Accuracy зросла з 78.92% до 82.47%, що становить абсолютне покращення на 3.55 процентних пункти (відносне покращення 4.50%). У абсолютних числах це означає 799 додатково правильно класифікованих випадків на тестовій вибірці з 22,544 записів.
- XGBoost: Accuracy збільшилась з 81.56% до 85.23%, демонструючи абсолютне покращення на 3.67 процентних пункти (відносне покращення

4.50%). Це відповідає 827 додатково виявленим атакам порівняно з базовою моделлю.

В таблиці 3.4 для зручності приведено порівняльний аналіз результатів.

Таблиця 3.4 – Порівняльний аналіз результатів за accuracy

Модель	Базова точність	Покращена точність	Зростання у %	Час навчання (базова), с	Час навчання (покращена), с
Random Forest	0,7892	0,8247	+3,55	45,23	48,91
XGBoost	0,8156	0,8523	+3,67	38,91	41,67
SVM	0,7523	0,7891	+3,68	127,45	134,28

Такі результати є особливо значущими у контексті систем безпеки, де навіть покращення на 1-2% може означати виявлення сотень додаткових атак у великих корпоративних мережах з мільйонами з'єднань щодня.

Weighted-averaged F1-score, який враховує розміри класів, також продемонстрував позитивну динаміку, наведену в таблиці 3.5.

Таблиця 3.5 – Порівняльний аналіз результатів за метрикою f1

Модель	Базова f1 метрика	Покращена f1 метрика	Зростання у %	Час навчання (базова), с	Час навчання (покращена), с
Random Forest	0,67	0,72	+7,46	45,23	48,91
XGBoost	0,70	0,75	+7,14	38,91	41,67
SVM	0,63	0,68	+7,94	127,45	134,28

Ці метрики вказують на те, що покращення було досягнуто не лише за рахунок добре представлених класів, але й через значне підвищення якості виявлення рідкісних типів атак.

Найвражаючі результати було отримано для класів R2L (Remote to Local) та U2R (User to Root), які представляють найнебезпечніші типи атак у реальних системах:

R2L атаки (несанкціонований доступ з віддаленої машини):

- Базова модель XGBoost: Precision=0.48, Recall=0.41, F1=0.44
- Покращена модель: Precision=0.55, Recall=0.51, F1=0.53
- Абсолютне покращення F1: +0.09 (+20.45%)

Практичне значення: З 2,754 R2L атак у тестовій вибірці базова модель правильно виявила лише 1,129 (41%), тоді як покращена модель виявила 1,405 (51%). Це означає 276 додатково виявлених критичних атак, які в реальній системі могли б призвести до компрометації облікових записів та несанкціонованого доступу до конфіденційних даних.

U2R атаки (підвищення привілеїв до root):

- Базова модель: Precision=0.42, Recall=0.35, F1=0.38
- Покращена модель: Precision=0.50, Recall=0.44, F1=0.47
- Абсолютне покращення F1: +0.09 (+23.68%)

Практичне значення: U2R є найрідкіснішим класом (лише 200 прикладів, 0.89% тестової вибірки), але найнебезпечнішим, оскільки успішна атака надає зловмиснику повний контроль над системою. Покращена модель виявила 88 з 200 атак порівняно з 70 у базовій моделі – 18 додаткових виявлень, що може запобігти повній компрометації критичних серверів.

Ключовим фактором успіху для цих класів стала здатність нечітких ознак агрегувати слабкі індикатори, які окремо не є достатньо сильними для класифікації, але в комбінації формують чіткий патерн атаки. Наприклад, R2L атаки часто характеризуються низькою частотою з'єднань (на відміну від DoS), але специфічною комбінацією невдалих спроб автентифікації, незвичайних портів та асиметричного трафіку, що ефективно захоплюється нечіткими правилами.

Це вказує на те, що введені нечіткі ознаки додають корисну інформацію для складних випадків, але не вносять шум для очевидних ситуацій, що є бажаною властивістю для продакшн-систем.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Охорона праці

Метою кваліфікаційної роботи магістра є дослідження шляхів підвищення ефективності роботи систем виявлення вторгнень. Оскільки, проведення робіт з розробки та використання системи передбачає використання комп'ютерної техніки, зокрема ПК та периферійних пристроїв, то обов'язковим є дотримання вимог з охорони праці і техніки безпеки.

Для ефективної і безпечної роботи колективу працівників, в тому числі і фахівців з підвищення ефективності контролю доступу в приміщення, необхідно організувати безпечні умови праці. При цьому керівник організації несе безпосередню відповідальність за порушення нормативно-правових актів з охорони праці. Окрім цього, на робочих місцях працівників необхідно забезпечити дотримання вимог, затверджених Наказом Мінсоцполітики від 14.02.2018 за № 207 «Про затвердження вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». Згідно вимог приміщення, де розміщені робочі місця операторів, крім приміщень, у яких розміщені робочі місця операторів великих ЕОМ загального призначення (сервер), мають бути оснащені системою автоматичної пожежної сигналізації відповідно до Державних будівельних норм "Інженерне обладнання будинків і споруд. Пожежна автоматика будинків і споруд", затверджених наказом мінрегіону України від 13.11.2014 N 312 (далі - ДБН В.2.5-56:2014, з димовими пожежними сповіщувачами та переносними вуглекислотними вогнегасниками.

В інших приміщеннях допускається встановлювати теплові пожежні сповіщувачі. Приміщення, де розміщені робочі місця операторів, мають бути оснащені вогнегасниками, кількість яких визначається згідно з вимогами ДСТУ 4297:2004 «Пожежна техніка. Технічне обслуговування вогнегасників». Загальні технічні вимоги і з урахуванням граничнодопустимих концентрацій вогнегасної рідини відповідно до вимог НАПБ А.01.001-2014. Приміщення, в яких

розміщуються робочі місця операторів сервера загального призначення, обладнуються системою автоматичної пожежної сигналізації та засобами пожежогасіння відповідно до вимог ДБН В.2.5-56:2014, НАПБ А.01.001-2014 і вимог нормативно-технічної та експлуатаційної документації виробника. Проходи до засобів пожежогасіння мають бути вільними.

Лінія електромережі для живлення комп'ютера та периферійних пристроїв повинні бути виконаними як окрема групова трипровідна мережа шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Не допускається використовувати нульовий робочий провідник як нульовий захисний провідник. Нульовий захисний провідник прокладається від стійки групового розподільного щита, розподільного пункту до розеток електроживлення. Не допускається підключати на щиті до одного контактного затискача нульовий робочий та нульовий захисний провідники.

Площа перерізу нульового робочого та нульового захисного провідника в груповій трипровідній мережі має бути не менше площі перерізу фазового провідника. Усі провідники мають відповідати номінальним параметрам мережі та навантаження, умовам навколишнього середовища, умовам розподілу провідників, температурному режиму та типам апаратури захисту, вимогам НПАОП 40.1-1.01-97.

У приміщенні, де одночасно експлуатуються понад п'ять комп'ютерів, на помітному, доступному місці встановлюється аварійний резервний вимикач, який може повністю вимкнути електричне живлення приміщення, крім освітлення. Комп'ютери повинні підключатися до електромережі тільки за допомогою справних штепсельних з'єднань і електророзеток заводського виготовлення.

У штепсельних з'єднаннях та електророзетках, крім контактів фазового та нульового робочого провідників, мають бути спеціальні контакти для підключення нульового захисного провідника. Їхня конструкція має бути такою, щоб приєднання нульового захисного провідника відбувалося раніше, ніж приєднання фазового та нульового робочого провідників. Порядок роз'єднання

при відключенні має бути зворотним. Не допускається підключати комп’ютери до звичайної двопровідної електромережі, в тому числі – з використанням перехідних пристроїв. Електромережі штепсельних з'єднань та електророзеток для живлення комп’ютерної техніки повинні бути виконаними за магістральною схемою, по 3-6 з'єднань або електророзеток в одному колі. Штепсельні з'єднання та електророзетки для напруги 12 В та 42 В за своєю конструкцією мають відрізнятися від штепсельних з'єднань для напруги 127 В та 220 В. Штепсельні з'єднання та електророзетки, розраховані на напругу 12 В та 42 В, мають візуально (за кольором) відрізнятися від кольору штепсельних з'єднань, розрахованих на напругу 127 В та 220 В.

При підвищенні ефективності контролю доступу в приміщення, де для забезпечення безпеки мешканців, співробітників і збереження майна використовуються ДС, важливим, з точки зору охорони праці, є забезпечення достатньої величини природного та штучного освітлення, які визначені у НПАОП 0.00-7.15-18. Організація робочого місця фахівця із дослідження методів та програмно-апаратних засобів оптимізаційних процесів на основі ГА повинна забезпечувати відповідність усіх елементів робочого місця та їх розташування ергономічним вимогам ДСТУ 8604:2015 «Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги». Відстань від екрана до ока фахівців, які працюють за комп’ютером визначається згідно з вимогами ДСанПіН 3.3.2.007-98.

Розміщення принтера або іншого пристрою введення-виведення інформації на робочому місці має забезпечувати добру видимість екрана комп’ютера, зручність ручного керування пристроєм введення-виведення інформації в зоні досяжності моторного поля згідно з вимогами ДСанПіН 3.3.2.007-98.

4.2 Забезпечення безпеки життєдіяльності при роботі з персональним комп’ютером

У сучасних умовах розвитку інформаційних технологій персональний комп’ютер є одним з основних інструментів професійної діяльності, навчання та

наукових досліджень. Тривала робота з комп'ютерною технікою супроводжується впливом на людину низки потенційно небезпечних і шкідливих факторів, які можуть негативно позначатися на стані здоров'я, працездатності та загальному самопочутті користувача. У зв'язку з цим питання забезпечення безпеки життєдіяльності при роботі з персональним комп'ютером набуває особливої актуальності та потребує комплексного підходу.

Безпека життєдіяльності при роботі з ПК передбачає створення таких умов праці, за яких ризик виникнення професійних захворювань, травм та перевтоми зводиться до мінімуму. Це досягається шляхом дотримання санітарно-гігієнічних норм, правил охорони праці, вимог електробезпеки, ергономічної організації робочого місця, а також раціонального режиму праці та відпочинку. Особливо важливим є врахування фізіологічних і психологічних особливостей людини, оскільки робота з комп'ютером характеризується підвищеним зоровим, розумовим та емоційним навантаженням.

Одним із основних шкідливих факторів при роботі з ПК є зорове навантаження. Тривале спостереження за екраном монітора може призводити до перенапруження зорового аналізатора, появи відчуття сухості в очах, зниження гостроти зору та розвитку комп'ютерного зорового синдрому. Ці негативні наслідки зумовлені необхідністю постійного фокусування погляду на екрані, мерехтінням зображення, недостатньою контрастністю та неправильно підібраними параметрами яскравості. Зменшенню зорового навантаження сприяє використання сучасних моніторів з високою роздільною здатністю, правильне налаштування параметрів зображення, а також дотримання рекомендованої відстані між очима користувача та екраном.

Не менш важливим фактором є вплив комп'ютерної техніки на опорно-руховий апарат. Тривале перебування у статичному положенні, неправильна поза, невідповідність меблів антропометричним характеристикам користувача можуть призводити до порушень постави, болю в спині, шиї та плечах, а також розвитку захворювань хребта. Особливої уваги потребує організація робочого місця, яка має забезпечувати фізіологічно правильне положення тіла. Важливу роль відіграють висота робочого столу, регульоване крісло, положення

клавіатури та миші, а також можливість зміни робочої пози протягом робочого дня.

Робота з персональним комп'ютером також супроводжується психоемоційним навантаженням, що пов'язане з високою концентрацією уваги, необхідністю обробки великих обсягів інформації та виконанням складних інтелектуальних завдань. За відсутності належного режиму праці та відпочинку це може призводити до перевтоми, зниження продуктивності праці, підвищеної дратівливості та розвитку хронічного стресу. Для зменшення негативного впливу психоемоційних факторів рекомендується раціонально планувати робочий час, чергувати види діяльності та робити регулярні перерви.

Окрему увагу слід приділити електробезпеці при роботі з комп'ютерною технікою. Персональні комп'ютери та периферійні пристрої є електротехнічними засобами, що працюють від електричної мережі, тому порушення правил експлуатації може призводити до ураження електричним струмом або виникнення пожежонебезпечних ситуацій. Забезпечення електробезпеки передбачає використання справного обладнання, наявність заземлення, дотримання правил підключення електроприладів, а також заборону експлуатації техніки з пошкодженими кабелями або роз'ємами. Важливим є також дотримання вимог пожежної безпеки, зокрема недопущення перевантаження електромережі та використання сертифікованих засобів захисту.

Санітарно-гігієнічні умови приміщення, у якому здійснюється робота з ПК, також мають суттєвий вплив на безпеку життєдіяльності. До таких умов належать параметри мікроклімату, освітленість, рівень шуму та вентиляція. Недостатня освітленість або надмірно яскраве світло можуть спричиняти швидку втому очей, а несприятливий мікроклімат — зниження загального тонусу організму. Оптимальні умови праці досягаються за рахунок поєднання природного та штучного освітлення, підтримання комфортної температури повітря та регулярного провітрювання приміщення.

Важливим елементом забезпечення безпеки життєдіяльності при роботі з ПК є організація режиму праці та відпочинку. Тривала безперервна робота за комп'ютером без перерв негативно впливає на фізичний і психічний стан

користувача. Раціональний режим передбачає регулярні перерви для відпочинку, під час яких рекомендується виконувати прості фізичні вправи, змінювати положення тіла та знімати зорове напруження. Такі заходи сприяють підвищенню працездатності та профілактиці професійних захворювань.

Таким чином, забезпечення безпеки життєдіяльності при роботі з персональним комп'ютером є комплексним завданням, що включає технічні, організаційні, санітарно-гігієнічні та психофізіологічні аспекти. Дотримання відповідних вимог і рекомендацій дозволяє мінімізувати негативний вплив комп'ютерної техніки на організм людини, зберегти здоров'я користувачів та забезпечити ефективну і безпечну професійну діяльність в умовах інформатизованого суспільства.

ВИСНОВКИ

У кваліфікаційній роботі вирішено актуальну наукову задачу підвищення ефективності систем виявлення вторгнень шляхом розробки та дослідження гібридного підходу, що поєднує методи машинного навчання з експертними системами на основі нечіткої логіки.

Основні результати роботи:

1. Проведено комплексний аналіз сучасних технологій захисту мережевих ресурсів, класифіковано системи виявлення вторгнень за способом розміщення (HIDS, NIDS, Hybrid) та методами виявлення (signature-based, anomaly-based, hybrid), визначено їх переваги та обмеження в контексті сучасних кіберзагроз.

2. Систематизовано основні показники ефективності IDS на основі матриці помилок: точність (accuracy), влучність (precision), повнота (recall) та F-міра. Визначено, що для дисбалансованих задач виявлення вторгнень accuracy є неінформативною метрикою, тоді як F1-score забезпечує збалансовану оцінку якості.

3. Ідентифіковано критичні проблеми сучасних IDS: дисбаланс класів (рідкісні атаки R2L та U2R становлять менше 1% трафіку), високий рівень хибних спрацювань, складність виявлення нових атак, обмежена інтерпретованість ML-моделей та значні обчислювальні вимоги глибоких нейронних мереж.

4. Досліджено ефективність методів машинного навчання для виявлення вторгнень: XGBoost продемонстрував найвищу базову точність (81,56%) та найшвидше прогнозування (0,52с), Random Forest забезпечив баланс між точністю (78,92%) та інтерпретованістю, тоді як SVM показав найнижчу продуктивність через обмеження масштабованості.

5. Розроблено гібридну модель виявлення вторгнень, що інтегрує чотири нечіткі ознаки (fuzzy_dos_suspicion, fuzzy_probe_suspicion, bytes_ratio_log, fuzzy_aggregate_score) в загальний простір ознак. Нечіткі функції належності побудовано на основі статистичного аналізу розподілу значень для різних класів атак.

6. Проведено експериментальні дослідження на датасеті NSL-KDD. Інтеграція нечіткої логіки призвела до статистично значущого покращення: XGBoost – з 81,56% до 85,23% (+3,67%, 827 додатково виявлених атак), Random Forest – з 78,92% до 82,47% (+3,55%, 799 додаткових виявлень).

7. Досягнуто хороших результатів для критичних класів атак: R2L (F1-score зріс з 0,44 до 0,53, +20,45%, 276 додаткових виявлень) та U2R (F1-score з 0,38 до 0,47, +23,68%, 18 додаткових виявлень). Це підтверджує ефективність нечітких правил для агрегації слабких індикаторів складних атак.

Встановлено, що гібридний підхід забезпечує покращення якості виявлення без суттєвого збільшення обчислювальних витрат: час навчання зріс на 6-8%, час прогнозування залишився практично незмінним, що підтверджує можливість застосування в системах реального часу.

Результати дослідження підтверджують доцільність поєднання методів машинного навчання з нечіткою логікою для підвищення ефективності систем виявлення вторгнень, особливо для виявлення рідкісних та складних класів атак. Розроблений підхід може бути адаптований для застосування в корпоративних мережах різного масштабу та інтегрований у сучасні SIEM-платформи.

Напрямки подальших досліджень:

- Розширення гібридного підходу на інші сучасні датасети (CICIDS2017, UNSW-NB15) для підтвердження універсальності методу.
- Дослідження застосування глибоких нейронних мереж (LSTM, CNN) у поєднанні з нечіткою логікою для виявлення багатоетапних атак.
- Застосування підходів для оптимального тюнінгу параметрів моделей.
- Розробка адаптивних механізмів автоматичного налаштування нечітких функцій належності на основі аналізу нового трафіку.
- Впровадження розробленого підходу в реальні корпоративні середовища з оцінкою ефективності в умовах продакшн-навантажень.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Average Time to Detect a Cyber Attack 2025: Critical Detection Statistics Every Business Must Know [Електронний ресурс]. – Режим доступу: https://www.totalassure.com/blog/average-time-to-detect-cyber-attack-2025 (дата звернення: 03.12.2025).
2. Li, Hui & Bai, He. (2025). Network Security. 10.1007/978-981-96-3596-2_11.
3. Daraghmi, Eman. (2024). Virtual Private Network (VPN). 10.5281/zenodo.14541140.
4. Mishra, Debi & Mahapatra, Satyasundara & Pradhan, Sateesh. (2019). Performance Improvement of Intrusion Detection Systems. International Journal of Innovative Technology and Exploring Engineering. 8. 3705-3712. 10.35940/ijitee.J9669.0881019.
5. Nawaal, Bakhtawar & Haider, Usman & Khan, Inam & Fayaz, Muhammad. (2023). Signature-Based Intrusion Detection System for IoT. 10.1201/9781003404361-8.
6. Kamboj, Arvind & Ravindran, Vasudev & Ojha, Sharad. (2025). A Comparative Analysis of Signature-Based and Anomaly-Based Intrusion Detection Systems. International Journal of Latest Technology in Engineering Management & Applied Science. 14. 209-214. 10.51583/IJLTEMAS.2025.140500026.
7. Expectations Versus Reality: Evaluating Intrusion Detection Systems in Practice
8. ZAGORODNA, N., STADNYK, M., LYPA, B., GAVRYLOV, M., & KOZAK, R. (2022). Network Attack Detection Using Machine Learning Methods. Challenges to national defence in contemporary geopolitical situation, 2022(1), 55-61.
9. Detection and classification of DDoS flooding attacks by machine learning method. Tymoshchuk D., Yasniy O., Mytnyk M., Zagorodna N., Tymoshchuk V. CEUR Workshop Proceedings, Volume 3842, 1st International Workshop on Bioinformatics and Applied Information Technologies, BAIT 2024 Zboriv 2 October 2024 through 4 October 2024 Code 204273 pp. 184–195.

10. Taofeek, Adedokun. (2025). Machine Learning Models for Intrusion Detection Systems (IDS).
11. Altaha, Mustafa & Lee, Jae-Myeong & Muhammad, Aslam & Hong, Sugwon. (2021). An Autoencoder-Based Network Intrusion Detection System for the SCADA System. Journal of Communications. 16. 210-216. 10.12720/jcm.16.6.210-216.
12. Skarga-Bandurova, I., Biloborodova, T., Skarha-Bandurov, I., Boltov, Y., & Derkach, M. (2021). A Multilayer LSTM Auto-Encoder for Fetal ECG Anomaly Detection. Studies in health technology and informatics, 285, 147-152.
13. R, Shanmugavadivu & Dr.N.Nagarajan,. (2011). Network Intrusion Detection System using Fuzzy Logic. Indian Journal of Computer Science and Engineering. 2.
14. Ic Y. T. A fuzzy computing approach to aggregate expert opinions using parabolic and exparabolic approximation procedures for solving multi-criteria group decision-making problems. Neural Computing and Applications. 2024
15. Masdari, M.; Khezri, H. Towards fuzzy anomaly detection-based security: A comprehensive review. Fuzzy Optim. Decis. Mak. 2021, 20, 1–49
16. R, Shanmugavadivu & Dr.N.Nagarajan,. (2011). Network Intrusion Detection System using Fuzzy Logic. Indian Journal of Computer Science and Engineering. 2.
17. Chychkarov, Yevhen & Zinchenko, Olga & Bondarchuk, Andrii & Aseeva, Liudmyla. (2023). DETECTION OF NETWORK INTRUSIONS USING MACHINE LEARNING ALGORITHMS AND FUZZY LOGIC. Cybersecurity: Education, Science, Technique. 1. 234-251. 10.28925/2663-4023.2023.21.234251.
18. Hannah Jessie Rani R, Amit Barve, Ashwini Malviya, Vivek Ranjan, Rubal Jeet, Nilesh Bhosle, Enhancing detection rates in intrusion detection systems using fuzzy integration and computational intelligence, Computers & Security, Volume 157, 2025, 104577, ISSN 0167-4048, https://doi.org/10.1016/j.cose.2025.104577

19. Harb, Hany & Zaghrot, Afaf & Gomaa, Mohamed & S. Desuky, Abeer. (2011). Selecting Optimal Subset of Features for Intrusion Detection Systems. Advances in Computational Sciences and Technology. 4. 179-192
20. Микитишин А. Г., Митник М. М., Стухляк П. Д. Телекомунікаційні системи та мережі. Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2017. 384 с.
21. Karpinski M., Korchenko A., Vikulov P., Kochan R. The Etalon Models of Linguistic Variables for Sniffing-Attack Detection, in Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), 2017 IEEE 9th International Conference on, 2017, pp. 258-264.
22. Computer Ontology of Mathematical Models of Cyclic Space-Time Structure SignalsNnamene, C.C., Lupenko, S., Volyanyk, O., Orobchuk, O.CEUR Workshop ProceedingsThis link is disabled., 2022, 3156, pp. 103–118
23. T. Lechachenko, R. Kozak, Y. Skorenkyy, O. Kramar, O. Karelina. Cybersecurity Aspects of Smart Manufacturing Transition to Industry 5.0 Model. CEUR Workshop Proceedings, 2023, 3628, pp. 325–329
24. Mishko O., Matiuk D., Derkach M. (2024) Security of remote iot system management by integrating firewall configuration into tunneled traffic. Scientific Journal of TNTU (Tern.), vol 115, no 3, pp. 122–129.
25. Микитишин, А. Г., Митник, М. М., Голотенко, О. С., & Карташов, В. В. (2023). Комплексна безпека інформаційних мережевих систем. Навчальний посібник для студентів спеціальності 174 «Автоматизація, комп’ютерно-інтегровані технології та робототехніка».
26. ДБН В.2.5 – 28 – 2018 Природне і штучне освітлення : вид. офіц. Київ : Мінрегіон України, 2018. 133 с
27. Методичний посібник для здобувачів освітнього ступеня «магістр» всіх спеціальностей денної та заочної (дистанційної) форм навчання «БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ» / В.С. Стручок – Тернопіль: ФОП Паляниця В.А., – 156 с. Отримано з https://elartu.tntu.edu.ua/handle/lib/39196.

Додаток А Публікація

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ

М А Т Е Р І А Л И

XIII НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

«ІНФОРМАЦІЙНІ МОДЕЛІ, СИСТЕМИ ТА ТЕХНОЛОГІЇ»

17-18 грудня 2025 року

ТЕРНОПІЛЬ
2025

УДК 004.056
Л. Гнатківський; А. Турчманович; Н. Загородна, к. т. н., доц.
(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

ДОСЛІДЖЕННЯ ПІДХОДІВ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ СИСТЕМ ВИЯВЛЕННЯ ВТОРГНЕНЬ

UDC 004.056
L. Hnatkivskyi; A. Turchmanovych; N. Zagorodna, PhD., Assoc. Prof.

RESEARCH OF APPROACHES TO IMPROVE EFFICIENCY OF INTRUSION DETECTION SYSTEMS

Сучасна мережева безпека базується на багаторівневому підході, що включає різноманітні технологічні рішення. Міжмережеві екрани (firewalls) контролюють потоки трафіку на основі встановлених правил, системи виявлення (IDS) та запобігання вторгненням (IPS) аналізують мережеву активність на предмет шкідливої поведінки, а SIEM-платформи агрегують та аналізують інформацію про події безпеки з усіх джерел.

У цій екосистемі IDS відіграють особливу роль інтелектуального аналітичного інструменту. На відміну від активних систем блокування, IDS функціонує в пасивному режимі моніторингу, аналізуючи копію мережевого трафіку без втручання в його потік, що дає можливість ідентифікувати потенційні загрози безпеці, виявляти складні багатоетапні атаки та несанкціоновані спроби доступу до корпоративних ресурсів. Однак стрімка еволюція методів кібератак, зростання обсягів мережевого трафіку та поява нових технологій вимагають постійного вдосконалення можливостей IDS для підтримання їхньої ефективності. Більшість джерел за способом виявлення атак поділяють IDS на такі, що працюють:

- на основі сигнатур (Signature-Based) та порівнюють мережеву активність з базою відомих патернів атак;
- на основі аномалій (Anomaly-Based) та класифікують мережеву активність на нормальну та аномальну на основі правил або евристик, здатні виявляти раніше невідомі загрози;
- на основі специфікації (Specification-Based) та відстежують процеси та порівнюють фактичні дані з очікуваною поведінкою програм;
- гібридні (Hybrid) поєднують різні методи, щоб забезпечити більш точне та ефективне виявлення атак.

Основними шляхами підвищення ефективності роботи IDS вважають впровадження методів штучного інтелекту, використання методів нечіткої логіки, розробку механізмів безперервного навчання та використання алгоритмів машинного навчання для аналізу контексту подій, що значно знижує кількість false positives.

Література

1. Rajasekaran, K. (2020). Classification and Importance of Intrusion Detection System. International Journal of Computer Science and Information Security,. 10. 44.
2. Alrayes, Fatma & Zakariah, Mohammed & Alzaylaee, Mohammed & Amin, Syed & Khan, Zafar Iqbal. (2025). Optimizing Intrusion Detection System (IDS) with Hybrid Random Forest and CNN-LSTM Models for Improved Accuracy and Efficiency. 10.21203/rs.3.rs-6766340/v1.

Додаток Б Лістинг для імпорту бібліотек та завантаження даних

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.model_selection import train_test_split

# Завантаження даних
columns = ['duration', 'protocol_type', 'service', 'flag',
'src_bytes',
           'dst_bytes', 'land', 'wrong_fragment', 'urgent',
'hot',
           'num_failed_logins', 'logged_in', 'num_compromised',
'root_shell',
           'su_attempted', 'num_root', 'num_file_creations',
'num_shells',
           'num_access_files', 'num_outbound_cmds',
'is_host_login',
           'is_guest_login', 'count', 'srv_count',
'serror_rate',
           'srv_serror_rate', 'rerror_rate', 'srv_rerror_rate',
           'same_srv_rate', 'diff_srv_rate',
'srv_diff_host_rate',
           'dst_host_count', 'dst_host_srv_count',
'dst_host_same_srv_rate',
           'dst_host_diff_srv_rate',
'dst_host_same_src_port_rate',
           'dst_host_srv_diff_host_rate',
'dst_host_serror_rate',
           'dst_host_srv_serror_rate', 'dst_host_rerror_rate',
           'dst_host_srv_rerror_rate', 'label', 'difficulty']

train_data = pd.read_csv('KDDTrain+.txt', names=columns)
test_data = pd.read_csv('KDDTest+.txt', names=columns)

# Видалення колонки difficulty (не використовується для
класифікації)
```

```
train_data = train_data.drop('difficulty', axis=1)
test_data = test_data.drop('difficulty', axis=1)

# Аналіз розподілу класів
print("Розподіл типів атак у навчальній вибірці:")
print(train_data['label'].value_counts())
```

Результат аналізу розподілу:

```
normal          67343
neptune         41214
satan           31683
ipsweep         31056

# Групування атак за категоріями
attack_mapping = {
    'normal': 'Normal',
    'neptune': 'DoS', 'smurf': 'DoS', 'pod': 'DoS', 'teardrop':
'DoS',
    'land': 'DoS', 'back': 'DoS', 'apache2': 'DoS', 'udpstorm':
'DoS',
    'processtable': 'DoS', 'mailbomb': 'DoS',

    'satan': 'Probe', 'portsweep': 'Probe', 'ipsweep': 'Probe',
    'nmap': 'Probe', 'mscan': 'Probe', 'saint': 'Probe',

    'warezclient': 'R2L', 'warezmaster': 'R2L', 'guess_passwd':
'R2L',
    'imap': 'R2L', 'ftp_write': 'R2L', 'multihop': 'R2L', 'phf':
'R2L',
    'spy': 'R2L', 'sendmail': 'R2L', 'named': 'R2L',
'snmpgetattack': 'R2L',
    'snmpguess': 'R2L', 'xlock': 'R2L', 'xsnoop': 'R2L', 'worm':
'R2L',
```

```
    'buffer_overflow': 'U2R', 'rootkit': 'U2R', 'loadmodule':
'U2R',
    'perl': 'U2R', 'sqlattack': 'U2R', 'xterm': 'U2R', 'ps':
'U2R'
}

train_data['attack_category'] =
train_data['label'].map(attack_mapping)
test_data['attack_category'] =
test_data['label'].map(attack_mapping)

# Обробка невідомих атак у тестовій вибірці
test_data['attack_category'].fillna('Unknown', inplace=True)


# Ідентифікація категоріальних ознак
categorical_features = ['protocol_type', 'service', 'flag']

# Label Encoding для категоріальних ознак
label_encoders = {}
for feature in categorical_features:
    le = LabelEncoder()
    train_data[feature] = le.fit_transform(train_data[feature])
    # Обробка нових значень у тестовій вибірці
    test_data[feature] = test_data[feature].map(
        lambda x: le.transform([x])[0] if x in le.classes_ else
-1
    )
    label_encoders[feature] = le

# Кодування міток класів
le_target = LabelEncoder()
y_train = le_target.fit_transform(train_data['attack_category'])
y_test = le_target.transform(test_data['attack_category'])
# Підготовка ознак
X_train = train_data.drop(['label', 'attack_category'], axis=1)
X_test = test_data.drop(['label', 'attack_category'], axis=1)
```

Додаток В Лістинги для навчання та тестування класифікаційних моделей XGBoost та SVM

```
import xgboost as xgb

# Навчання
print("\nНавчання XGBoost...")
start_time = time.time()

xgb_model = xgb.XGBClassifier(
    n_estimators=100,
    max_depth=10,
    learning_rate=0.1,
    subsample=0.8,
    colsample_bytree=0.8,
    objective='multi:softmax',
    num_class=len(le_target.classes_),
    random_state=42,
    n_jobs=-1
)

xgb_model.fit(X_train_scaled, y_train)
xgb_train_time = time.time() - start_time

# Прогнозування
start_time = time.time()
y_pred_xgb = xgb_model.predict(X_test_scaled)
xgb_pred_time = time.time() - start_time

# Оцінка продуктивності
xgb_accuracy = accuracy_score(y_test, y_pred_xgb)
print(f"XGBoost - Час навчання: {xgb_train_time:.2f}c")
print(f"XGBoost - Час прогнозування: {xgb_pred_time:.2f}c")
print(f"XGBoost - Accuracy: {xgb_accuracy:.4f}")

print("\nClassification Report (XGBoost):")
```

```
print(classification_report(y_test, y_pred_xgb,
target_names=le_target.classes_))
```

Результати XGBoost:

```
XGBoost - Час навчання: 38.91с
XGBoost - Час прогнозування: 0.52с
XGBoost - Accuracy: 0.8156

Classification Report (XGBoost):
              precision    recall  f1-score   support

         DoS       0.98      0.95      0.96      7458
      Normal       0.87      0.93      0.90      9711
       Probe       0.85      0.81      0.83      2421
         R2L       0.48      0.41      0.44      2754
         U2R       0.42      0.35      0.38       200

    accuracy                           0.82     22544
   macro avg       0.72      0.69      0.70     22544
weighted avg       0.81      0.82      0.81     22544


from sklearn.svm import SVC

# Через обмеження SVM у часі, навчаємо на підвибірці
train_sample_indices = np.random.choice(len(X_train_scaled),
10000, replace=False)
X_train_sample = X_train_scaled[train_sample_indices]
y_train_sample = y_train[train_sample_indices]

print("\nНавчання SVM (на вибірці 10000 зразків)...")
start_time = time.time()

svm_model = SVC(
    kernel='rbf',
```

```
    C=10,
    gamma='scale',
    class_weight='balanced',
    random_state=42
)

svm_model.fit(X_train_sample, y_train_sample)
svm_train_time = time.time() - start_time

# Прогнозування
start_time = time.time()
y_pred_svm = svm_model.predict(X_test_scaled)
svm_pred_time = time.time() - start_time

# Оцінка
svm_accuracy = accuracy_score(y_test, y_pred_svm)
print(f"SVM - Час навчання: {svm_train_time:.2f}c")
print(f"SVM - Час прогнозування: {svm_pred_time:.2f}c")
print(f"SVM - Accuracy: {svm_accuracy:.4f}")
```

Результати SVM:

```
SVM - Час навчання: 127.45c
SVM - Час прогнозування: 18.32c
SVM - Accuracy: 0.7523
```