ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ТЕЛЕКОМУНІКАЦІЙ
КАФЕДРА ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ ТА МЕРЕЖ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «МЕТОДИ ВИЯВЛЕННЯ АНОМАЛІЙ У ТЕЛЕКОМУНІКАЦІЙНИХ МЕРЕЖАХ»

на здобуття освітнього ступеня магістра
зі спеціальності 172 Електронні комунікації та радіотехніка
(код, найменування спеціальності)
освітньо-професійної програми Телекомунікаційні системи та мережі
(назва)

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

_____ Євгеній СОРОКА
(підпис) *(Ім'я ПРІЗВИЩЕ)*

Виконав: здобувач вищої освіти гр. _ТСДМ-63_
Євгеній СОРОКА
Ім'я, ПРІЗВИЩЕ

Керівник: Вікторія КОРЕЦЬКА
к.п.н., доцент *Ім'я, ПРІЗВИЩЕ*

Рецензент: _____
Ім'я, ПРІЗВИЩЕ

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
Навчально-науковий інститут Телекомунікацій

Кафедра ___Телекомунікаційних систем та мереж_____
Ступінь вищої освіти __магістр_____
Спеціальність ___172 Електронні комунікації та радіотехніка _____
Освітньо-професійна програма _____Телекомунікаційні системи та мережі_____

ЗАТВЕРДЖУЮ
Завідувач кафедри __ТСМ_____
_____ Віктор ЗАЇКА
«____» _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Сороці Євгенію Сергійовичу
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Методи виявлення аномалій у телекомунікаційних мережах
керівник кваліфікаційної роботи Вікторія КОРЕЦЬКА, к.п.н., доцент
(Ім'я, ПРІЗВИЩЕ науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «30» 10. 2025 р. № 467

2. Строк подання кваліфікаційної роботи «___» _____ 2025 р.

3. Вихідні дані до кваліфікаційної роботи:
1. Аномалії у телекомунікаційних мережах.
2. СВВ.
3. Побудова нейромережевого аналізатору.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Визначення класів мережевих вузлів.
2. Дослідження рішень у сфері СВВ.
3. Проєктування модуля СВВ.
4. Побудова нейромережевого аналізатору.

5. Перелік ілюстративного матеріалу: ___презентація на ____слайдах_____

6. Дата видачі завдання «___» _____ 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір та аналіз науково-технічної літератури	15.10.2025	Викон.
2	Розробка першого розділу	22.10.2025	Викон.
3	Розробка другого розділу	02.11.2025	Викон.
4	Розробка третього розділу	12.11.2025	Викон.
5	Розробка четвертого розділу	22.11.2025	Викон.
6	Висновки, вступ, реферат	02.12.2025	Викон.
7	Розробка презентації	12.12.2025	Викон.

Здобувач вищої освіти _____ Євгеній СОРОКА
(підпис) *(Ім'я ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи _____ Вікторія КОРЕЦЬКА
(підпис) *(Ім'я ПРІЗВИЩЕ)*

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
Навчально-науковий інститут телекомунікацій
ПОДАННЯ
ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ
ЩОДО ЗАХИСТУ КВАЛІФІКАЦІЙНОЇ РОБОТИ
на здобуття освітнього ступеня магістра

Направляється здобувач Сорока Є.С. до захисту кваліфікаційної роботи
(прізвище та ініціали)
за спеціальністю 172 Електронні комунікації та радіотехніка
(код, найменування спеціальності)
освітньо-професійної програми Телекомунікаційні системи та мережі
(назва)
на тему: «Методи виявлення аномалій у телекомунікаційних мережах»

Кваліфікаційна робота і рецензія додаються.

Директор ННІТ _____ Владислав КРАВЧЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Висновок керівника кваліфікаційної роботи

Здобувач Сорока Євгеній Сергійович показав гарну теоретичну та інженерну підготовку, уміння володіти новими комп'ютерними технологіями, користуватися навчальною, довідковою і науково-технічною літературою. Працюючи над завданнями, які були поставлені керівником, проявив ініціативність, сумлінність та хист до наукової роботи.

Все це дозволяє оцінити виконану кваліфікаційну роботу здобувача Сороки Євгенія Сергійовича на оцінку «відмінно» та присвоїти йому кваліфікацію магістр з електронних комунікацій та радіотехніки.

Керівник кваліфікаційної роботи _____ Вікторія КОРЕЦЬКА
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

«____» _____ 20___ року

Висновок кафедри про кваліфікаційну роботу

Кваліфікаційна робота розглянута. Здобувач Сорока Є.С. допускається до захисту даної роботи в Екзаменаційній комісії.

Завідувач кафедри телекомунікаційних систем та мереж _____ Віктор ЗАЇКА
(назва) *(підпис)* *(Ім'я, ПРІЗВИЩЕ)*

«____» _____ 20___ року

ВІДГУК РЕЦЕНЗЕНТА
на кваліфікаційну роботу
на здобуття освітнього ступеня магістра

здобувача вищої освіти Сороки Євгенія Сергійовича
(прізвище, ім'я, по батькові)
на тему «Методи виявлення аномалій у телекомунікаційних мережах»

Актуальність.
Стрімкий розвиток телекомунікаційних мереж, збільшення кількості підключених пристроїв та зростання обсягів переданих даних супроводжуються різким підвищенням рівня кіберзагроз. Традиційні сигнатурні системи захисту виявляються малоефективними проти нових або модифікованих атак, що не мають опису у базах даних. В умовах цифровізації промислових, транспортних і державних інфраструктур зростає потреба у розумних системах моніторингу, здатних автоматично виявляти аномалії у мережевому трафіку без активного втручання в роботу мережі. Застосування методів машинного навчання та кластеризації (SOM, GNG, DBSCAN, AutoML) є перспективним напрямом, що дозволяє створювати адаптивні системи виявлення загроз у реальному часі.
Позитивні сторони.
1. У роботі побудовано чітку математичну модель опису вузлів мережі та векторів ознак.
2. У роботі запропоновано поєднання традиційних методів аналізу трафіку з AutoML і ансамблевими моделями класифікації, що підвищує точність детекції.
Недоліки.
1. У роботі питання продуктивності системи при роботі з великими обсягами трафіку розглянуто поверхнево.
2. У роботі було б доцільно додати порівняння результатів моделі з існуючими рішеннями (наприклад, Snort, Suricata, Zeek).

Відзначені зауваження не впливають на загальну позитивну оцінку магістерської кваліфікаційної роботи

Висновок: *кваліфікаційна робота на здобуття ступеня магістра заслуговує оцінку "відмінно", а здобувач Сорока Євгеній Сергійович заслуговує присвоєння кваліфікації: магістр з електронних комунікацій та радіотехніки.*

Рецензент
науковий ступінь, вчене звання _____ (підпис) _____ *(Ім'я ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 82 стор., 15 рис., 30 джерел.

Мета роботи – розроблення та дослідження методу автоматизованого виявлення аномальної активності у телекомунікаційних мережах на основі аналізу мережевого трафіку з використанням алгоритмів машинного навчання, що забезпечують підвищення точності класифікації мережевих вузлів і своєчасне виявлення кіберзагроз у реальному часі.

Об’єкт дослідження – процеси передавання та аналізу даних у телекомунікаційних мережах, що впливають на їхню безпеку та стабільність функціонування.

Предмет дослідження – методи та алгоритми машинного навчання і кластеризації, призначені для виявлення аномалій і класифікації вузлів телекомунікаційних мереж за їхнім профілем трафіку.

Короткий зміст роботи: у кваліфікаційній роботі проведено дослідження методів виявлення аномалій у телекомунікаційних мережах і розроблено підхід до створення модулів, здатних виявляти відхилення від норми в реальному часі. Проаналізовано класифікації IDS, методи збору даних та поєднання сигнатурних і поведінкових підходів. Особливу увагу приділено машинному навчанню, зокрема алгоритмам кластеризації (SOM, GNG). Створено прототип модуля Python, що підтвердив ефективність виявлення аномалій. Робота доводить доцільність інтеграції сигнатурних і адаптивних методів, а також перспективність гібридних IDS, здатних до самонавчання у реальному часі.

КЛЮЧОВІ СЛОВА: ВИЯВЛЕННЯ АНОМАЛІЙ, ТЕЛЕКОМУНІКАЦІЙНІ МЕРЕЖІ, СИСТЕМИ ВИЯВЛЕННЯ ВТОРГНЕНЬ (IDS), КІБЕРБЕЗПЕКА, МАШИННЕ НАВЧАННЯ, КЛАСТЕРИЗАЦІЯ, САМООРГАНІЗОВУВАНІ КАРТИ КОХОНЕНА (SOM), ЗРОСТАЮЧИЙ НЕЙРОННИЙ ГАЗ (GNG), AUTOML, PYTHON, МЕРЕЖЕВИЙ ТРАФІК

ABSTRACT

Text part of the master's qualification work: 82 pages, 15 figures, 30 sources.

The purpose of the work is to develop and study a method for automated detection of anomalous activity in telecommunication networks based on network traffic analysis using machine learning algorithms that ensure increased accuracy of network node classification and timely detection of cyber threats in real time.

Object of the study is the processes of data transmission and analysis in telecommunication networks that affect their security and stability of operation.

Subject of the study is methods and algorithms of machine learning and clustering designed to detect anomalies and classify nodes of telecommunication networks according to their traffic profile.

Summary of the work: in the qualification work, a study of methods for detecting anomalies in telecommunication networks was conducted and an approach to creating modules capable of detecting deviations from the norm in real time was developed. IDS classifications, data collection methods, and a combination of signature and behavioral approaches were analyzed. Special attention is paid to machine learning, in particular clustering algorithms (SOM, GNG). A prototype of a Python module was created, which confirmed the effectiveness of anomaly detection. The work proves the feasibility of integrating signature and adaptive methods, as well as the prospects of hybrid IDS capable of self-learning in real time.

KEYWORDS: ANOMALIES DETECTION, TELECOMMUNICATION NETWORKS, INTRUSION DETECTION SYSTEMS (IDS), CYBERSECURITY, MACHINE LEARNING, CLUSTERING, SELF-ORGANIZING KOHONEN MAPS (SOM), GROWING NEURAL GAS (GNG), AUTOML, PYTHON, NETWORK TRAFFIC

ЗМІСТ

ВСТУП 9
РОЗДІЛ 1 ВИЗНАЧЕННЯ КЛАСІВ МЕРЕЖЕВИХ ВУЗЛІВ 11
1.1 Виявлення аномалій в активності вузлів по мережному трафіку 11
1.2 Математична модель 14
1.3 Реалізація моделі детекції 23
РОЗДІЛ 2 ДОСЛІДЖЕННЯ РІШЕНЬ У СФЕРІ СВВ 27
2.1 Типи систем виявлення вторгнень 27
2.2 Типи IDS 39
2.3 Діючі системи IDS 42
РОЗДІЛ 3 ПРОЄКТУВАННЯ МОДУЛЯ СВВ 45
3.1 Можливі атаки 45
3.2 Ймовірні атаки на різні рівні 57
3.3 Опис типової архітектури IDS 60
РОЗДІЛ 4 ПОБУДОВА НЕЙРОМЕРЕЖЕВОГО АНАЛІЗАТОРУ 64
4.1 Опис принципу 64
4.2 Архітектура модуля 73
4.3 Робота модуля та інтерпритація результатів 76
ВИСНОВКИ 81
ПЕРЕЛІК ПОСИЛАНЬ 83
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ 86

ВСТУП

Стрімкий розвиток телекомунікаційних мереж, збільшення кількості підключених пристроїв та зростання обсягів передаваних даних створюють нові можливості для цифрової трансформації суспільства, але водночас підвищують рівень кіберзагроз. Сучасні мережі функціонують у складному і динамічному середовищі, де атакувальні дії можуть мати різний характер – від простих спроб несанкціонованого доступу до складних багаторівневих атак, що використовують вразливості як у прикладних сервісах, так і у протоколах нижчих рівнів. У таких умовах традиційних засобів захисту, які базуються лише на статичних правилах або сигнатурних базах, уже недостатньо.

Виявлення аномалій у мережевій активності стає ключовим завданням у забезпеченні інформаційної безпеки телекомунікаційних систем. Його сутність полягає у фіксації відхилень від «нормальної» поведінки мережі чи окремих її вузлів. Це дозволяє своєчасно ідентифікувати нові та раніше невідомі загрози, які ще не мають опису у сигнатурних базах. Водночас вирішення цього завдання супроводжується низкою труднощів: необхідністю формування адекватної моделі нормального функціонування, значним обсягом вхідних даних, ризиком перенавчання та потребою у швидкій обробці інформації у реальному часі.

У сучасних дослідженнях усе ширше застосовуються методи машинного навчання, які дозволяють автоматизувати процес виявлення аномалій. Перспективним напрямом є використання кластеризаційних алгоритмів, зокрема самоорганізовуваних карт Кохонена та алгоритмів «зростаючого нейронного газу» (GNG та IGNG). Їхня ключова перевага полягає у здатності формувати адаптивні моделі без жорсткої прив’язки до наперед заданої кількості кластерів, що забезпечує гнучкість і пристосовуваність до змін умов експлуатації мереж.

Об’єктом дослідження є процеси передачі даних у телекомунікаційних мережах. Предметом дослідження є методи та алгоритми виявлення аномальної активності у цих мережах.

Наукова новизна роботи полягає у поєднанні класичних методів аналізу трафіку з адаптивними алгоритмами машинного навчання, що дозволяє підвищити точність і надійність виявлення загроз. Практичне значення отриманих результатів визначається можливістю використання розробленого прототипу у системах кіберзахисту телекомунікаційних мереж різного масштабу.

1 ВИЗНАЧЕННЯ КЛАСІВ МЕРЕЖЕВИХ ВУЗЛІВ

1.1 Виявлення аномалій в активності вузлів по мережному трафіку

У сучасних умовах цифровізації промисловості та транспорту мережі передавання даних стають невід'ємною складовою виробничої інфраструктури. Вони широко використовуються на промислових підприємствах, а також у складних технічних системах, таких як великі залізничні склади чи морські судна. Передавана через такі мережі інформація нерідко має критичне значення для безперервності технологічних процесів, а тому виникає об'єктивна потреба у її захисті від несанкціонованого доступу, модифікації чи порушення цілісності. Саме для цього створюються та застосовуються спеціалізовані засоби забезпечення мережевої безпеки.

Однак ефективність цих засобів безпосередньо залежить від повноти інформації про мережеву інфраструктуру. Необхідно мати відомості про всі наявні вузли, їхнє розташування в адресному просторі, функціональне призначення, а також характер їхньої взаємодії з іншими компонентами системи. Без таких даних будь-які захисні механізми працюватимуть неефективно або навіть створюватимуть хибне відчуття безпеки.

У даній роботі пропонується один із методів автоматизованого визначення типів мережевих вузлів із використанням сучасних методів машинного навчання. Цей підхід дозволяє підвищити ефективність процесу інвентаризації та моніторингу мережевих ресурсів, а також забезпечити раннє виявлення аномалій у роботі інформаційної інфраструктури.

На перший погляд може здатися, що подібні задачі вже давно вирішуються за допомогою відомих інструментів. Прикладом є широко застосовуваний засіб Nmap, який здатний виявляти та класифікувати мережеві вузли. Проте використання таких інструментів пов'язане з активним скануванням, що в умовах промислових мереж є небажаним або навіть небезпечним. Наприклад, надмірна кількість опитувальних запитів може викликати збої у програмованих логічних контролерах (ПЛК), що

вкрай критично для виробничих процесів, де кожен зайвий пакет може призвести до аварійного стану або некоректної роботи обладнання.

Крім цього, більшість традиційних засобів орієнтовані на наперед визначений набір класів вузлів та спираються на жорстко закріплені правила. Це означає, що будь-який новий або нетиповий пристрій, не передбачений правилами, залишиться невиявленим. Запропонований метод машинного навчання позбавлений цього обмеження. Він дозволяє динамічно адаптуватися до особливостей мережі та ідентифікувати вузли за їхньою реальною поведінкою у трафіку.

Ще однією перевагою методу є можливість паралельного виявлення аномалій у взаємодії вузлів. Це дає змогу не лише створити повну «картину» мережі, а й вчасно зафіксувати підозрілу активність, яка може свідчити про внутрішню помилку, несправність обладнання чи потенційну атаку.

Особливо актуальним цей підхід стає у ситуаціях, коли активне сканування є недоцільним або неможливим. Тоді єдиним способом отримати повну картину мережевої інфраструктури залишається трудомістка ручна робота. Автоматизація цього процесу на основі профілювання трафіку дозволяє значно скоротити часові витрати, усунути людський фактор і водночас зберегти стабільність функціонування захищуваної системи.

Концепція методу

Основна ідея запропонованого методу виявлення та класифікації вузлів у телекомунікаційних мережах багато в чому перегукується з принципами функціонування систем виявлення вторгнень (СВВ, IDS). Подібно до IDS, він ґрунтується на пасивному прослуховуванні всього мережевого трафіку без внесення змін у саму мережу, після чого відбувається його багаторівнева обробка. На етапі попереднього аналізу трафік сегментується на окремі потоки, які відповідають встановленим з’єднанням між вузлами з унікальними адресами та певним набором протоколів різних рівнів моделі OSI. Для кожного вузла підраховується статистика частоти використання протоколів і службових

повідомлень, після чого формується вектор ознак, що відображає його унікальний «мережевий портрет» і дозволяє виокремлювати характерні риси поведінки.

Далі, на основі попередньо зібраних і розмічених даних тренувальної вибірки, із застосуванням інструментів автоматизованого машинного навчання (AutoML) здійснюється підбір оптимальної архітектури моделі. Це може бути ансамбль дерев рішень, глибока нейронна мережа або інші алгоритми, які найкраще підходять для класифікації вузлів за їхнім профілем трафіку. Після завершення навчання така модель здатна автоматично приймати на вхід вектори ознак невідомих вузлів і визначати їхній тип (наприклад, сервер, клієнтська станція, мережеве обладнання тощо). Таким чином створюється повністю автоматизована система маркування вузлів, що дозволяє значно знизити навантаження на аналітиків та скоротити час виявлення нових пристроїв у мережі.

Для розв’язання задачі виявлення аномалій застосовується спрощена методика. На початковому етапі для кожного вузла формується набір ознак, після чого використовуються алгоритми навчання без учителя (наприклад, k-means, DBSCAN чи SOM-карти Кохонена). Оптимальна кількість кластерів визначається за допомогою метрик якості кластеризації, після чого вузли групуються відповідно до подібності їхньої поведінки у трафіку. Номери сформованих кластерів виступають своєрідними «мітками», які використовуються для подальшого навчання нейронної мережі. Коли у мережі з’являються нові потоки, формується оновлений вектор ознак, що подається на вхід нейромережі. Якщо прогнозований клас вузла збігається з раніше зафіксованим, система вважає його поведінку штатною; у випадку ж відхилення формується сигнал тривоги, який вказує на появу потенційної аномалії чи загрози.

Для більш глибокої оцінки критичності виявлених інцидентів можуть застосовуватися методи ієрархічної кластеризації. Вони дозволяють визначати «відстань» між вузлами у кластерній ієрархії та оцінювати рівень відхилень від норми не лише за якісними, а й за кількісними характеристиками. Це дає можливість розрізняти події з різним рівнем небезпеки. Наприклад, поява нового клієнтського вузла у файлового сервера може розглядатися як подія з низьким

рівнем критичності, тоді як формування нового керуючого з’єднання до комутатора, яке раніше ніколи не спостерігалося, вказує на високу ймовірність атакуючих дій і потребує негайного реагування.

У результаті описаний метод формує комплексний підхід до аналізу мережевого трафіку, який поєднує принципи IDS, інструменти машинного навчання та методи кластеризації, створюючи універсальну основу як для автоматизованої ідентифікації вузлів, так і для ефективного виявлення аномалій у великих розподілених мережах.

1.2 Математична модель

Математика справедливо вважається універсальною мовою науки, яка дає змогу формалізувати та узагальнювати складні технічні процеси. Саме тому спробуємо описати задачу визначення мережевого класу вузла на основі його трафіку у вигляді математичної моделі, що дозволить не лише чітко зафіксувати вихідні умови, але й закласти основу для подальшого використання алгоритмів машинного навчання.

На вхід розглядуваної моделі надходить мережевий трафік захищуваної системи, тобто сукупність пакетів даних, що передаються між вузлами. Для зручності введемо визначення: набір мережевих пакетів, які використовують однаковий стек протоколів на всіх задіяних рівнях моделі OSI та спрямовані від одного джерела (IP-адреси відправника) до одного приймача (IP-адреси одержувача), будемо називати потоком (*Flow*). Таким чином, потік є елементарною одиницею спостереження у процесі аналізу трафіку.

Кожен потік *Flow* описується у вигляді вектора ознак, що містить числові характеристики, які відображають структуру та властивості використаних протоколів. Формально це можна записати так:

$$Flow = \{feature\}_{i=0}^{numProto} \quad (1.1)$$

де *numProto* – це кількість мережевих протоколів, присутніх у конкретному пакеті чи потоці. Чим складніший стек протоколів використовується (наприклад, поєднання TCP/IP, HTTP, TLS тощо), тим більшою буде розмірність вектора ознак.

Оскільки мережа складається з багатьох таких потоків, для моделі важливо розглядати не окремі з них, а повний набір, що характеризує діяльність вузла у цілому. Тому сукупність усіх потоків, пов'язаних із конкретним вузлом, позначимо як *Flows*. Вона визначається наступним чином:

$$Flows = \{Flow\}_{j=0}^{lenFlows} \quad (1.2)$$

де *lenFlows* – це кількість потоків, зафіксованих для аналізованого вузла протягом визначеного інтервалу часу.

Таким чином, кожен вузол у мережі можна описати через відповідний набір потоків, що утворюють багатовимірний простір ознак. Побудована у такий спосіб математична модель дозволяє здійснювати класифікацію вузлів за їхнім «мережевим почерком» і формувати основу для виявлення відхилень у поведінці, які можуть свідчити про аномалії чи потенційні вторгнення.

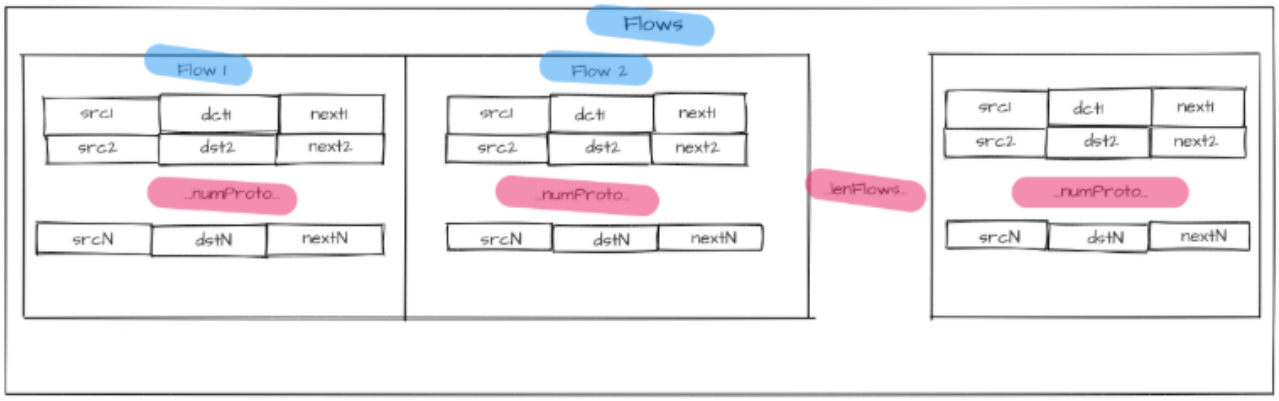


Рис. 1.1 Група потоків

Для унікальної ідентифікації кожного мережевого вузла, що входить до множини класифікованих вузлів *Hosts*, використовується набір адрес джерела (*source addresses*), зафіксованих на усіх рівнях мережевої моделі, де застосовуються відповідні протоколи. Іншими словами, ідентифікатор вузла формується як комбінація його адрес на різних рівнях протокольного стека. Це дозволяє забезпечити точність розпізнавання навіть у випадку, коли окремі вузли мають

однакові ідентифікатори на певному рівні (наприклад, локальні MAC-адреси або IP-адреси).

Кількість рівнів адресації позначається як *numProto* і визначається числом протоколів, що беруть участь у формуванні пакета даних. Таким чином, кожен елемент множини *Hosts* може бути поданий як вектор вкладених адрес, що охоплює всі рівні протоколів. Повна множина *Hosts* довжиною *lenHosts* описується наступним виразом:

$$Hosts = \{host\}_{i=0}^{numHosts} = \left\{\{host.src\}_{j=0}^{numProto}\right\}_{i=0}^{numHosts} \quad (1.3)$$

Цей запис означає, що кожен вузол *host* характеризується власним набором джерельних адрес (*host.src*), які визначаються для всіх рівнів протокольного стеку від фізичного до прикладного. Саме ця багаторівнева адресація і використовується для побудови узагальненого ідентифікатора вузла, придатного для подальшої класифікації.

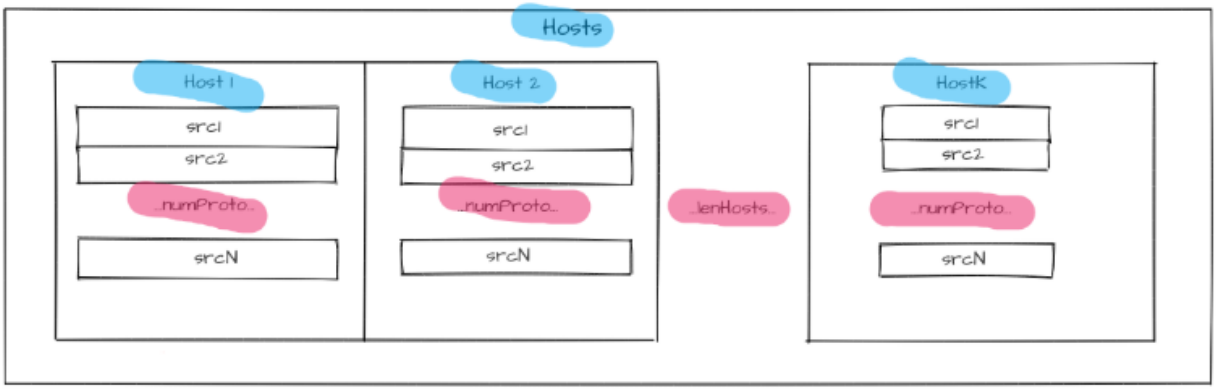


Рис. 1.2 Ідентифікатори вузлів

Визначення характеристик мережевого вузла є одним із ключових етапів побудови математичної моделі його поведінки у мережі. Кожен вузол описується низкою параметрів, які відображають як внутрішні властивості (кількість потоків, набір використовуваних протоколів), так і зовнішні зв’язки з іншими вузлами (частка трафіку, спрямованого на певні класи пристроїв).

Перш за все, важливим показником є кількість потоків (*host.num_flows*), що асоціюються з конкретним вузлом. Цей параметр відображає активність пристрою у мережі та дозволяє оцінити його функціональне навантаження. Не менш значущим є набір протоколів, що використовуються у цих потоках. Саме частота їх

застосування формує своєрідний «відбиток» вузла, який можна використати для класифікації.

Для цього вводиться вектор протокольних характеристик:

$$host.proto = \{frequency\}_{i=0}^{host.numKnownProto} \quad (1.4)$$

Довжина цього вектора (*host.numKnownProto*) відповідає кількості всіх відомих протоколів, що зустрічаються у глобальному масиві потоків. Кожен елемент *frequency* нормалізується у діапазоні від 0 до 1 та показує, наскільки часто вузол використовує відповідний протокол у порівнянні з усіма іншими потоками. Таким чином, формується унікальний профіль використання протоколів для кожного вузла.

Окрім протокольних характеристик, вузол має набір ярликів (міток), які визначають його приналежність до певних класів:

$$host.labels = \{label\}_{k}^{numTypes} \quad (1.5)$$

Класифікація відбувається за двома незалежними ознаками:

1. **Роль вузла у мережі** (наприклад, програмований логічний контролер – ПЛК, автоматизоване робоче місце – АРМ, сервер АСУ, файловий сервер тощо).
2. **Операційна система**, під управлінням якої функціонує вузол (Windows XP, Windows 7, Linux Ubuntu, Linux CentOS та ін.).

Також кожен вузол характеризується **долею потоків**, що поєднують його з іншими вузлами, які належать до певних класів. Цей показник позначається як *host.connectedWith* та відображає специфіку взаємодії між різними сегментами мережі. Наприклад, для файлового сервера очікувано домінує трафік від клієнтських робочих станцій, тоді як для ПЛК характерні керуючі з’єднання від систем диспетчерського контролю.

Усі вищезгадані параметри інтегруються у характеристичний вектор вузла *host.vector*, що слугує основою для аналізу за допомогою алгоритмів машинного навчання. Він має вигляд:

$$host.vector = \{host.proto, host.some_inf\} \quad (1.6)$$

де *host.some_inf* – це додаткова інформація про вузол (наприклад, метадані або статистичні параметри), яка може покращувати точність класифікації. У сукупності з частотними характеристиками використання протоколів це утворює профіль трафіку вузла.

Повний набір таких векторів для всіх вузлів мережі формує масив вхідних даних для моделі:

$$data = \{host.vector\}_i^{lenHosts} \tag{1.7}$$

Цей масив є узагальненим описом усієї системи, що дозволяє алгоритмам навчання виявляти закономірності та виконувати класифікацію.

Для того щоб модель могла здійснювати навчання з учителем, потрібна апріорна інформація про класи вузлів. Тобто, для кожного елемента *host_i* із множини *Hosts* має бути заздалегідь визначено його тип (*label*). Це означає, що дамп трафіку, який використовується для навчання, повинен складатися з набору потоків, для яких відомі не лише адреси джерела, але й коректні мітки класів вузлів:

$$host.labels = \{label\}_k^{numTypes} \tag{1.8}$$

Таким чином, система формує структурований набір навчальних даних, де кожен вузол описується як вектор ознак із прив'язаною класовою належністю. Це дає змогу будувати точні моделі машинного навчання для автоматизованої ідентифікації, класифікації та подальшого моніторингу мережевих вузлів.

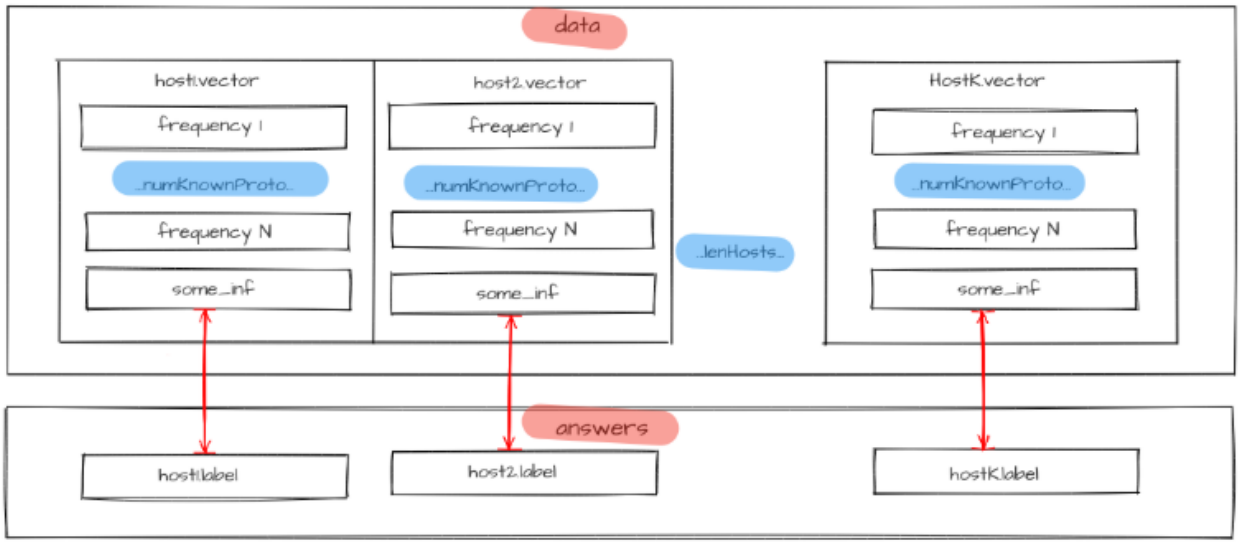


Рис. 1.3 Дані для навчання моделі

У процесі побудови та перевірки моделі машинного навчання наявний масив трафіку з попередньо розміченими вузлами підлягає розподілу на дві основні частини: тренувальну вибірку та тестову вибірку. Такий підхід є класичним у теорії машинного навчання, оскільки він дозволяє оцінити здатність моделі не лише запам’ятовувати вхідні дані, а й узагальнювати закономірності для нових, раніше невідомих прикладів.

На першому етапі здійснюється навчання моделі. Для цього використовується тренувальна вибірка, у якій для кожного вузла заздалегідь відомі правильні класи (*host.label*). Завдяки цьому модель має змогу «навчитися» знаходити залежності між характеристичними векторами вузлів (*host.vector*) та їхніми класами.

Після завершення навчання модель переходить до етапу тестування, у якому на вхід подається тестова вибірка — дані, що не використовувалися на етапі тренування. Модель виконує класифікацію вузлів і видає результати у вигляді передбачених міток *host.labelPredicted*. Далі отримані передбачення порівнюються із заздалегідь відомими справжніми мітками *host.label*.

Для кількісної оцінки якості моделі застосовуються стандартні метрики точності класифікації, серед яких:

- **Accuracy** – загальна частка правильно класифікованих вузлів серед усіх;
- **Precision** – точність визначення конкретного класу (яка частка передбачених позитивних випадків є справді правильними);
- **Recall** – повнота або чутливість (яка частка правильних прикладів була знайдена моделлю);
- **F1-score** – гармонійне середнє між точністю та повнотою, що дозволяє збалансовано оцінити модель у разі наявності нерівномірно представлених класів.

Якщо значення обраних метрик відповідають заздалегідь встановленим вимогам до якості класифікації, модель вважається успішно побудованою та зберігається для подальшого використання у задачах розпізнавання. У випадку ж, коли отримані результати є незадовільними, виконується корекція структури моделі

(наприклад, зміна архітектури нейронної мережі, кількості шарів, типу функцій активації) або її гіперпараметрів (швидкість навчання, розмір батчу, коефіцієнти регуляризації тощо).

Після цього цикл «навчання → тестування → налаштування моделі» повторюється доти, доки не буде досягнуто прийнятного рівня передбачувальної здатності. Такий ітеративний підхід дає змогу поступово покращувати якість роботи системи та забезпечує її адаптацію до особливостей конкретного мережевого середовища.

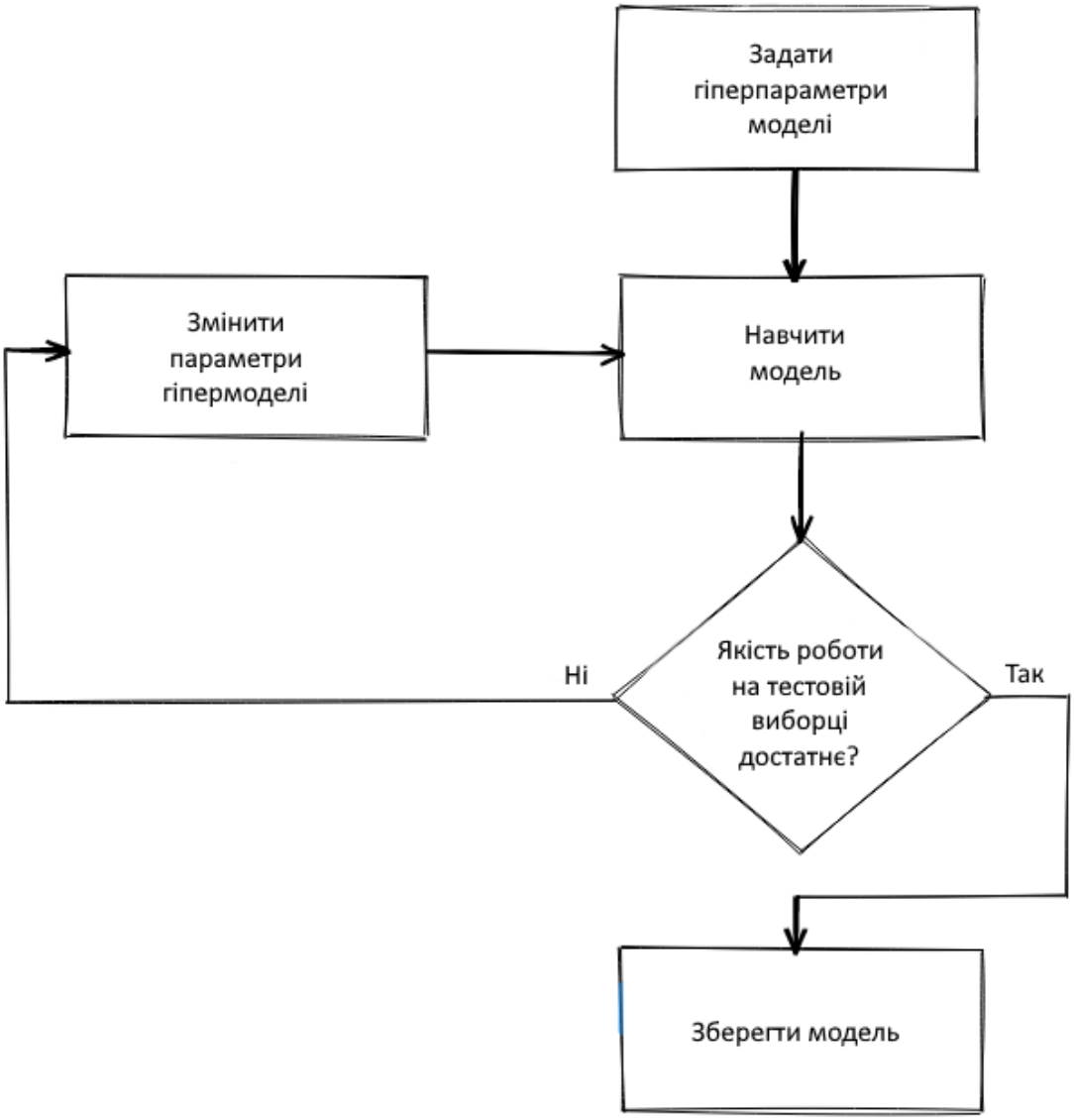


Рис. 1.4 Процес створення моделі

Другим важливим завданням після класифікації вузлів є виявлення аномальної поведінки у мережевому трафіку. Аномалії можуть свідчити як про збої

у роботі обладнання чи помилки конфігурації, так і про потенційні кіберзагрози — спроби несанкціонованого доступу або зміну характеру взаємодії вузлів.

Для вирішення цього завдання використовується методика, що ґрунтується на аналізі характеристичних векторів вузлів. На першому етапі з існуючого набору потоків *Flows* формуються вектори ознак *host.vector*. Ці вектори відображають усі ключові параметри взаємодії вузлів (частоту використання протоколів, кількість з'єднань із певними класами пристроїв тощо) та слугують основою для подальшої кластеризації.

Далі модель-кластеризатор здійснює розподіл вузлів на групи. Кожному вузлу призначається мітка *host.label* відповідно до тієї групи, у яку він потрапив. У результаті багатовимірний простір, розмірність якого дорівнює довжині вектора ознак *host.vector*, розділяється на певну кількість областей. Кількість цих областей визначається числом кластерів, сформованих алгоритмом. Паралельно застосовується ієрархічна кластеризація, яка створює деревоподібну структуру зв'язків між вузлами. Це дає змогу не лише згрупувати вузли, а й визначати відстань між ними у межах ієрархії, що стане важливим для оцінки критичності відхилень.

Після формування кластерів система переходить у режим моніторингу. При надходженні нових потоків *Flows* відповідні вектори ознак вузлів оновлюються й подаються на вхід моделі. Кластеризатор призначає їм нові мітки *host.new_label*, які порівнюються з початковими мітками *host.label*. Якщо виявляється розбіжність:

$$(host_i.label \neq host_i.new_label), \tag{1.9}$$

це інтерпретується як виявлення аномалії. Потоки, що призвели до зміни характеристичного вектора вузла (*host.vector*), розглядаються як ініціатори аномалії. Вони передаються оператору системи як потенційно небезпечні, із зазначенням вузла *host*, поведінку якого визнано відхиленням від норми.

Важливим етапом є оцінка критичності аномалії. Для цього використовується ієрархічна кластеризація: критичність зміни класу вузла визначається за кофенетичною відстанню (довжиною шляху між вузлами у дереві ієрархії). Наприклад, якщо сервер отримав додаткового клієнта, це може бути відносно

малокритична подія, тоді як поява нового керуючого з’єднання з комутатором, яке раніше ніколи не спостерігалося, може вказувати на серйозну загрозу.

Якщо ж розбіжностей між попередніми та новими мітками не зафіксовано:

$$(host_i.label = host_i.new_label), \quad (1.10)$$

вважається, що аномальної активності немає. У цьому випадку модель проходить повторне навчання з урахуванням оновлених векторів ознак вузлів *host.vector*. Таким чином, здійснюється перерозмітка багатовимірного простору з метою його уточнення та адаптації до змін мережевого середовища. Одночасно будується й оновлене ієрархічне дерево кластерів.

У результаті система переходить у режим очікування наступного набору потоків *Flows*. Такий підхід дозволяє не лише своєчасно виявляти аномалії, а й постійно адаптувати модель до нових умов, підтримуючи актуальність та надійність моніторингу.

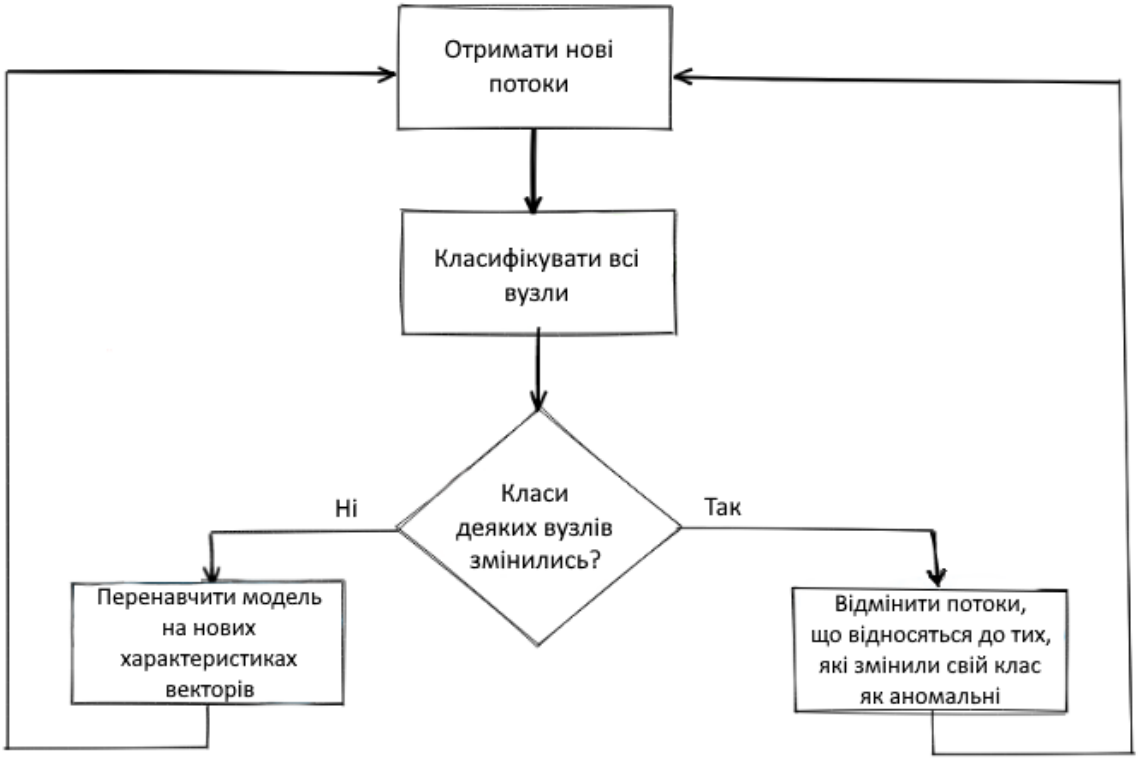


Рис. 1.5 Виявлення аномальних потоків

Запропонована формальна модель забезпечує можливість не лише визначати класифікаційні ознаки мережевих вузлів, але й виявляти відхилення у їхній поведінці, що інтерпретуються як аномалії.

1.3 Реалізація моделі детекції

Для формування потоків із сирого мережевого трафіку було використано модуль DPI (Deep Packet Inspection) програмного комплексу *CyberLympha DATAPK*. Завдяки цьому стало можливим виконати детальний аналіз переданих пакетів, виокремити їх за протокольними ознаками та перетворити у стандартизовані мережеві потоки. Наступним етапом було створення характеристичних векторів вузлів – цей процес здійснювався за допомогою спеціально розробленого скрипта мовою Python, що дозволило автоматизувати виділення ключових параметрів потоків та уніфікувати дані для подальшої обробки.

Задача вибору оптимальної архітектури моделей вирішувалася із застосуванням інструментів автоматизованого машинного навчання (AutoML). Для цього було використано бібліотеку TPOT (Tree-based Pipeline Optimization Tool), реалізовану у вигляді модуля для Python. Даний підхід дав змогу автоматично проводити відбір і налаштування комбінацій моделей та їхніх гіперпараметрів, що значно скорочує час розробки та підвищує якість отриманої моделі у порівнянні з ручним підбором.

У результаті оптимізації було побудовано ансамбль моделей, організований у два блоки, об’єднані методом стекінгу (stacking). Така архітектура передбачає ієрархічне поєднання моделей, коли результати роботи алгоритмів нижнього рівня виступають додатковими ознаками для алгоритмів верхнього рівня, підвищуючи тим самим загальну точність передбачення.

Перший блок ансамблю також являє собою ансамбль моделей К-найближчих сусідів (KNN), об’єднаних за допомогою стекінгу. Завдяки такій організації, відповіді окремих моделей нижчого рівня (KNN-класифікаторів з різними налаштуваннями) не лише враховуються безпосередньо, але й перетворюються на

додаткові характеристики для моделей вищого рівня. Це забезпечує більш стійку класифікацію, дозволяє зменшити ризик перенавчання та підвищує здатність ансамблю працювати з різнорідними даними.

Таким чином, поєднання технології глибинного аналізу пакетів (DPI), інструментів AutoML та ансамблевих методів класифікації створює гнучку й ефективну систему для розв’язання завдання ідентифікації вузлів і виявлення аномалій у мережевому трафіку.

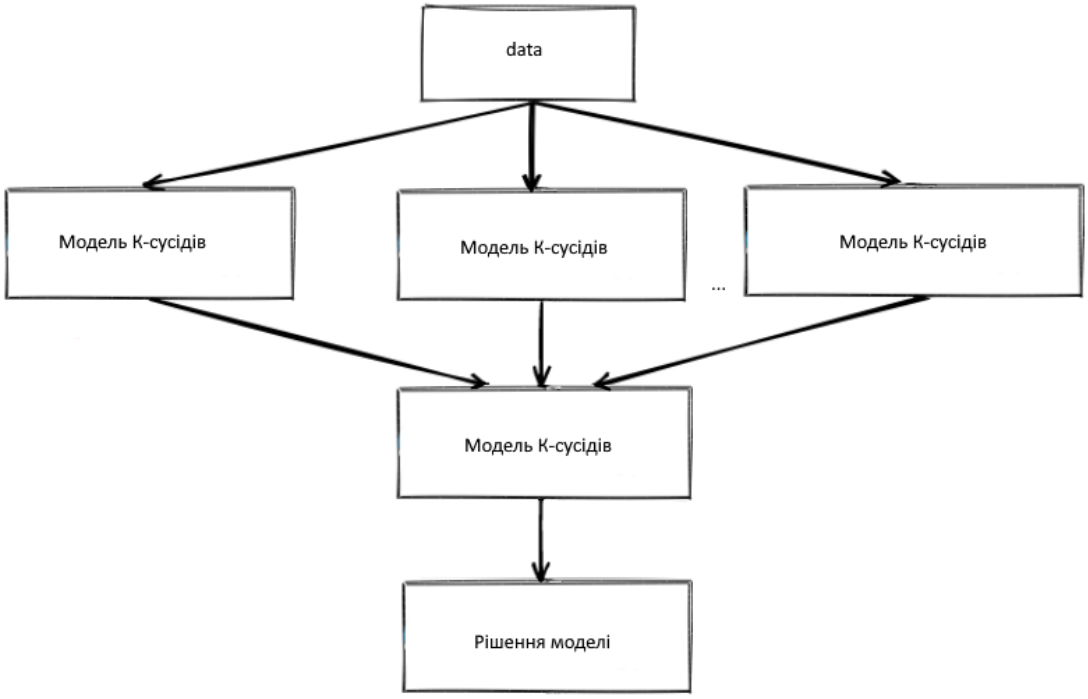


Рис. 1.6 Перший блок моделі

Другий блок – модель байєсівського класифікатора з мінімальним згладжуванням.

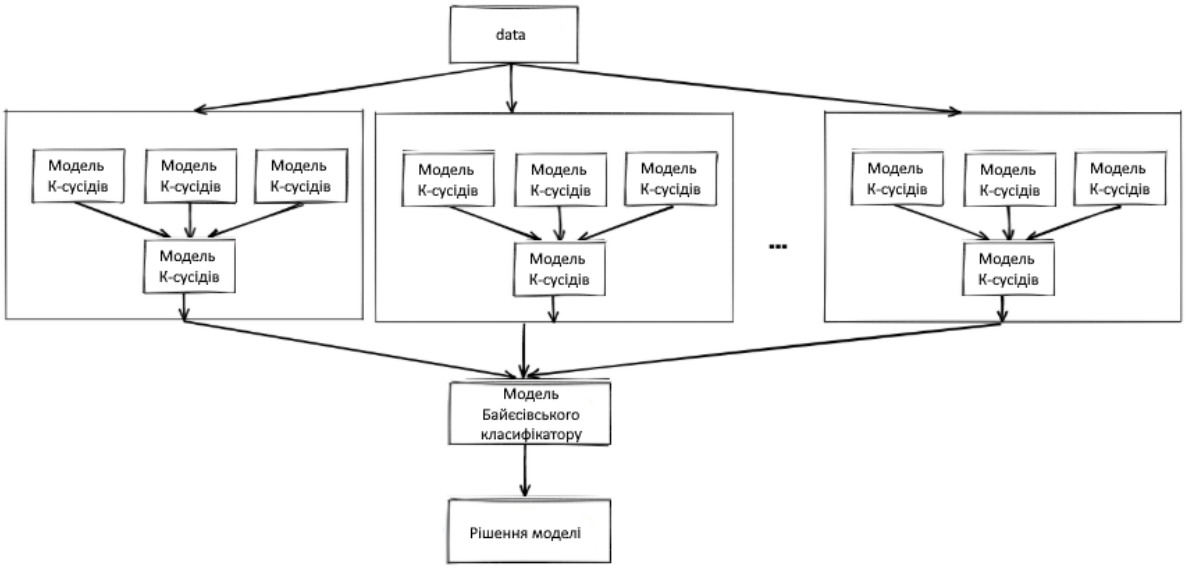


Рис. 1.7 Підсумкова модель

На даному етапі дослідження обсяг зібраних даних є обмеженим, що не дозволяє здійснити повну й об’єктивну оцінку ефективності роботи розроблених модулів. Проте отримані результати попередніх випробувань свідчать про значний потенціал запропонованого підходу. У майбутньому, коли система перейде у фазу дослідної або промислової експлуатації та з’являться більші обсяги репрезентативних даних, можна буде представити вичерпні статистичні показники й аргументовано підтвердити ефективність розробленого рішення.

Під час проведення експериментального тестування на незалежному наборі даних, який принципово відрізнявся від тренувальної вибірки та жодним чином не використовувався на етапі навчання моделі, було отримано значення F1-міри на рівні близько 80 %. Такий результат уже можна вважати задовільним для початкової стадії впровадження системи, адже він свідчить про здатність розробленої моделі адекватно розпізнавати та класифікувати об’єкти навіть за умов обмеженого обсягу вхідної інформації та потенційної різнорідності даних. Це є важливим підтвердженням життєздатності підходу, оскільки модель не «запам’ятала» специфіку тренувальних даних, а продемонструвала реальну здатність до узагальнення.

Окремої уваги заслуговують запропоновані вдосконалення, спрямовані на підвищення точності системи у випадках, коли рівень впевненості моделі у

прийнятому рішенні є недостатнім. Традиційно класифікатор повертає лише один результат, що в умовах невизначеності може призвести до серйозних помилок. Запропонований підхід передбачає, що у таких випадках користувачеві пропонуються не один, а два найбільш імовірні класи. Така модифікація дозволяє істотно знизити ризик хибної класифікації, надати оператору додаткову інформацію для прийняття остаточного рішення та підвищити гнучкість роботи системи у динамічних середовищах.

Результати тестування довели, що використання подібного «режиму невизначеності» суттєво впливає на якість класифікації: показник F1-міри зріс до рівня 95–98 %, що можна вважати надзвичайно високим показником для задач такого типу. При цьому кількість випадків, коли модель переходила у зазначений режим роботи, виявилася відносно невеликою у порівнянні з обсягом усієї вибірки. Це є надзвичайно важливим, адже воно підтверджує, що впровадження додаткового механізму не створює суттєвого навантаження на оператора, не потребує значних додаткових ресурсів та не знижує загальну продуктивність системи. Навпаки, подібний підхід формує додатковий рівень захисту від хибнопозитивних результатів, що підвищує довіру до системи з боку кінцевих користувачів.

Таким чином, навіть у рамках відносно обмежених експериментальних умов розроблений метод продемонстрував високу ефективність, універсальність і потенціал до масштабування. Впровадження механізму альтернативної пропозиції класів у випадках невизначеності не лише суттєво підвищує якість класифікації, але й робить систему більш стійкою до впливу непередбачуваних факторів та складних сценаріїв, які можуть виникати у реальному середовищі. Це, у свою чергу, відкриває широкі перспективи для подальшого вдосконалення моделі, інтеграції додаткових методів оптимізації та практичного застосування у задачах інтелектуального моніторингу, аналізу та захисту телекомунікаційної інфраструктури від різноманітних загроз.

2 ДОСЛІДЖЕННЯ РІШЕНЬ У СФЕРІ СВВ

2.1 Типи систем виявлення вторгнень

Традиційно системи виявлення вторгнень (СВВ) прийнято поділяти на три основні категорії. По-перше, це системи мережевого рівня (Network-based IDS), які аналізують трафік, що проходить через маршрутизатори чи інші вузлові точки, та виявляють аномалії або спроби атак безпосередньо у транспортних потоках. По-друге, існують системи рівня хоста (Host-based IDS), що зосереджуються на окремих машинах і здійснюють моніторинг локальних подій – від аналізу системних журналів і перевірки цілісності файлів до виявлення підозрілої мережевої активності. По-третє, до цієї групи належать системи, засновані на оцінюванні вразливостей (Vulnerability-assessment systems), які дозволяють визначати слабкі місця програмного забезпечення та конфігурацій, аби попередити потенційні атаки.

Втім, таке трирівневе розмежування можна розширити. Якщо розглядати системи не лише з позицій рівня функціонування, а й з точки зору їхніх завдань та принципів роботи, то можна виділити кілька універсальних функцій. Будь-яка система подібного класу фактично має відповідати на два фундаментальні питання: чи існує проблема в аналізованому середовищі та який її характер? Відповідь на це запитання формується на основі отриманих даних та способів їх інтерпретації.

Звідси випливає, що функціонал СВВ можна узагальнити у вигляді трьох основних завдань:

1. **Збирання даних** – забезпечення збору всієї релевантної інформації, яка відображає діяльність користувачів, додатків чи мережевих пристроїв.
2. **Інтерпретація даних** – застосування методів аналізу (сигнатурного, поведінкового, статистичного або гібридного) з метою визначення відхилень від норми.
3. **Представлення результатів** – формування звітів, сповіщень чи сигналів тривоги, зрозумілих оператору або інтегрованих у системи реагування.

Важливим є те, що всі системи такого класу можна класифікувати ще за кількома додатковими ознаками:

- **Тип зібраних даних** (мережеві пакети, системні журнали, метрики продуктивності, поведінкові показники тощо).
- **Метод отримання даних** (пасивне спостереження, активне опитування, аналіз копій трафіку, використання агентів на хостах).
- **Метод інтерпретації** (сигнатурний аналіз, статистичні методи, машинне навчання, евристичні алгоритми).
- **Метод представлення результату** (графічні інтерфейси, автоматизовані звіти, інтеграція з іншими системами безпеки).

Окремо варто виділити так звану **позасистемну характеристику** – тип реакції на отриманий результат. Тут можна виокремити два підходи:

- **Інформаційний режим**, коли система лише інформує адміністратора чи відповідальних осіб про знайдені загрози.
- **Активний режим**, коли система сама ініціює дії у відповідь, наприклад блокує підозрілий IP-адрес, відключає з’єднання або змінює параметри доступу.

За цією ознакою системи поділяють на IDS (Intrusion Detection System) та IPS (Intrusion Prevention System). Проте цей поділ є умовним: у багатьох випадках IDS та IPS можна розглядати як дві складові єдиної комплексної системи, де IDS виконує «розвідувальну» функцію (виявлення та попередження), а IPS – «силову» функцію (активне реагування та блокування). Таким чином, будь-яка IDS, за наявності відповідних модулів, може бути інтегрована у склад IPS.

Надалі в дипломній роботі основна увага буде приділена саме питанням архітектури, методам збору та аналізу даних, тоді як дихотомія IDS/IPS розглядатиметься лише як допоміжна, оскільки вона не змінює суті загальних принципів побудови систем виявлення вторгнень.

Класи зібраних даних у системах виявлення вторгнень

При розробці систем виявлення вторгнень надзвичайно важливим аспектом є правильна класифікація та розмежування типів даних, що збираються для

подальшого аналізу. Від того, наскільки повноцінно й структуровано буде сформовано набір даних, безпосередньо залежить якість виявлення атак, швидкість реакції системи та рівень адаптивності до нових загроз. Традиційні підходи у цій сфері зазвичай обмежуються поділом даних лише на два основних класи: інформація, що характеризує окремий вузол, та дані, які відображають стан усієї мережі. Проте сучасні дослідження та практика побудови більш гнучких моделей доводять доцільність виокремлення ще й третього класу — гібридного. Він дозволяє розглядати систему у багатовимірному «конфігураційному просторі», де комбінуються різні методики збору й аналізу інформації, що суттєво підвищує ефективність роботи.

Відповідно, можна виділити такі три основні класи зібраних даних:

1. **Дані, що збираються про окремий вузол мережі.**
2. **Дані, що збираються про всю мережу загалом.**
3. **Гібридні системи, які поєднують перші два підходи.**

Дані про окремий вузол

Цей тип охоплює інформацію, яка безпосередньо характеризує поведінку одного конкретного пристрою, а також частково відображає його взаємодію з іншими вузлами мережі. Основна мета аналізу таких даних полягає у відповіді на ключове запитання: «Чи спрямована атака безпосередньо на даний хост?»

Збір інформації зазвичай здійснюється локально — безпосередньо на самому вузлі. Це можуть бути різноманітні програмні агенти, що відстежують активність системи та формують докладні журнали подій. Водночас існують і методи віддаленого отримання відомостей: наприклад, мережеві сканери здатні виявляти відкриті порти пристрою ззовні, навіть без можливості запуску на ньому коду чи встановлення додаткового програмного забезпечення.

До типових прикладів даних цього класу належать:

- **мережева активність вузла** — кількість і напрям з’єднань, обсяг переданого й отриманого трафіку, частота використання різних протоколів;
- **налаштування мережі** — конфігурація інтерфейсів, таблиці маршрутизації, параметри безпеки, політики доступу;

- **дані про файли** — структури директорій, контрольні суми, метадані, операції з файлами, виявлення підозрілих змін;
- **дані про процеси** — список запущених служб і програм, рівень споживання ресурсів, наявність невідомих або несанкціонованих додатків.

Особливу увагу слід звернути на різноманітність функціонального призначення вузлів. Наприклад, робочі станції користувачів, що не надають мережевих сервісів, генерують один тип даних, тоді як сервери, що виконують критично важливі функції (файлові, поштові чи баз даних), створюють зовсім інший профіль активності. Окремо виділяється категорія спеціально створених вузлів-пасток (*honeypot*). Їхня ключова особливість полягає в тому, що вони навмисно робляться вразливими для атак. Це дозволяє дослідникам збирати цінну інформацію про методи зловмисників, виявляти джерела атакуючої активності та навіть використовувати отримані дані для подальшого навчання алгоритмів машинного інтелекту. У таких випадках вважається, що будь-яке звернення до вузла є потенційною атакою, а отже, вся зібрана інформація автоматично набуває значення для аналізу.

Таким чином, дані про окремий вузол формують фундамент для побудови моделей виявлення вторгнень на рівні хостів. Вони дозволяють своєчасно виявляти підозрілу активність, контролювати цілісність файлової системи, аналізувати налаштування та виявляти потенційно небезпечні процеси, що у комплексі значно підвищує рівень кіберзахисту мережевої інфраструктури.

Дані про мережу

Другий клас даних охоплює інформацію, яка дозволяє формувати цілісне уявлення про функціонування всієї мережевої інфраструктури в цілому. На відміну від аналізу окремого вузла, у цьому випадку головна увага приділяється глобальній динаміці взаємодії між усіма вузлами, трафіку, що циркулює між ними, та закономірностям у поведінці мережевих потоків. Такий підхід дозволяє отримати комплексне бачення стану мережі, виявляти підозрілі аномалії, які можуть залишатися непоміченими на рівні окремих хостів, і своєчасно реагувати на загрози, що мають масштабний або розподілений характер.

Повний збір усього мережевого трафіку у більшості випадків вважається практично нереалізованим завданням через надмірні вимоги до ресурсів. Для його зберігання та подальшої обробки потрібні колосальні обчислювальні потужності, великі обсяги оперативної пам’яті й довготривале сховище з високою пропускною здатністю. Саме тому на практиці часто використовуються методи відбору даних, статистичної агрегації або ж аналізу лише окремих фрагментів потоків, що дозволяє досягти балансу між точністю оцінки та витратами ресурсів.

Традиційно застосовується припущення, що зловмисник розташований за межами внутрішньої мережі й обов’язково здійснює взаємодію із зовнішнім середовищем. Це спрощує процес моніторингу, адже значна частина аналізу фокусується на вхідному та вихідному трафіку. Однак історія кіберзагроз довела, що такий підхід не завжди є виправданим. Численні реальні інциденти свідчать про те, що навіть ізольовані системи з фізичним розривом (*air-gap*) не гарантують абсолютного захисту. Відомі випадки, коли зловмисники успішно долали обмеження, використовуючи методи побічних каналів, заражені зовнішні носії чи навіть інфільтрацію через апаратні закладки.

З технічного погляду найчастіше аналіз трафіку здійснюється на рівні маршрутизаторів або комутаторів верхнього рівня. Для цього використовується спеціальний SPAN-порт (Switched Port Analyzer), який дозволяє створювати копію всього трафіку та передавати її до системи виявлення вторгнень для подальшої обробки. Такий підхід забезпечує відносно централізоване спостереження та зручність адміністрування, але вимагає ретельної оптимізації, щоб не допустити перевантаження обладнання.

У низці випадків можливим є і збір даних безпосередньо на вузлі, де встановлена система виявлення вторгнень (СВВ). Це створює додатковий рівень локального контролю й дозволяє оперативно виявляти підозрілі події, перш ніж вони встигнуть поширитися мережею. Крім того, у деяких архітектурах передбачається розподілений збір даних на різних сегментах мережі, що забезпечує більшу гнучкість і стійкість системи.

Ще один підхід полягає у використанні режиму перехоплення всього трафіку мережевими адаптерами вузлів. Проте він вважається надмірно ресурсомістким, оскільки вимагає, щоб мережеві карти працювали у нефункціональному для них штатному режимі, що призводить до значних втрат продуктивності. Єдиним випадком, коли такий метод може бути виправданим, є розподілений аналіз потоків, де збір даних здійснюється одночасно з багатьох точок для отримання повнішої картини. Навіть тоді цей підхід використовується досить рідко та лише з великими застереженнями, оскільки він створює ризик перенавантаження інфраструктури.

Гібридні системи

Третій клас об’єднує переваги перших двох підходів, створюючи більш універсальний та комплексний механізм аналізу. Гібридні системи поєднують дані, зібрані як на рівні окремих вузлів, так і на рівні мережі загалом, що дозволяє отримати багатогранне уявлення про роботу всієї інфраструктури. Такий підхід забезпечує не лише синергію різних джерел інформації, а й формує багаторівневу модель спостереження, яка поєднує локальний і глобальний контекст.

Наприклад, аналіз мережевого трафіку на маршрутизаторі здатний виявити аномалії у взаємодії між вузлами — раптовий ріст кількості з’єднань, незвичне використання портів або ж появу підозрілих потоків. Проте лише завдяки додатковому моніторингу безпосередньо на самому хості можна зрозуміти, які саме процеси, служби чи файли стали джерелом цієї активності. Це створює умови для побудови логічного ланцюжка «мережевий сигнал — локальна причина», що суттєво підвищує точність і швидкість реагування на загрози.

Гібридні рішення також дозволяють реалізувати кореляцію подій різного рівня: дані з мережевого сегмента можна зіставляти з журналами подій операційних систем, системними логами або інформацією від спеціалізованих сенсорів. Це дає змогу відокремлювати хибнопозитивні спрацьовування від реальних атак, зменшуючи кількість «шуму» у звітах та підвищуючи ефективність роботи аналітиків безпеки.

Особливо важливою є роль гібридних систем у великих корпоративних мережах, де поєднуються сотні або навіть тисячі вузлів із різнорідними сервісами

— від робочих станцій користувачів і мобільних пристроїв до серверів баз даних, систем віртуалізації, хмарних рішень і спеціалізованого обладнання. У таких інфраструктурах простого мережевого аналізу може бути недостатньо, оскільки загроза здатна маскуватися під легітимну активність. Локальний же моніторинг окремих вузлів без урахування мережевого контексту також може дати неповну картину. Саме тому гібридні системи забезпечують найбільш збалансований підхід, поєднуючи «широкий кут зору» з можливістю детального занурення у конкретні інциденти.

Ще однією суттєвою перевагою гібридного підходу є його здатність до масштабування та адаптації. У разі зростання мережі чи підключення нових сервісів система не потребує повного переосмислення архітектури: достатньо додати нові точки збору даних і інтегрувати їх у вже існуючий аналітичний контур. Це робить такі рішення актуальними для організацій із підвищеними вимогами до безпеки, зокрема у сфері фінансових послуг, енергетики, телекомунікацій та державного сектору.

Класи методів отримання даних у системах виявлення вторгнень

Для ефективного функціонування систем виявлення вторгнень (СВВ) важливим є не лише тип зібраних даних, але й метод їхнього отримання. Саме спосіб збору інформації визначає ступінь впливу системи на мережу, рівень її інтеграції в інфраструктуру та потенційні ризики. Залежно від цього, методи отримання даних поділяються на три основні класи:

1. **Пасивні методи.** Система працює у режимі спостереження, не впливаючи безпосередньо на роботу мережі чи окремих хостів. Усі висновки робляться лише на основі аналізу перехопленого трафіку та зареєстрованих подій.
2. **Активні методи.** Система ініціює вплив на мережу або вузли з метою викликати реакцію, яка дозволяє виявити вразливості. Це може бути тестування відомими наборами атак або застосування методів генерації випадкових аномальних запитів (*fuzzing*).
3. **Змішані методи.** Поєднують риси обох підходів, що дозволяє балансувати між точністю виявлення та рівнем навантаження на мережу.

Пасивне виявлення

Пасивні методи передбачають, що система лише відстежує ситуацію в мережі, не змінюючи її. Це найбільш поширений підхід, який реалізується у більшості сучасних IDS. Він дозволяє забезпечити безперервний моніторинг із мінімальним впливом на функціонування середовища.

Системи рівня хоста зазвичай також застосовують цей підхід. Наприклад, замість того, щоб навмисно видаляти системний файл для перевірки реакції операційної системи, система просто аналізує права доступу до нього та звіряє їх зі зразками у базі даних. У разі виявлення невідповідності генерується попередження.

Таким чином, пасивні методи характеризуються безпечністю та стабільністю, але їхнім недоліком є обмежена здатність виявляти потенційні вразливості, які ще не проявилися у вигляді аномалій.

Активний пошук вразливостей

Активні методи передбачають провокування реакцій у системі шляхом спеціально сформованих дій. Це можуть бути як відомі тестові впливи, так і випадково згенеровані запити (наприклад, у рамках *fuzzy testing*).

Реакція системи на такі впливи аналізується й порівнюється з базою знань або з типовими моделями поведінки. Цей клас методів є характерним для сканерів уразливостей.

Існує два основні підходи до інтерпретації результатів:

- **Аналіз відповідей за базою сигнатур.** Наприклад, характерна реакція на SQL-ін’єкцію може бути ознакою відомої вразливості.
- **Поведенковий аналіз.** Оцінюється, як змінюється поведінка системи після впливу. Якщо, наприклад, надсилання некоректного IP-пакета призводить до збою у роботі сервера і він перестає відповідати на запити, це свідчить про наявність критичної вразливості.

Аналогічно, можна зафіксувати і непрямі аномалії. Наприклад, якщо після відправлення ICMP-пакета нетипового розміру (777 байт, заповненого значенням 0xDEADBEEF) різко зростає активність у мережі, а згодом падає, це розцінюється

як аномалія. У нормальних умовах мережеве навантаження при цьому має залишатися стабільним.

Переваги активних методів

- **Превентивність.** Дають можливість виявити вразливості ще до того, як відбудеться реальна атака.
- **Глибина аналізу.** Мережа перевіряється за сценаріями, максимально наближеними до дій зловмисників.
- **Підвищена ймовірність виявлення слабких місць.** Система здатна визначити навіть ті вразливості, які ще не призвели до інцидентів.

Недоліки активних методів

- **Додаткове навантаження на мережу.** Створюваний тестовий трафік може знижувати продуктивність.
- **Ризик ненавмисних інцидентів.** У процесі сканування можливе виникнення відмов у роботі сервісів, наприклад DoS.
- **Залежність від бази знань.** Сканери уразливостей часто ґрунтуються на відомих шаблонах, які швидко застарівають.
- **Ілюзія безпеки.** Відсутність виявлених вразливостей не гарантує абсолютного захисту, оскільки ризик цільових атак усе одно зберігається.

Змішані методи

Змішані підходи у системах виявлення вторгнень являють собою поєднання двох базових стратегій — пасивного моніторингу та активного тестування. Їхня головна мета полягає у створенні такого балансу, який дозволяє з одного боку уникати надмірного навантаження на мережеву інфраструктуру, а з іншого — зберігати здатність до своєчасного й навіть превентивного виявлення потенційних загроз. Завдяки такій інтеграції формується гнучка система, здатна адаптуватися до різних сценаріїв і забезпечувати багаторівневий захист.

Пасивний моніторинг у цьому контексті виступає фундаментом, що забезпечує безперервність контролю, стабільність роботи системи та мінімальний вплив на продуктивність мережі. Він дозволяє спостерігати за реальною поведінкою користувачів, процесів і сервісів без втручання у трафік. Проте

обмеженість такого підходу проявляється у тому, що він не завжди дає змогу виявити приховані або ще не активовані вразливості.

Активне тестування, навпаки, спрямоване на імітацію або штучне створення потенційних атак із метою виявлення слабких місць ще до того, як ними скористається зловмисник. До активних методів належать сканування портів, моделювання вторгнень, ін’єкція контрольованих запитів чи використання спеціалізованих фреймворків для тестування на проникнення. Водночас такі методи супроводжуються низкою ризиків: підвищеним навантаженням на систему, можливістю хибних спрацювань, ризиком порушення нормальної роботи сервісів або навіть створенням додаткових векторів атак.

Змішані методи інтегрують обидва підходи, дозволяючи компенсувати їхні недоліки. У реальних умовах це може виглядати так: система постійно працює у режимі пасивного моніторингу, але при виявленні підозрілої активності автоматично запускає активні перевірки конкретних вузлів або сегментів мережі. Така логіка дозволяє зменшити кількість непотрібних перевірок, мінімізувати ризики від надмірного використання активних інструментів і водночас гарантувати своєчасне виявлення критичних вразливостей.

Особливу актуальність змішані методи мають у великих корпоративних мережах та інфраструктурах із підвищеними вимогами до безпеки, де присутня значна кількість вузлів, гетерогенність сервісів і складна структура взаємодій. У таких середовищах надмірно агресивне активне тестування може дестабілізувати роботу систем, тоді як лише пасивний моніторинг може виявитися недостатнім. Саме тому змішані системи створюють оптимальний компроміс, дозволяючи зберегти стабільність та безпечність, не втрачаючи глибини аналізу і здатності діяти на випередження.

Таким чином, змішані методи можна розглядати як найбільш збалансований варіант у сучасних системах виявлення вторгнень. Вони поєднують переваги пасивних рішень, які гарантують стабільність і невидимість, та активних, які забезпечують превентивне виявлення загроз. Завдяки цьому створюється гнучкий і

ефективний механізм, що дозволяє підтримувати оптимальний рівень захисту навіть у складних і динамічних мережевих середовищах.

Методи представлення результатів у системах виявлення вторгнень

Одним із важливих аспектів роботи систем виявлення вторгнень (СВВ) є форма подання результатів аналізу. Від того, наскільки зрозуміло та інформативно система представляє дані, залежить швидкість реакції оператора та ефективність усієї системи захисту. Аналіз існуючих підходів дозволяє виокремити кілька основних методів.

1. Двокласовий метод

Цей метод передбачає бінарну відповідь на запитання: «Чи є проблеми?». Результат формується у вигляді відповіді *«так»* або *«ні»*.

Такий підхід є типовим для академічних досліджень, де задача спрощується для формального порівняння алгоритмів або тестування нових методів. Разом із тим, він може використовуватися і в реальних системах, наприклад у вигляді датчиків чи модулів, які виконують роль базового рівня моніторингу. Їхнє завдання – швидко зафіксувати факт відхилення й передати інформацію для подальшого опрацювання.

Однак головним недоліком цього підходу є його обмеженість: оператор отримує лише констатацію факту, але не має відомостей про природу чи характер виявленої проблеми.

2. Багатокласовий метод

На відміну від попереднього, цей метод дозволяє дати відповідь на більш деталізоване запитання: «Які саме проблеми зафіксовані?». Якщо система не виявляє відхилень, результатом буде відповідь *«жодні»*.

Багатокласовий підхід є більш інформативним, адже дозволяє ідентифікувати конкретний тип загрози — наприклад, атаку типу *SQL Injection*, *DoS*, *Brute Force* тощо. Це дає змогу не лише зафіксувати факт аномалії, але й оцінити її потенційну небезпеку та швидко застосувати цільові заходи реагування.

Тим не менш, цей метод вимагає більш складних моделей, розширених баз даних сигнатур та точніших алгоритмів інтерпретації, що підвищує його обчислювальну вартість.

3. Ймовірнісний підхід

У сучасних системах, особливо тих, що побудовані на нейронних мережах та алгоритмах машинного навчання, результати рідко бувають однозначними. Насправді модель завжди формує **розподіл ймовірностей** для кожного можливого класу. Це означає, що на виході ми отримуємо не просту відповідь «так/ні», а числові значення, які відображають ступінь упевненості системи у кожному варіанті.

Проте в більшості практичних реалізацій ця багатозначна інформація редукується: вибирається клас із найбільшою ймовірністю, який вважається «правильним», тоді як решта варіантів ігноруються. Таким чином, система зводить ймовірнісний простір до двійкового результату (1 для найімовірнішого класу та 0 для інших).

Такий підхід є зручним для реалізації, проте він значно зменшує потенціал системи, оскільки відкидає додаткові дані, які можуть бути корисними для більш глибокого аналізу. Наприклад, якщо система оцінює атаку *SQL Injection* із ймовірністю 51 %, а атаку *XSS* із ймовірністю 47 %, то редукція залишає лише перший варіант, хоча насправді обидві загрози майже рівноймовірні.

4. Ймовірнісно-багатокласовий метод (перспективний)

Одним із перспективних напрямів розвитку є використання ймовірнісного підходу без редукції. У такому випадку результат представляється у вигляді розподілу ймовірностей для всіх можливих класів. Це дозволяє не лише врахувати найбільш ймовірний сценарій, але й оцінити ступінь невизначеності системи.

Наприклад, замість відповіді «атака типу DoS» система може сформувати вихід:

- DoS — 70 %;
- SQL Injection — 20 %;
- Normal traffic — 10 %.

Такий формат є значно кориснішим для оператора, оскільки він може застосувати багаторівневу стратегію реагування: підтвердити найбільш ймовірну атаку, водночас підготувавши додаткові захисні заходи проти інших варіантів.

Крім того, збереження ймовірнісного розподілу дозволяє інтегрувати результати у системи вищого рівня, що працюють із ризик-орієнтованими моделями або сценаріями прогнозування.

2.2 Типи IDS

На даному етапі параметр «Метод представлення результату» не має ключового значення, оскільки він був запроваджений лише як допоміжний критерій для розмежування детекторів. Для спрощення подальшого аналізу усі системи доцільно звести до меншої кількості типів, які узагальнюють найхарактерніші варіанти побудови.

- **Типи 0, 1, 2.** До цієї групи належать IDS рівня вузла, що реалізують три підходи: сигнатурний (виявлення відомих зразків атак), поведінковий (виявлення аномальної активності) та гібридний (поєднання перших двох). Таких систем існує значна кількість, і вони представлені як академічними розробками, так і промисловими рішеннями. До цього ж класу логічно віднести й антивірусні програми, оскільки вони реалізують схожі механізми.
- **Типи 3, 4, 5.** Це **сканери вразливостей рівня вузла**, які також можуть базуватися на сигнатурному, аномалійному чи гібридному аналізі. Їхня основна функція полягає у виявленні невиправлених вразливостей у системі. Часто такий функціонал інтегрується в антивірусні пакети, створюючи універсальні рішення безпеки. З певними застереженнями до цієї групи можна віднести інструмент AVZ, хоча у подальшому цей тип розглядатися окремо не буде.
- **Типи 6, 7, 8.** Сюди належать гібридні IDS рівня вузла, що поєднують можливість пасивного моніторингу та активного сканування. Вони

дозволяють як фіксувати атаки у реальному часі, так і активно перевіряти систему на наявність слабких місць.

- **Типи 9, 10, 11.** Це **IDS мережевого рівня, які можуть працювати у сигнатурному режимі, аналізувати аномалії у трафіку** або комбінувати обидва підходи. Такі системи забезпечують глобальний моніторинг мережі та дають змогу виявляти атаки незалежно від окремих хостів.
- **Типи 12, 13, 14.** До цієї групи належать сканери вразливостей **мережевого рівня**. Вони можуть ідентифікувати відомі шаблони атак, виявляти нетипові реакції систем на специфічні запити або працювати у змішаному режимі. Такі рішення здатні виявляти потенційні точки проникнення в мережу ще до того, як відбудеться реальна атака.
- **Типи 15, 16, 17.** Це **гібридні IDS мережевого рівня, які дозволяють поєднувати активне сканування та пасивний аналіз трафіку**. Вони орієнтовані на комплексний захист мережевого середовища, забезпечуючи як виявлення відомих загроз, так і фіксацію нових аномалій. Зазвичай такі системи існують не як окремі продукти, а інтегруються у більш масштабні рішення, що охоплюють декілька рівнів захисту одночасно.
- **Тип 26.** Окремо виділяється гібридна система, побудована на основі **агентів**. У цьому випадку кожен агент може реалізовувати один із перелічених вище типів. Подібна архітектура забезпечує масштабованість та гнучкість: у мережу можна розгортати довільну кількість агентів, які спеціалізуються на конкретних функціях. Саме тому проміжні типи (18–25) зазвичай не розглядаються окремо, а узагальнюються у цьому інтегрованому класі.

Honeypot як допоміжний інструмент

Окремої уваги заслуговує спеціальний клас систем, які формально не належать до класичних IDS, але широко застосовуються як допоміжні рішення у сфері кіберзахисту. Йдеться про honeypot — так звані вузли-пастки, які навмисно створюються вразливими для атакувальників. Основна ідея їх використання полягає у тому, щоб приваблювати зловмисників, фіксувати всі їхні дії та детально аналізувати застосовані методики атак. На відміну від реальних вузлів мережі,

honeypot не виконує критично важливих функцій, а існує виключно з дослідницькою та захисною метою.

Зібрана з таких систем інформація має подвійне практичне значення. По-перше, вона використовується для поповнення бази сигнатур у традиційних системах виявлення вторгнень, що дозволяє швидше реагувати на вже відомі атаки. По-друге, дані про нові або рідкісні типи атак застосовуються у системах поведінкового аналізу чи навіть для навчання нейронних мереж, які покликані виявляти аномальні дії за непрямими ознаками. Таким чином, honeypot виступає не лише як «пастка» для атакувальників, а і як інструмент накопичення знань для вдосконалення всієї системи кіберзахисту.

Крім того, honeypot дозволяє не просто виявляти атакуючі вузли, а й досліджувати їхню поведінку у режимі реального часу. Це робить його надзвичайно цінним у сфері кіберрозвідки, де важливо зрозуміти мотивацію, тактики та інструменти зловмисників. Зокрема, honeypot може допомогти ідентифікувати, які саме вразливості найчастіше експлуатуються, які протоколи зловмисники намагаються використати для вторгнення, а також які шкідливі програми чи сценарії застосовуються для подальшого закріплення у системі.

У практиці виділяють кілька типів honeypot: від простих, які імітують базові сервіси (наприклад, відкритий порт SSH чи HTTP), до складних honeynet — мережевих середовищ, які симулюють цілу інфраструктуру з декількома взаємодіючими вузлами. Це дозволяє моделювати більш реалістичні сценарії атак і отримувати глибші аналітичні дані.

Таким чином, використання honeypot суттєво розширює можливості систем виявлення вторгнень. Вони забезпечують додатковий рівень захисту, знижують ризик несподіваних атак і водночас створюють потужну базу знань для дослідників та аналітиків. У поєднанні з класичними сигнатурними IDS, сучасними гібридними архітектурами на базі агентів і системами поведінкового аналізу honeypot формує частину комплексного підходу до захисту телекомунікаційних мереж.

Отже, запропонована класифікація методів і засобів дозволяє впорядкувати різноманітні підходи до побудови систем виявлення вторгнень і допоміжних

інструментів. Вона охоплює як традиційні рішення, так і сучасні інноваційні практики, створюючи основу для подальшого порівняльного аналізу їхньої ефективності. Це, у свою чергу, дає змогу обирати оптимальну модель захисту, що відповідає конкретним умовам функціонування мережевої інфраструктури та рівню актуальних кіберзагроз.

2.3 Діючі системи IDS

Snort

Snort є однією з найбільш відомих і класичних систем виявлення вторгнень мережевого рівня. Основний принцип її роботи полягає в аналізі мережевого трафіку на предмет відповідності правилам, що зберігаються у базі даних (по суті — сигнатурам). Це означає, що Snort здатна виявляти насамперед відомі атаки, для яких заздалегідь описані характерні ознаки.

Важливою перевагою Snort є її гнучкість та розширюваність. Створення власних модулів чи правил є відносно нескладним завданням, що й обумовило популярність цієї системи у наукових дослідженнях та практичних розробках. Багато комерційних рішень у сфері інформаційної безпеки побудовані саме на основі Snort, що підкреслює її надійність та перевірену часом архітектуру.

Хоча система першочергово орієнтована на сигнатурний аналіз, Snort може бути розширена додатковими модулями — евристичними, статистичними чи навіть нейромережевими. Наприклад, існує працездатний модуль статистичного виявлення аномалій, інтегрований у Snort, який дозволяє підвищити ефективність виявлення нових та невідомих загроз. Таким чином, Snort може виступати не лише як сигнатурна IDS, але і як основа для побудови гібридних рішень.

Suricata

Suricata — це ще одна потужна IDS мережевого рівня, яка в багатьох аспектах близька до Snort, але має низку важливих відмінностей і розширених можливостей.

Серед основних особливостей Suricata можна виділити:

- сумісність баз сигнатур зі Snort, що дозволяє без труднощів використовувати вже створені правила;
- підтримку аналізу прикладних протоколів, а не лише мережевого та транспортного рівнів, що суттєво підвищує точність виявлення загроз;
- можливість створення правил за допомогою мови Lua, що дає змогу гнучко адаптувати систему до специфічних умов експлуатації;
- здатність аналізувати не тільки окремі пакети чи сесії, а й повну взаємодію між вузлами, що дозволяє виявляти, наприклад, атаки перебору паролів;

наявність підсистеми IP reputation, яка призначає кожній IP-адресі показник «рівня довіри» або «репутації». Цей показник може змінюватися під час роботи системи залежно від поведінки вузла, що робить Suricata більш адаптивною до динамічних умов мережі.

Таким чином, Suricata поєднує у собі традиційні механізми сигнатурного аналізу з інноваційними підходами, орієнтованими на динамічну зміну характеристик мережі. Це робить її більш «інтелектуальною» у порівнянні зі Snort та здатною не лише виявляти відомі атаки, а й підлаштовуватися до нових умов.

Bro (нині Zeek)

Bro, нині відомий під назвою Zeek, — це не просто IDS мережевого рівня, а ціла платформа для створення спеціалізованих систем аналізу мережевого трафіку. Її особливістю є гібридний підхід, у якому поєднано можливості виявлення відомих загроз та аналізу аномалій.

Bro функціонує на транспортному, мережевому та прикладному рівнях, що дозволяє комплексно аналізувати трафік. Система має власну мову сценаріїв, за допомогою якої користувач може описувати політики аналізу подій та визначати правила реагування. Це робить Bro унікальною платформою, яка дає змогу адаптувати IDS під специфічні потреби конкретної мережі.

Можливості виявлення аномалій у Bro охоплюють як формальну перевірку коректності даних (наприклад, підозрілий TCP-пакет із усіма встановленими прапорцями), так і аналіз поведінки на основі політик. Таким чином, система здатна

виявляти як низькорівневі аномалії у пакетах, так і більш високорівневі порушення у логіці взаємодії вузлів.

Технічно Bro реалізовано на базі так званої подієвої машини. Замість прямого аналізу потоків пакетів, вони перетворюються у серію високорівневих подій (наприклад, HTTP-запит трансформується у подію http_request із параметрами). Далі ці події обробляються інтерпретатором політик, у якому визначені сценарії та обробники. На цьому рівні можливе накопичення статистики, збереження контексту та врахування часової динаміки, що забезпечує значно більш глибокий аналіз у порівнянні зі звичайними сигнатурними системами.

Унікальною рисою Bro є також його застосування для діагностики проблем у мережі. Тобто система може використовуватися не лише як IDS, але й як універсальний інструмент моніторингу та аналізу трафіку, що значно розширює сферу її практичного застосування.

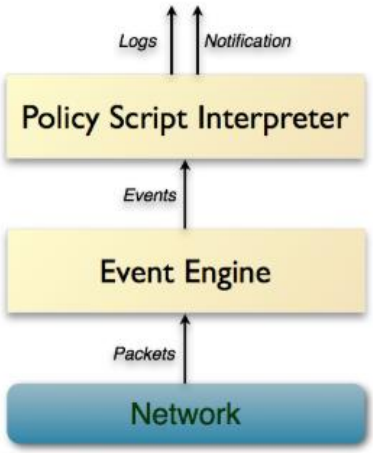


Рис. 2.1 Приклад роботи ядра Bro

3 ПРОЄКТУВАННЯ МОДУЛЯ СВВ

3.1 Можливі атаки

Для того щоб правильно визначити стратегію захисту інформаційних систем і мереж, необхідно чітко розуміти, від яких саме загроз потрібно оборонятися, де найчастіше проявляються атаки та які їхні характерні риси.

Зміщення векторів атак

Розглядати канальний рівень та нижчі рівні мережевої моделі OSI у межах даного дослідження доцільно лише частково, оскільки їхня специфіка значною мірою залежить від конкретного обладнання та середовища передачі. Протоколи цього рівня мають вузькоспеціалізований характер, вони часто реалізуються апаратно та не передбачають широкого спектра програмних механізмів аналізу. Тому більшість систем виявлення вторгнень (IDS) фактично не працюють на фізичному чи канальному рівнях. Винятком можна вважати лише окремі випадки, наприклад, виявлення ARP-spoofing у мережах Ethernet, коли зловмисник намагається підмінити ARP-таблиці для перехоплення чи модифікації трафіку. Інші атаки цього рівня, зокрема на протоколи типу PPP чи спеціалізовані канальні протоколи, застосовуються досить рідко, є локалізованими та досліджені ще недостатньо.

Проте, якщо аналізувати динаміку розвитку кіберзагроз упродовж останніх десятиліть, можна простежити чітку тенденцію до зміщення векторів атак. Починаючи з кінця 1990-х – початку 2000-х років основний фокус зловмисників поступово перемістився з мережевого та транспортного рівнів на прикладний рівень. Якщо на початкових етапах становлення Інтернету найбільш поширеними були атаки, спрямовані на низькорівневі протоколи (наприклад, ICMP-flood, SYN-flood, різноманітні спроби порушити роботу TCP/UDP-з’єднань чи викликати відмову в обслуговуванні), то з часом ці методи почали втрачати ефективність через розвиток механізмів захисту на рівні обладнання та програмних стеків.

Сьогодні найбільші ризики зосереджені саме на прикладному рівні, де функціонують сервіси, веб-додатки та бізнес-критичні системи. Це обумовлено кількома чинниками. По-перше, прикладний рівень є найбільш різноманітним та складним: тисячі різних протоколів і додатків відкривають широкі можливості для пошуку вразливостей. По-друге, саме тут зосереджені критично важливі дані користувачів і організацій — від банківських транзакцій до персональної інформації. По-третє, прикладний рівень безпосередньо взаємодіє з кінцевим користувачем, а отже, є доступним для зловмисників у глобальному масштабі.

Зміщення векторів атак від низькорівневих протоколів до прикладних сервісів можна розглядати як природну еволюцію кіберзагроз. Розвиток систем фільтрації, вдосконалення протоколів маршрутизації та впровадження захищених транспортних механізмів зменшили ефективність атак на мережевому й транспортному рівнях. Натомість швидке зростання кількості веб-додатків, поширення хмарних сервісів і цифровізація бізнесу відкрили новий фронт для атак. Саме тому сучасні IDS та інші системи захисту дедалі частіше зосереджуються на аналізі HTTP/HTTPS-трафіку, SQL-запитів, API-взаємодій і виявленні таких атак, як SQL-injection, XSS, CSRF, RCE та інші.

Класичні мережеві атаки

На початкових етапах становлення глобальних комп’ютерних мереж та Інтернету найбільш поширеними були атаки, спрямовані на транспортний і мережевий рівні моделі OSI. Вони використовували базові вразливості протоколів і простоту реалізації мережевих з’єднань у той час. Метою таких атак зазвичай було порушення стабільності роботи сервісів, виведення систем з ладу або створення умов для несанкціонованого доступу.

До найвідоміших прикладів класичних атак належать:

- **Ping of Death** — відправлення неправильно сформованих або надмірно великих ICMP-пакетів, що спричиняло збої у стеку TCP/IP старих операційних систем і могло повністю паралізувати роботу вузла.

- **IP Spoofing** — підробка IP-адреси джерела у пакеті з метою видавання себе за інший вузол. Такий підхід дозволяв обходити прості механізми автентифікації або запускати атаки типу DoS, приховуючи справжнє джерело.
- **SYN Flood** — перевантаження сервера великою кількістю запитів на встановлення з'єднання TCP, при цьому тристороннє рукостискання (*three-way handshake*) не завершувалося. Це призводило до виснаження ресурсів і відмови в обслуговуванні легальних користувачів.
- **ARP cache poisoning** (підміна ARP-таблиці) — маніпуляція протоколом ARP у локальних мережах, коли зловмисник змушував вузли асоціювати неправильну MAC-адресу з певною IP-адресою. У результаті ставало можливим перехоплення, модифікація чи блокування трафіку.

Ці атаки стали класикою кіберзагроз, адже вони заклали основу для розуміння, наскільки вразливими можуть бути базові протоколи мережевої взаємодії. Водночас вони стимулювали розвиток систем захисту — від простих фільтрів у міжмережевих екранах до перших спеціалізованих рішень IDS.

З розвитком Інтернету і збільшенням його масштабів зловмисники почали зміщувати акценти з низькорівневих атак на сервіси, які забезпечують критично важливі функції мережевої інфраструктури. Одним із таких сервісів став DNS (Domain Name System). Сьогодні його справедливо вважають своєрідною «нижньою точкою відліку» для більшості сучасних атак, оскільки компрометація системи імен може паралізувати роботу будь-якої організації.

DNS виступає фундаментом функціонування Інтернету, а отже, будь-які атаки на нього мають масштабні наслідки. Найбільш поширеними є:

- **DNS cache poisoning** — підміна кешованих записів на рекурсивних серверах, що спричиняє перенаправлення користувачів на підроблені ресурси.
- **DNS amplification attack** — використання серверів DNS для здійснення відбивних DDoS-атак, коли незначний за обсягом запит провокує значно більший за обсягом трафік на адресу жертви.
- **Domain hijacking** — захоплення контролю над доменом шляхом зламу облікових записів реєстраторів або експлуатації процедурних вразливостей.

Таким чином, можна відзначити, що класичні атаки нижчих рівнів стали важливим історичним етапом у розвитку кіберзагроз. Вони продемонстрували слабкі місця базових протоколів і стимулювали вдосконалення архітектури мережевої безпеки. Сьогодні ж акценти змістилися вище — на рівень критичних служб, серед яких DNS займає ключове місце. Це пояснюється тим, що будь-які порушення у його роботі здатні призвести до глобальних збоїв, а отже, він стає однією з найпривабливіших цілей для атакувальників.

Сучасні прикладні атаки

Вразливості прикладного рівня сьогодні становлять один із головних і найнебезпечніших векторів атак, оскільки саме тут зосереджені бізнес-критичні сервіси, бази даних, веб-застосунки та інтерфейси взаємодії з користувачами. Зловмисники активно експлуатують будь-які недоліки у логіці застосунків, помилки розробників або неправильні конфігурації, що відкриває шлях до отримання несанкціонованого доступу чи викрадення конфіденційних даних.

Серед найвідоміших уразливостей прикладного рівня залишаються ін’єкції коду. Класичні приклади — SQL injection або PHP injection, які дозволяють вбудовувати шкідливі запити у вихідний код веб-додатків. Хоча у сучасних фреймворках і системах керування базами даних такі проблеми частково вирішені завдяки параметризованим запитам та вбудованим механізмам захисту, вони досі зустрічаються у самописних веб-застосунках, внутрішніх корпоративних сервісах або у вбудованому програмному забезпеченні мережевого обладнання (наприклад, у прошивках маршрутизаторів чи IoT-пристроїв). Це робить ін’єкції актуальною загрозою навіть у 2020-х роках.

Значно поширенішими стали атаки на клієнтську сторону, зокрема:

- **XSS (Cross-Site Scripting)** — впровадження шкідливого коду у веб-сторінки, що дозволяє викрадати cookies, облікові дані користувачів або перенаправляти їх на фішингові сайти.
- **CSRF (Cross-Site Request Forgery)** — примус користувача виконувати несанкціоновані дії у внутрішніх системах (наприклад, підтверджувати

транзакції або змінювати налаштування), використовуючи його активну сесію.

Хоча сучасні веб-фреймворки впроваджують механізми захисту від подібних атак (наприклад, токени перевірки запитів, шифрування cookies, політику SameSite), уразливості продовжують з’являтися внаслідок людських помилок, недосвідченості розробників або використання застарілих бібліотек.

Окрему категорію становлять атаки на інфраструктурні служби корпоративних мереж, серед яких найбільш критичною є система Active Directory (AD). Компрометація AD дозволяє зловмиснику отримати контроль над обліковими записами, службовими політиками та доступом до ресурсів усієї організації. Через свою важливість AD є однією з найпривабливіших цілей атак і потребує ретельного захисту.

З відносно «традиційних» прикладних атак актуальними залишаються експлойти проти SMB-протоколу, який використовується для спільного доступу до файлів і ресурсів у Windows-мережах. Яскравий приклад — поширення шкідливого ПЗ WannaCry яки1 використовував вразливості SMB для швидкої інфекції тисяч комп’ютерів по всьому світу.

Поки триває поступовий повний перехід на використання HTTPS у поєднанні з HSTS (HTTP Strict Transport Security), залишаються можливими атаки на кшталт SSL stripping, коли зловмисник примусово знижує рівень захисту з HTTPS до HTTP, перехоплюючи передані дані у відкритому вигляді. Додатково слід згадати експлуатацію відомих уразливостей у застарілому програмному забезпеченні, таких як HeartBleed у бібліотеці OpenSSL, яка дозволяла викрадати секретні ключі та конфіденційні дані з пам’яті серверів.

Метою більшості сучасних прикладних атак є не лише виведення сервісів з ладу, а насамперед перехоплення управління, викрадення облікових даних, фінансових ресурсів чи критичних бізнес-даних. Це пояснює, чому прикладний рівень сьогодні вважається найбільш привабливим для атакувальників і потребує впровадження комплексних рішень — від безпечної розробки (Secure Coding) та

регулярного пентестингу до використання WAF (Web Application Firewall) та систем поведінкового аналізу.

DoS та DDoS атаки

Атаки відмови в обслуговуванні (DoS, Denial of Service) становлять окремий і надзвичайно небезпечний клас загроз, спрямованих на порушення доступності інформаційних ресурсів. Їхня основна мета полягає у створенні таких умов, за яких легальні користувачі не можуть отримати доступ до сервісу або цей доступ стає нестабільним та непридатним для повноцінного використання. З огляду на те, що сучасні інформаційні системи є критично залежними від безперервності роботи, DoS-атаки мають потенціал завдавати значних економічних збитків і навіть паралізувати діяльність компаній та організацій.

На початкових етапах розвитку Інтернету DoS-атаки здебільшого реалізовувалися на транспортному рівні. Найбільш відомим прикладом є SYN Flood, коли зловмисник масово надсилає запити на встановлення TCP-з'єднання, але не завершує процедуру тристороннього рукостискання. У результаті сервер змушений утримувати велику кількість напіввідкритих з'єднань, що призводить до виснаження ресурсів і відмови в обслуговуванні легітимних клієнтів. Подібні атаки демонстрували, наскільки вразливими були протоколи нижчих рівнів до навмисного зловживання.

Проте з розвитком технологій захисту, удосконаленням стеків TCP/IP та впровадженням механізмів захисту на рівні обладнання, вектори атак поступово змістилися на прикладний рівень. Сучасні атаки часто реалізуються у вигляді DDoS (Distributed Denial of Service), коли атака здійснюється не з одного комп'ютера, а з великої кількості вузлів — зазвичай це цілі ботнет-мережі, що складаються з тисяч або навіть мільйонів заражених пристроїв. Масові запити до сервера створюють навантаження, яке у багато разів перевищує його можливості, внаслідок чого навіть потужні інфраструктури не витримують і виходять з ладу.

Одним із найпоширеніших сценаріїв є HTTP Flood, коли на веб-сервер надсилаються величезні обсяги звичайних, на перший погляд, легальних HTTP-запитів. На відміну від класичних транспортних атак, такий підхід складніше

виявити, адже запити імітують активність реальних користувачів. Це вимагає від систем захисту застосування складних механізмів аналізу поведінки трафіку, виявлення аномалій та впровадження інтелектуальних систем фільтрації.

Крім того, сучасні DDoS-атаки часто комбінують різні техніки: одночасне навантаження на канали зв'язку, виснаження процесорних і пам'яттєвих ресурсів сервера, а також спроби зламати або заблокувати проміжні служби (наприклад, DNS чи CDN). З огляду на масштаби таких атак, для їхньої реалізації нерідко використовуються заражені IoT-пристрої (камери відеоспостереження, маршрутизатори, «розумні» побутові прилади), які є слабко захищеними й легко піддаються компрометації.

Таким чином, сучасні DoS- і DDoS-атаки еволюціонували від простих перевантажень протоколів нижчих рівнів до витончених багаторівневих атак, що орієнтуються на прикладний рівень і використовують глобально розподілену інфраструктуру. Їхня небезпека полягає не лише у масштабності, а й у складності виявлення та відбиття, що робить цей клас загроз одним із ключових викликів у сфері кібербезпеки.

Фішингові атаки

Фішинг і сьогодні залишається одним із найбільш ефективних методів компрометації інформаційних систем, адже він спрямований не на технологічні вразливості, а на людський фактор. Головна ідея полягає у введенні користувача в оману з метою отримання його облікових даних, фінансової інформації або примушення до виконання дій, що призводять до компрометації системи. Саме через це навіть найсучасніші технічні засоби захисту не завжди можуть гарантувати повний захист від фішингу, адже користувачі часто залишаються найслабшою ланкою у системі кібербезпеки.

Типові фішингові атаки здійснюються через електронну пошту, коли зловмисники надсилають повідомлення, стилізовані під офіційні листи від банків, державних установ чи відомих компаній. Такі листи містять підроблені посилання або вкладення, відкриття яких призводить до викрадення даних або зараження пристрою шкідливим програмним забезпеченням. У сучасних умовах активно

використовуються також SMS-повідомлення (smishing) та телефонні дзвінки (vishing), що розширює арсенал атакуючих і робить загрозу ще більш поширеною.

Окрему категорію становить спеар-фішинг (spear-phishing) — цілеспрямовані атаки на конкретних осіб чи організації. Такі атаки відрізняються високим рівнем персоналізації: повідомлення містять реальні дані про жертву, що підвищує довіру та збільшує ймовірність успішної компрометації. Найнебезпечнішими є whaling-атаки, спрямовані на керівників компаній або топ-менеджерів, адже компрометація їхніх облікових записів відкриває доступ до критично важливої інформації та фінансових ресурсів.

Фішинг активно використовується як перший етап складних багатоступеневих атак (APT — Advanced Persistent Threats). Отримавши доступ до корпоративної пошти або внутрішніх систем, зловмисники можуть непомітно закріпитися в інфраструктурі, здійснювати розвідку та готувати подальші кібератаки. Таким чином, навіть один успішний фішинговий лист може призвести до масштабних наслідків для всієї організації.

Варто підкреслити, що незважаючи на наявність численних технічних засобів захисту — фільтрації пошти, антиспам-систем, механізмів двофакторної автентифікації — людський фактор залишається головною проблемою. Брак обізнаності користувачів, відсутність регулярних тренінгів із кібергігієни та довіра до добре замаскованих підробок значно підвищують ризики успіху таких атак.

Таким чином, фішинг не лише зберігає актуальність, але й постійно еволюціонує, використовуючи нові канали комунікації та психологічні прийоми. Саме тому боротьба з ним потребує комплексного підходу: поєднання технічних засобів захисту з постійною просвітницькою роботою серед користувачів. Лише підвищення рівня обізнаності та уважності може істотно знизити ймовірність успішної компрометації за допомогою фішингових атак.

Хостові атаки та їхні особливості

Обмежувати дослідження лише мережевими атаками є суттєвою помилкою, адже сучасні зловмисники активно експлуатують уразливості не лише у комунікаційних протоколах, але й безпосередньо на рівні кінцевих пристроїв.

Прикладом може слугувати атака відмови в обслуговуванні: її можна реалізувати як через перевантаження серверного каналу величезними обсягами трафіку, так і шляхом компрометації адміністративної машини, яка зберігає ключі доступу до критично важливих ресурсів. У такому випадку зловмисник отримує можливість заблокувати або підмінити сервіси зсередини, що часто виявляється ще більш небезпечним, ніж зовнішній DoS-удар.

Дуже часто початковий етап вторгнення розпочинається з компрометації найбільш уразливих користувацьких вузлів. Саме вони стають першою ціллю зловмисників, оскільки зазвичай мають нижчий рівень захисту, недостатньо актуальні оновлення або ж використовуються недосвідченими користувачами. Після отримання контролю над таким вузлом атакувальники використовують його як плацдарм для подальшого поширення у корпоративній мережі. Надалі він може стати як інструментом для збору інформації про інфраструктуру, так і точкою для запуску шкідливого коду, мережевих черв’яків чи троянів.

Класичні атаки прикладного рівня на сервіси, такі як FTP, Sendmail чи Telnet, нині поступово відходять на другий план. На зміну їм приходять атаки безпосередньо на рівні хоста, які часто вимагають активних дій користувача. Прикладами можуть бути відкриття шкідливо модифікованого PDF-файлу, перехід за небезпечним посиланням або перегляд каталогу, зараженого експлойтом. Такі атаки стають особливо ефективними завдяки використанню соціальної інженерії, коли зловмисники маніпулюють користувачами, змушуючи їх самостійно запускати небезпечний код.

Виявлення подібних атак за допомогою IDS мережевого рівня часто виявляється малоефективним. Це пояснюється тим, що атакувальники широко застосовують обфускацію, шифрування чи доставку шкідливих агентів через побічні канали, що унеможливлює їх виявлення лише на основі аналізу трафіку. У таких випадках ключова роль відводиться антивірусним системам, які можуть локально ідентифікувати та нейтралізувати загрозу.

Втім, і тут існують суттєві обмеження: без централізованого керування антивіруси здатні лише локально усунути зараження, не формуючи глобального

контексту події. Наприклад, вони не можуть визначити, чи йдеться про випадкове зараження одного користувача, чи це лише початок масового поширення мережевого черв’яка, що невдовзі паралізує всю інфраструктуру.

Особливу небезпеку становить застосування rootkit та bootkit — спеціалізованих інструментів, які проникають у найнижчі рівні операційної системи, включаючи ядро або навіть завантажувач. Такі інструменти здатні повністю нейтралізувати роботу антивірусних програм, приховувати наявність шкідливого коду, модифікувати системні процеси та забезпечувати зловмиснику прихований і тривалий доступ до зараженого вузла.

Таким чином, атаки на рівні хоста є однією з найсерйозніших сучасних загроз. Вони характеризуються високим рівнем прихованості, широким використанням методів соціальної інженерії та здатністю обходити традиційні засоби захисту. Ефективна протидія таким атакам вимагає комплексного підходу, що поєднує централізовані системи управління безпекою, сучасні антивірусні рішення, інструменти аналізу поведінки та тісну інтеграцію з мережевими IDS/IPS для отримання повної картини інцидентів.

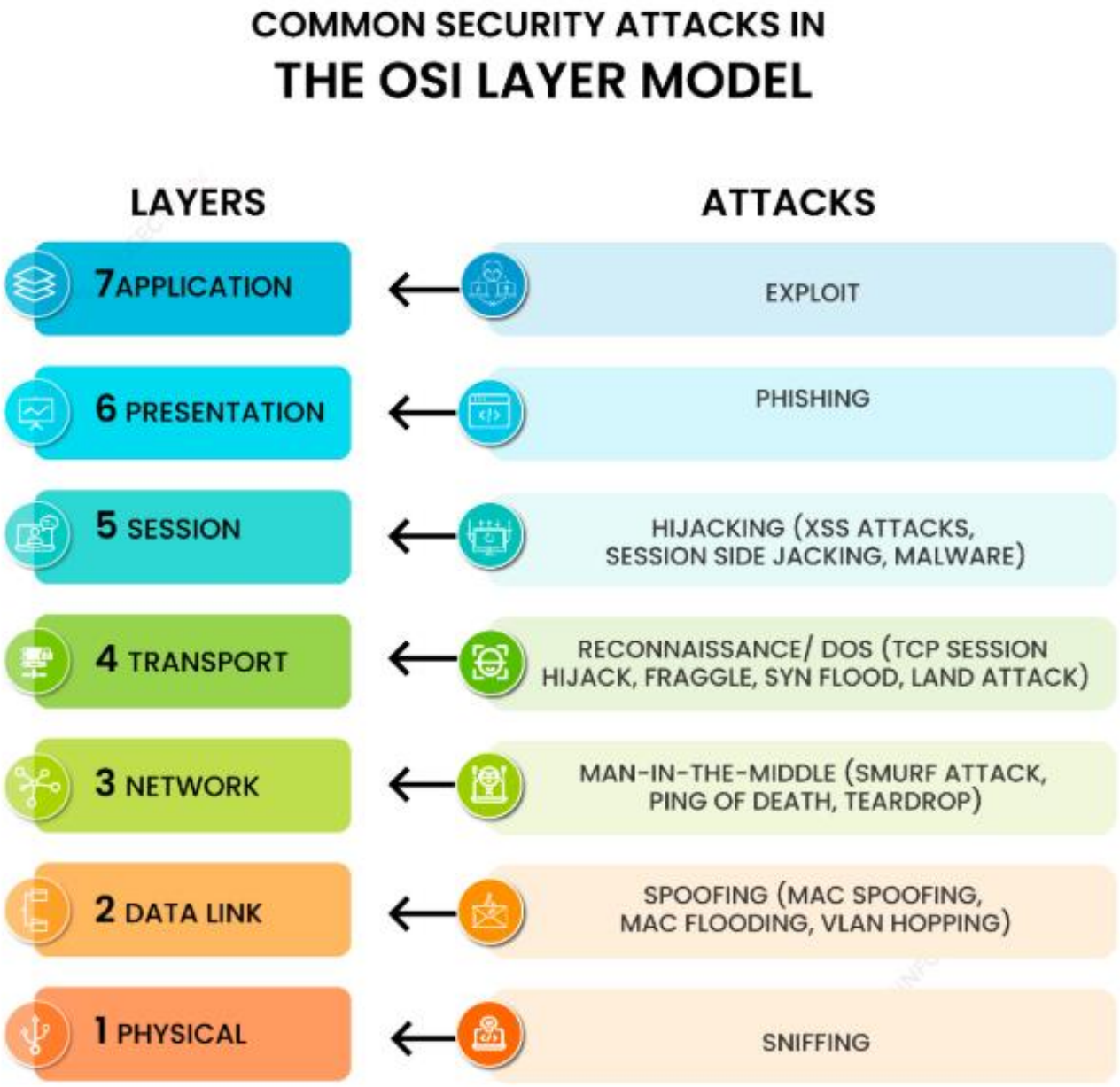


Рис. 3.1 Типові мережеві атаки на різних рівнях моделі OSI: від класичних атак нижчих рівнів до сучасних загроз прикладного рівня

Необхідність знань про мережу

Для успішної реалізації майже будь-якої кібернетичної атаки зловмиснику необхідні попередні знання про топологію та конфігурацію цільової мережі. Винятком можуть бути хіба що суто фізичні дії, як-от відключення сегмента мережі або руйнування обладнання, проте навіть у таких випадках нападник повинен розуміти, яке саме обладнання є критичним, які процеси воно обслуговує та які наслідки матиме його відключення для всієї інфраструктури. Без такої інформації

навіть грубі силові методи можуть виявитися неефективними або не досягти бажаного результату.

Отримати відомості про мережу можна різними способами. Прямі методи передбачають активне збирання даних у режимі взаємодії з інфраструктурою. Найбільш поширеним прикладом є сканування портів, служб і вузлів, яке дозволяє побудувати уявлення про те, які сервіси працюють на кожному пристрої, які версії програмного забезпечення використовуються та які протоколи є відкритими для зовнішніх підключень. Інструменти на кшталт *Nmap*, *Masscan* чи *Zmap* здатні швидко створити детальну карту доступних ресурсів, після чого зловмисник може зіставити отримані дані з відомими уразливостями.

Окрім цього, непрямі методи збору інформації часто є не менш ефективними. До них належать крадіжка конфігураційних файлів, доступ до внутрішньої документації, отриманої внаслідок соціальної інженерії чи витоку даних, а також перехоплення службових повідомлень, які містять метаінформацію про топологію. Наприклад, файл резервної конфігурації маршрутизатора може надати атакувальнику повну картину внутрішньої адресації, списки доступу та маршрути, що значно полегшить подальший етап проникнення.

Сканування у цьому контексті виступає базовим і практично обов’язковим етапом підготовки атаки. Воно дозволяє скласти карту мережевої інфраструктури, визначити активні вузли, виявити слабкі місця та окреслити потенційні точки входу. Навіть найпростіші дії, як-от перевірка доступності вузлів через ICMP-запити (ping sweep), уже формують базове розуміння масштабів мережі. Подальше ж глибоке сканування, яке включає ідентифікацію відкритих портів, виявлення служб, що працюють на них, та визначення версій програмного забезпечення, дає змогу точніше підібрати вектори атак.

Таким чином, знання про мережу є необхідною передумовою будь-якого цілеспрямованого нападу. Вони визначають, наскільки ефективними будуть інші дії зловмисника, від вибору інструментів до методів проникнення. Водночас, із погляду захисту, саме на етапі збору розвідувальної інформації існує можливість

вчасно виявити підозрілу активність, блокувати сканування та ускладнити складання повної картини інфраструктури.

3.2 Ймовірні атаки на різні рівні

Атаки, спрямовані безпосередньо на окремі вузли телекомунікаційної системи, становлять один із найпоширеніших і найбільш небезпечних класів кіберзагроз. Вони націлені на компрометацію ресурсів конкретного пристрою, отримання або розширення прав доступу, втручання в роботу системного програмного забезпечення чи зміну конфігураційного середовища. Такі атаки є критичними, оскільки один скомпрометований вузол може стати точкою входу для подальшого поширення шкідливої активності у всій мережі.

До прикладів атак на рівні вузла належать:

- несанкціоноване підвищення привілеїв, коли користувач із базовим рівнем доступу намагається отримати права адміністратора чи суперкористувача;
- несанкціонований запуск коду від імені зареєстрованого користувача, що дає змогу виконувати дії у системі без відома власника облікового запису;
- запуск коду з правами системних служб чи демонів, що відкриває можливість повного контролю над хостом;
- спроби зміни облікових даних – створення нових користувачів, підвищення привілеїв існуючих облікових записів або модифікація паролів;
- прямий запис у фізичні дискові пристрої в обхід драйверів файлової системи, що становить серйозну загрозу для цілісності даних;
- масове редагування або підміна файлів, включно з виконуваними модулями та бібліотеками;
- зміна системних налаштувань із метою дестабілізації або повного виведення системи з ладу.

Більшість подібних атак супроводжуються техніками маскування, спрямованими на уникнення виявлення антивірусами та системами моніторингу. Проте саме процес маскування може стати індикатором злочинної активності.

Наприклад, техніка ін’єкції в процес (process injection), що активно використовується шкідливими програмами, відстежується багатьма антивірусами й автоматично класифікується як небезпечна активність.

Окремим важливим фактором є аномальна динаміка використання ресурсів. Запуск шкідливого або додаткового коду часто призводить до:

- підвищення завантаження процесора (CPU);
- зростання кількості дискових операцій введення/виведення;
- збільшення використання оперативної пам’яті;
- підвищення інтенсивності мережевих звернень.

Зловмисники рідко намагаються маскувати ці зміни, оскільки вважають їх незначними. Проте при масовому ураженні вузлів формується чіткий патерн використання ресурсів, який може бути виявлений як індикатор широкомасштабної атаки.

Ще однією характерною ознакою атак на вузли є поява нетипової мережевої активності. Це можуть бути регулярні звернення до нових сайтів, які раніше не відвідувалися, або підключення до доменів із високою ентропією назв (виглядають як випадкові комбінації символів, а не словникові слова). На самому вузлі подібна активність може залишатися непоміченою, особливо якщо антивірус її не розпізнав. Однак IDS мережевого рівня здатна зафіксувати такі звернення та сигналізувати про можливу атаку навіть у тих випадках, коли локальний захист не спрацював.

Атаки на мережевому рівні

На відміну від атак на окремі вузли, мережеві атаки мають більш глобальний характер, адже вони спрямовані на компрометацію всієї інфраструктури. До найпоширеніших належать:

- DoS/DDoS-атаки – спроби відмови в обслуговуванні, що перевантажують сервери або канали зв’язку;
- перехоплення управління – підміна або контроль сеансів зв’язку між користувачами чи між клієнтом і сервером;

- несанкціонований доступ до даних – може мати пасивну форму (перехоплення трафіку, «прослуховування» каналів) або активну (ін’єкція шкідливих даних, модифікація повідомлень).

Головна мета таких атак – порушення роботи сервісів, компрометація конфіденційних даних, дестабілізація інфраструктури або створення умов для подальших цільових атак.

Сканування як підготовчий етап

У більшості випадків перед безпосередньою атакою виконується сканування, яке дозволяє зловмиснику скласти карту мережі, виявити слабкі місця та спланувати подальші дії. Виділяють два основні різновиди:

1. Сканування вузлів (host discovery) – множина вузлів (n) надсилає подібні запити до іншої множини вузлів (m), виключаючи самозвернення. Якщо інтенсивність сканування висока, то такий трафік легко виявляється IDS мережевого рівня. Якщо ж сканування виконується повільно й розтягнуте в часі, потрібні методи з меншою чутливістю до часових розривів.
2. Сканування сервісів (service discovery) – множина вузлів (n) надсилає запити до іншої множини (m) на кілька портів (p). Такий тип активності простіше виявити засобами на рівні самого вузла, адже він створює специфічні сліди у системних журналах і процесах.

Таким чином, сканування виступає розвідкою боєм, і його своєчасне виявлення може запобігти подальшій компрометації.

Збір та аналіз даних

Для ефективного виявлення атак важливо збирати показники як на рівні окремих вузлів, так і на рівні всієї мережі.

Дані вузла:

- завантаження CPU (%);
- кількість дискових операцій введення/виведення (операцій за секунду);
- використання оперативної пам’яті (%);
- інтенсивність мережевої активності (Мбіт/с).

Ці параметри формують вектор характеристик вузла, який дозволяє відстежувати його поведінку у часі та фіксувати відхилення від еталонної моделі.

Дані мережі:

- послідовності TCP-пакетів (флаги, порти відправника й отримувача, IP-адреси джерел);
- послідовності ICMP-пакетів (IP-адреса відправника, ICMP_ID);
- характеристики UDP-трафіку (IP-адреса джерела, довжина пакета, порти відправника та отримувача).

Аналізуючи ці дані, можливо виявляти не лише відомі атаки за сигнатурами, а й нові типи загроз за допомогою методів статистичного та поведінкового аналізу. Особливу роль відіграють адаптивні алгоритми машинного навчання, які здатні виявляти закономірності у високовимірних просторах ознак і автоматично сигналізувати про появу відхилень.

Отже, атаки як на рівні вузла, так і на мережевому рівні мають широкий спектр проявів та вимагають багаторівневого підходу до виявлення. Своєчасний збір та інтеграція даних, використання методів інтелектуального аналізу та кластеризації дозволяють значно підвищити ефективність систем безпеки та забезпечити надійний захист телекомунікаційних мереж від сучасних кіберзагроз.

3.3 Опис типової архітектури IDS

Система виявлення вторгнень (IDS) є складним комплексом програмних та апаратних засобів, призначених для виявлення ознак несанкціонованої активності у мережі чи на окремих хостах. Вона базується на багаторівневій архітектурі, де кожен компонент виконує свою спеціалізовану функцію, а у сукупності всі вони забезпечують повний цикл від збору даних до реагування на загрози.

Датчики (сенсори)

Датчики виступають першою лінією оборони. Їхнє головне завдання полягає у захопленні інформації з різних джерел, її попередньому опрацюванні та передачі у центральну систему для подальшого аналізу. У сучасних IDS датчики, як правило,

розподілені по декількох вузлах мережі чи навіть по різних сегментах інфраструктури, що дає змогу реалізувати принцип децентралізованого моніторингу.

До складу датчиків входять:

- **модулі захоплення** – перехоплюють трафік, системні події чи інші дані, що можуть містити ознаки атак;
- **модулі збору** – акумулюють отриману інформацію, структурують її та готують до подальшої обробки;
- **модулі інтеграції (сопряження)** – забезпечують взаємодію з центральною підсистемою аналізу та синхронізують роботу датчиків у розподіленому середовищі.

Датчики поділяються на кілька категорій залежно від рівня, з якого вони отримують інформацію:

- **датчики прикладного рівня** — відстежують роботу конкретного програмного забезпечення (наприклад, поштових серверів, веб-додатків або баз даних);
- **датчики хостового рівня** — збирають дані про діяльність окремої робочої станції чи сервера, включно з системними викликами, змінами у файловій системі чи поведінкою процесів;
- **мережеві датчики** — контролюють обмін пакетами та аналізують трафік у реальному часі;
- **міжмережеві датчики** — фіксують потоки даних між різними сегментами мережі, що дозволяє виявляти спроби вторгнень на кордонах інфраструктури.

Таким чином, датчики створюють багатовимірну картину стану системи, яка є основою для подальшого аналізу.

Підсистема аналізу

Зібрана сенсорами інформація надходить у підсистему аналізу, яка перетворює «сирі» дані на набір ознак і здійснює оцінку їх відповідності нормальним або аномальним сценаріям роботи.

До її складу входять:

- **модулі декодування** – переводять вхідні дані у формат, зручний для машинного аналізу, виділяючи ключові параметри (IP-адреси, порти, прапори TCP, послідовності запитів тощо);
- **модулі виявлення** – здійснюють безпосереднє розпізнавання атак або відхилень від політики безпеки.

Модулі виявлення можуть працювати у двох основних режимах:

1. **Сигнатурний аналіз.** Пошук у трафіку відомих шаблонів атак на основі попередньо підготовленої бази сигнатур. Такі бази постійно оновлюються та містять характерні ознаки поширених загроз.
2. **Аномальний аналіз.** Виявлення поведінки, що відрізняється від звичайної. Для цього застосовуються статистичні методи, евристичні правила або алгоритми машинного навчання.

Окрему увагу заслуговує використання **нейронних мереж і гібридних підходів**, які поєднують сигнатурні методи з алгоритмами поведінкового аналізу. Це дозволяє виявляти не лише відомі, але й нові, раніше невідомі типи атак.

Підсистема виведення та реагування

Результати роботи модулів аналізу передаються у підсистему виведення. Її завдання – надати оператору зрозумілу інформацію про загрози та забезпечити автоматичні або напівавтоматичні механізми реагування.

До неї входять:

- **модулі представлення результатів** – формують звіти, журнали подій, сповіщення адміністраторам, відображають інформацію у графічних інтерфейсах;
- **модулі реагування** – здійснюють активні дії у відповідь на загрози, наприклад, блокування підозрілих з’єднань за допомогою брандмауера, ізоляцію зараженого вузла від мережі, примусове завершення небезпечного процесу тощо.

Таким чином, IDS не лише сповіщає про потенційну атаку, а й може виступати у ролі інтегрованого елемента системи захисту, який безпосередньо взаємодіє з іншими компонентами інфраструктури кібербезпеки.



Рис. 3.2 Структура IDS

4 ПОБУДОВА НЕЙРОМЕРЕЖЕВОГО АНАЛІЗАТОРУ

4.1 Опис принципу

Аналіз сучасних підходів до побудови систем виявлення вторгнень свідчить, що найбільш ефективним рішенням є створення комбінованих детекторів, які поєднують здатність фіксувати як відомі атаки, так і нові, нетипові аномалії. Такий підхід дозволяє значно підвищити надійність системи безпеки, оскільки традиційні сигнатурні методи добре працюють для виявлення вже відомих загроз, але малоефективні проти нових типів атак, тоді як моделі поведінкового або статистичного аналізу здатні сигналізувати про відхилення навіть у випадках, коли сигнатура ще не створена.

В окремих сценаріях замість єдиного універсального детектора доцільно будувати ланцюжки взаємодіючих модулів, де кожен виконує спеціалізовану функцію. Наприклад, перший детектор може працювати за сигнатурним принципом, другий – виконувати поведінковий аналіз, а третій – застосовувати алгоритми машинного навчання для багатовимірної класифікації. Така багаторівнева архітектура підвищує стійкість системи до помилкових спрацьовувань та дозволяє значно розширити спектр виявлюваних загроз.

Крім того, ефективна IDS повинна мати механізми періодичного активного сканування, які дають змогу виявляти потенційні вразливості ще до того, як вони будуть використані зловмисниками. Таким чином, система поєднує дві ключові функції: превентивну (завдяки активному пошуку вразливостей) та реактивну (шляхом реєстрації аномалій і атак у реальному часі).

З огляду на великий обсяг завдань, необхідних для створення повноцінного комбінованого детектора, у даному дослідженні увага зосереджується на розробці концептуального модуля виявлення аномалій. Передбачається, що вхідні дані надаються у вже обробленому вигляді від модуля декодування у формі набору ознак. Такі ознаки подаються на вхід нейромережі або іншої алгоритмічної моделі.

У даній роботі як базу для навчання використовується NSL-KDD, яка є однією з найвідоміших репрезентативних вибірок для дослідження систем виявлення атак.

Враховуючи поставлену задачу, метод представлення результатів детектора доцільно будувати багатокласовим, адже це дозволяє не тільки фіксувати факт наявності аномалії, але й визначати її конкретний тип (наприклад, DoS, Probe, R2L тощо). Однак з метою спрощення на даному етапі реалізації обирається двокласова схема («аномалія/норма»), яка демонструє базовий підхід і може слугувати основою для подальшого розширення.

Усі детектори аномалій, незалежно від алгоритмічної моделі, базуються на принципі зіставлення поточного стану з еталонним. Еталон визначає нормальні умови функціонування системи, а будь-які суттєві відхилення сигналізують про потенційну атаку.

Для статистичних моделей таким еталоном є набір обчислених характеристик розподілу параметрів; для імунних систем – профіль нормальної активності; для нейронних мереж – ваги та структура, сформовані під час навчання. У випадку нейромережі «еталон» формується у процесі попереднього навчання на даних, що відображають звичайний стан мережі передачі даних. Після цього мережа здатна виявляти відхилення, що виходять за встановлені межі.

Разом із тим такий підхід має низку складних аспектів. По-перше, навчання повинно здійснюватися у стані нормального функціонування мережі, але неможливо гарантувати, що під час збору навчальних даних атаки не відбуваються. Якщо аномальна активність потрапляє до тренувальної вибірки, то мережа ризикує «прийняти» її за норму. По-друге, повне відключення мережі від зовнішніх каналів або її ізоляція задля збору «чистих» даних часто є неможливими через критичність її функціонування. У результаті будь-які відхилення у майбутньому можуть помилково вважатися аномалією.

Ще однією проблемою є необхідність збору значних обсягів даних для навчання, оскільки поведінка мережі змінюється залежно від часу доби, днів тижня та навіть сезонів року. Для отримання достовірної моделі потрібен великий часовий

зріз, який охоплює всі можливі сценарії «нормальної» активності. Це значно збільшує як обсяг роботи, так і ресурси, необхідні для навчання.

Таким чином, створення ефективного детектора аномалій вимагає ретельного вибору навчальної вибірки, оптимізації алгоритмів та балансування між чутливістю і кількістю хибних спрацьовувань. Перспективним напрямом розвитку таких систем є застосування гібридних підходів, які поєднують статистичні методи, машинне навчання та адаптивні алгоритми аналізу поведінки. Це дозволить IDS не лише ефективно реагувати на сучасні загрози, але й адаптуватися до нових умов функціонування мережевої інфраструктури.

Однією з фундаментальних труднощів у побудові систем виявлення аномалій є те, що процес навчання має здійснюватися у період нормального функціонування мережі передачі даних (МПД). Утім, повністю гарантувати відсутність атак у цей час неможливо. У низці підходів цю проблему намагаються розв’язати шляхом формалізації та створення набору правил, які описують роботу системи в штатному режимі. Такі правила дозволяють задати межі «еталонної» поведінки. Проте у випадках, коли трафік є різнорідним і динамічним, подібний метод втрачає ефективність і практичність.

Ще однією складністю є значний обсяг даних, необхідних для формування надійної моделі. Робота МПД може суттєво змінюватися залежно від часу доби, днів тижня, сезону року або навіть від зовнішніх факторів, наприклад, зростання навантаження у пікові періоди. У результаті модель, навчена на обмеженому наборі даних, ризикує сформувати викривлене уявлення про нормальну поведінку мережі. Це призводить або до зниження чутливості (атаки залишаються непоміченими), або до надлишкової кількості хибних спрацьовувань.

Частково розв’язати цю проблему дозволяє використання адаптивних методів підлаштування, які забезпечують динамічне коригування «нормального» профілю системи з урахуванням нових вхідних даних. Такий підхід дозволяє IDS поступово оновлювати уявлення про роботу мережі, уникати застарівання моделей та підвищувати точність виявлення. Зокрема, алгоритми машинного навчання можуть

реалізовувати онлайн-навчання, коли модель оновлюється безпосередньо під час експлуатації, інтегруючи нові патерни поведінки без повного перенавчання.

Практичне значення такого підходу можна продемонструвати на основі даних з бази NSL-KDD, що є стандартним набором для дослідження систем виявлення вторгнень. Як показує аналіз, профілі мережевої активності у випадку атак істотно відрізняються від профілів нормальної роботи. Це підтверджує доцільність використання моделей, здатних у реальному часі визначати та оновлювати межі нормальної поведінки.

Таким чином, адаптивна підбудова є одним із ключових напрямів підвищення ефективності сучасних IDS. Вона дозволяє компенсувати проблему різнорідності даних, забезпечує стійкість системи до змін умов експлуатації та знижує ймовірність помилкових спрацьовувань.

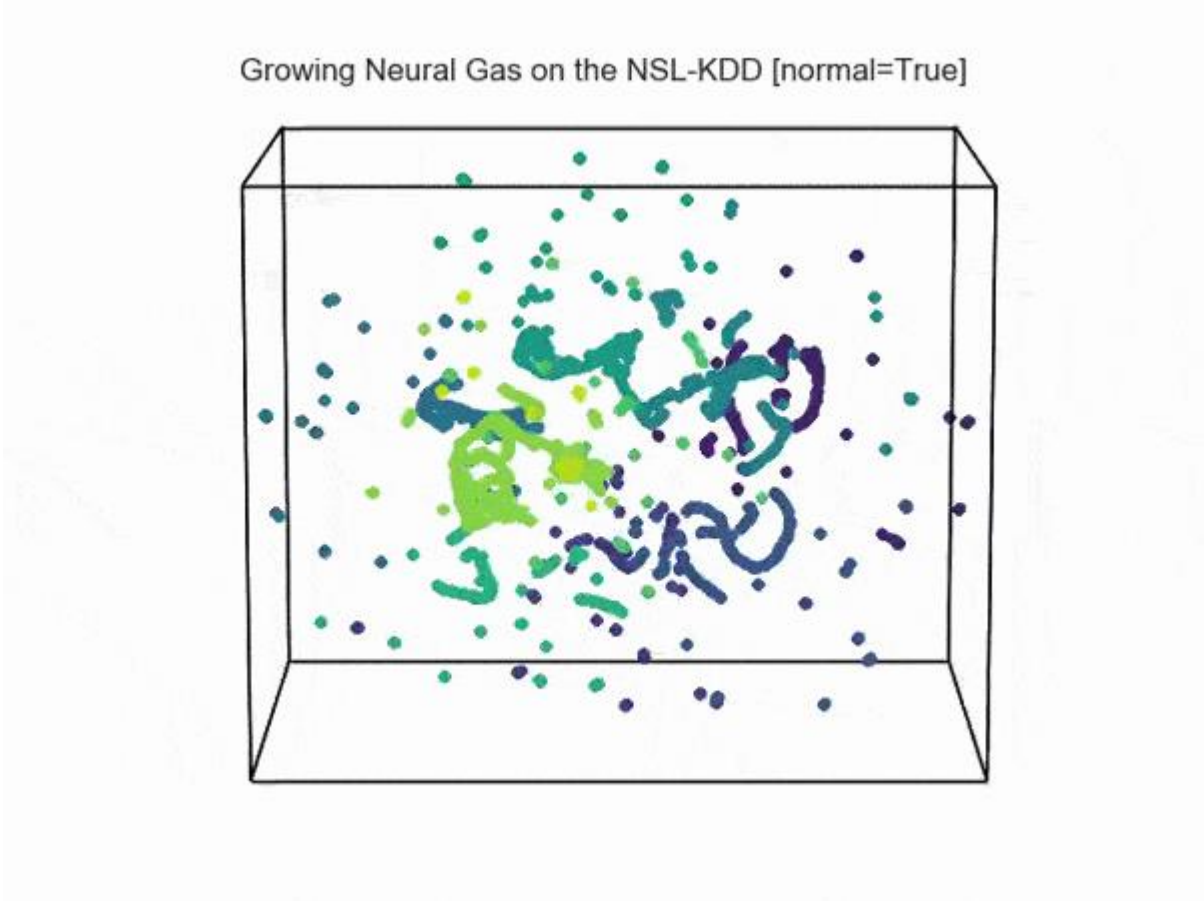


Рис. 4.1 Нормальна активність в мережі

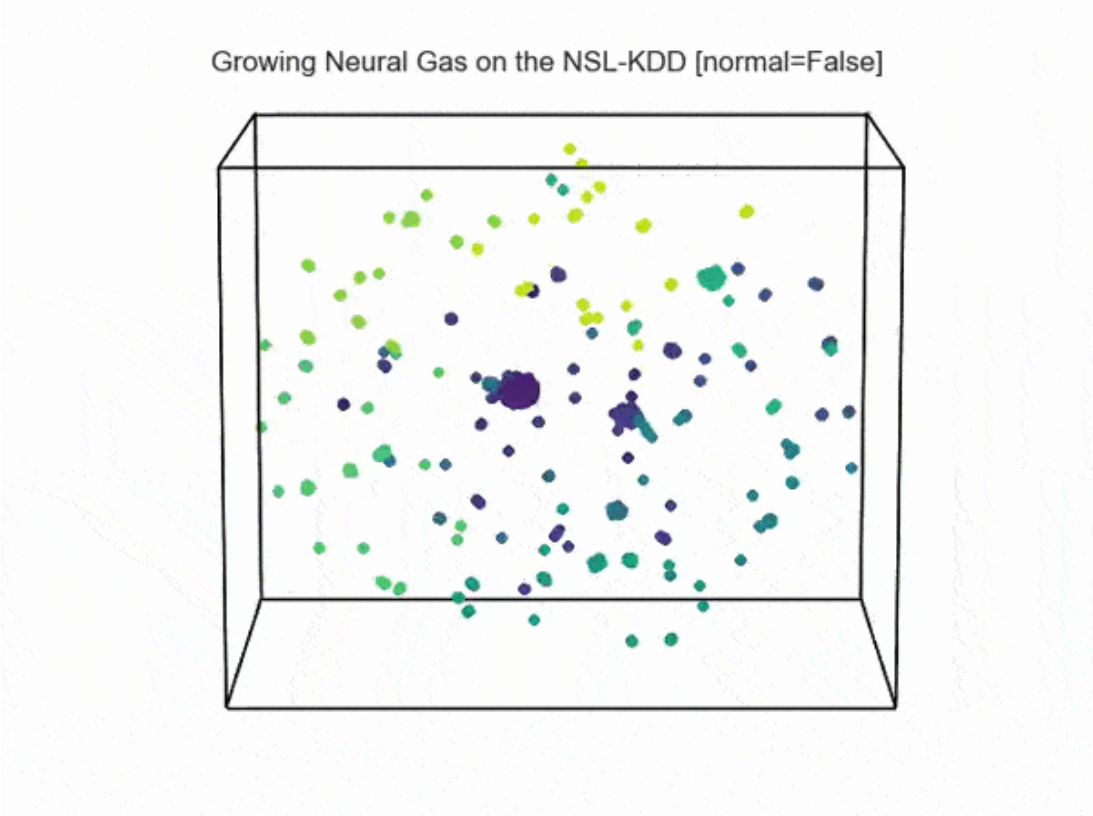


Рис. 4.2 Аномальна активність в мережі

Тобто, можна побачити аномальну активність атаки на тлі нормальної активності мережі:

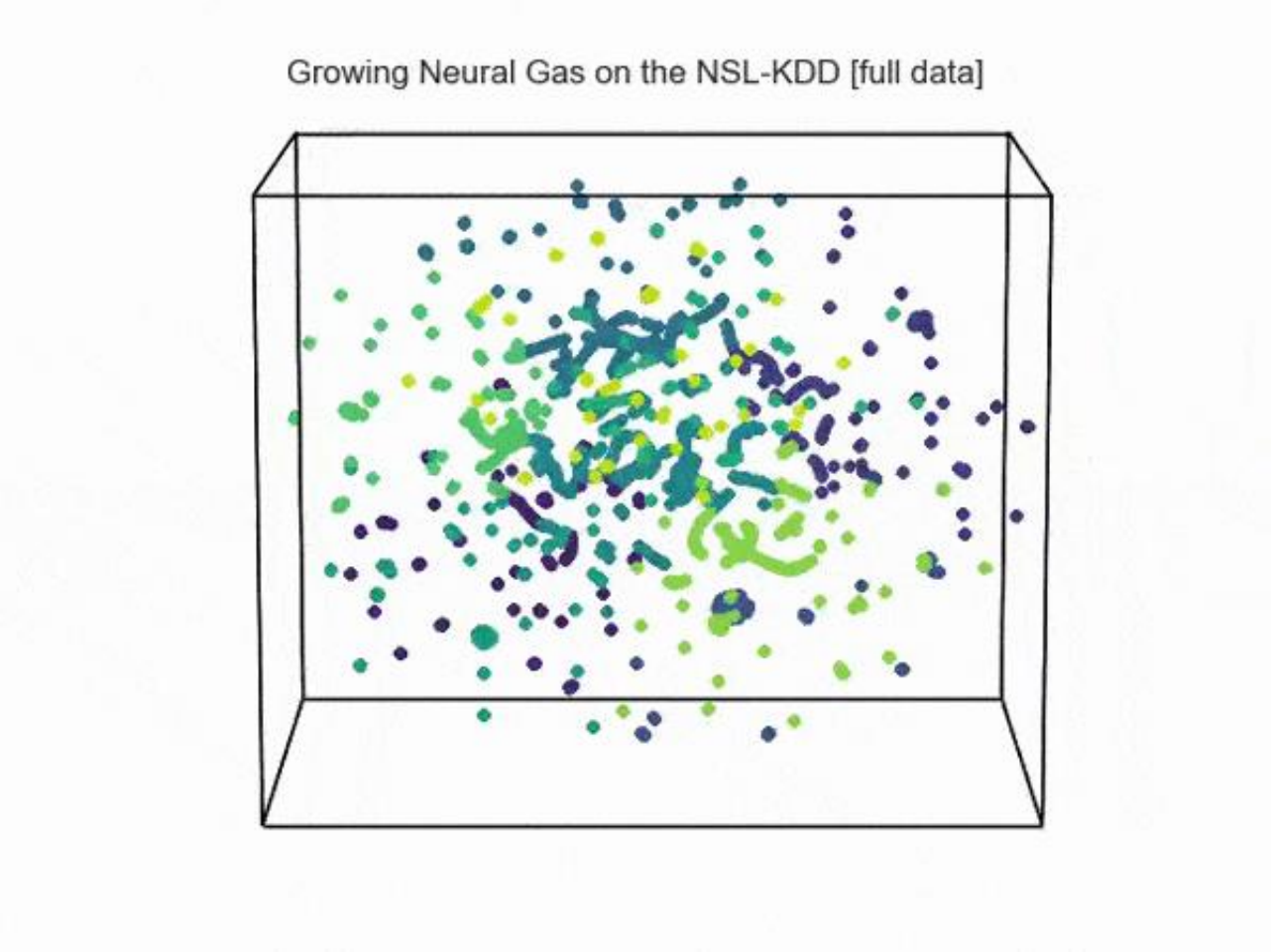


Рис. 4.3 Накладення аномальної поведінки в мережі, при взломі, на нормальну

Нижче представлено код, що використовувався для візуалізації:

```
import hypertools as hyp
from collections import OrderedDict
import csv

def read_ids_data(data_file, is_normal=True, labels_file='NSL_KDD/Field Names.csv',
with_host=False):
    selected_parameters = ['duration', 'protocol_type', 'service', 'flag', 'src_bytes',
'dst_bytes', 'land', 'wrong_fragment', 'urgent']

    # "Label" - "converter function" dictionary.
    label_dict = OrderedDict()
    result = []
```

```
    with open(labels_file) as lf:
        labels = csv.reader(lf)
        for label in labels:
            if len(label) == 1 or label[1] == 'continuous':
                label_dict[label[0]] = lambda l: np.float64(l)
            elif label[1] == 'symbolic':
                label_dict[label[0]] = lambda l: sh(l)

    f_list = [i for i in label_dict.values()]
    n_list = [i for i in label_dict.keys()]

    data_type = lambda t: t == 'normal' if is_normal else t != 'normal'

    with open(data_file) as df:
        # data = csv.DictReader(df, label_dict.keys())
        data = csv.reader(df)
        for d in data:
            if data_type(d[-2]):
                # Skip last two fields and add only specified fields.
                net_params = tuple(f_list[n](i) for n, i in enumerate(d[:-2]) if n_list[n] in
selected_parameters)

                if with_host:
                    host_params = generate_host_activity(is_normal)
                    result.append(net_params + host_params)
                else:
                    result.append(net_params)

    hyp.plot(np.array(result), '.', normalize='across', reduce='UMAP', ndims=3,
```

```
    n_clusters=10, animate='spin', palette='viridis',
    title='Growing Neural Gas on the NSL-KDD [normal={}]'.format(is_normal),
    # vectorizer='TfidfVectorizer',
    # precog=False,  bullettime=True, chemtrails=True, tail_duration=100,
    duration=3, rotations=1, legend=False, explore=False, show=True,
    save_path='./video.mp4')


read_ids_data('NSL_KDD/20 Percent Training Set.csv')
read_ids_data('NSL_KDD/20 Percent Training Set.csv', is_normal=False)
```

Ефективне виявлення аномалій у складних мережевих системах можливе завдяки застосуванню методів кластеризації поведінкових даних. Суть підходу полягає в тому, що загальна «картина мережевої активності» розглядається як сукупність багатовимірних ознак, які описують стан мережі у різні моменти часу. Подальше групування цих ознак у кластери дозволяє виділити типові сценарії функціонування системи та виявляти ті спостереження, які не відповідають жодному з відомих кластерів, тобто свідчать про аномалії або потенційні атаки.

У рамках нейромережевого підходу завдання кластеризації традиційно реалізується із застосуванням самоорганізовуваних карт Кохонена (Self-Organizing Maps, SOM). Цей метод має низку переваг: він дозволяє візуалізувати багатовимірні дані на двовимірній площині, зберігає топологічну подібність між об’єктами та формує зрозумілу структуру кластерів. Однак SOM вимагають заздалегідь визначати кількість кластерів, а також повторного навчання при зміні топології мережі чи структури трафіку, що значно обмежує їхню практичну застосовність у реальних умовах.

Щоб уникнути зазначених проблем, пропонується використовувати адаптивні нейромережеві алгоритми кластеризації, які автоматично підлаштовуються до змін у даних. Вони здатні самостійно створювати або видаляти кластери, адаптуючись до нових сценаріїв роботи мережі, що робить систему більш гнучкою та стійкою.

Основна ідея

Мережа навчається у фазі нормального функціонування мережі передачі даних (МПД) протягом певного проміжку часу. У результаті формується певна кількість кластерів, які відображають типові профілі роботи системи. Якщо алгоритм налаштовано коректно, ця кластерна структура стає еталонною моделлю нормальної поведінки.

Варіанти виявлення аномалій

1. **Навчання у реальному часі.** Система постійно оновлює свою модель, приймаючи нові дані. Поява аномалій призводить до змін у структурі кластерів – зростання їх кількості, утворення нових нейронів та зв’язків, або ж прискорення темпів їх появи й видалення. Така динаміка може розглядатися як сигнал про потенційну загрозу.
2. **Перевірка на близькість до існуючих кластерів.** У цьому випадку нові дані не впливають на структуру мережі, а лише перевіряються на відповідність уже сформованим кластерам. Якщо нові вектори ознак значно відрізняються від середніх значень «еталонних» кластерів, вони класифікуються як аномальні.

Переваги та проблеми підходів

Перший варіант є привабливим завдяки своїй адаптивності, але він має суттєві обмеження. Зокрема, нейронна мережа може змінювати кількість нейронів і зв’язків на кожній ітерації, що ускладнює визначення, які саме зміни є проявом атаки, а які — звичайним процесом адаптації. Крім того, постійне оновлення створює ризик перенавчання, коли аномальна поведінка з часом сприймається як нормальна.

Другий підхід вважається більш надійним, оскільки спирається на чітке вимірювання відхилень від еталону. Він дозволяє уникнути проблеми неконтрольованої динаміки кластерів і забезпечує стабільність результатів. Саме цей метод обрано як базовий у даному дослідженні.

4.2 Архітектура модуля

Збір та обробка даних у сучасних системах виявлення вторгнень передбачає інтеграцію показників з різних рівнів інфраструктури. Для цього агенти, розташовані на вузлах, здійснюють пересилання зібраної інформації до центральної системи. На першому етапі для кожного з параметрів (наприклад, завантаження CPU, споживання оперативної пам’яті, частота дискових операцій, обсяг вхідного та вихідного трафіку) обчислюється середнє значення, яке відображає базовий рівень використання ресурсів. Отримані значення не розглядаються окремо, а об’єднуються у вектор характеристик вузла, який, у свою чергу, інтегрується з вектором параметрів мережевої активності, що надходять від IDS. Таким чином формується єдиний профіль поведінки системи, який враховує як локальні показники окремих хостів, так і глобальні характеристики трафіку.

У прототипі генерація даних хостів здійснюється за допомогою функції, яка моделює аномальну поведінку. Зокрема, під час певної частки атак (на поточному етапі цей показник становить 25%) функція штучно збільшує рівень споживання системних ресурсів. Це дозволяє перевірити здатність системи розрізняти звичайні та відхилені сценарії роботи.

Дані про мережеву активність подаються до модуля IDS у стандартизованому форматі, що гарантує їхню сумісність із алгоритмами аналізу. У прототипі в якості вхідних даних використовуються записи з відомої бази NSL-KDD, яка є еталонною для досліджень у сфері виявлення вторгнень і містить різноманітні приклади нормальної та аномальної поведінки мережі.

На наступному етапі реалізації передбачено впровадження алгоритму, орієнтованого на виявлення відхилень від середніх значень параметрів. Ідея полягає у тому, що для кожного вектора характеристик обчислюється його близькість до еталонних середніх значень, отриманих у процесі навчання. Якщо значення перевищує допустимий поріг відхилення, воно класифікується як аномальне.

Такий підхід має низку переваг. По-перше, він дозволяє поєднати локальні та глобальні дані, що забезпечує багаторівневий контроль. По-друге, метод виявлення

відхилень від середнього є відносно простим з обчислювальної точки зору, що робить його придатним для реалізації у системах з високими вимогами до продуктивності. По-третє, у поєднанні з даними бази NSL-KDD цей підхід дає змогу ефективно відпрацьовувати як типові сценарії атак, так і рідкісні випадки, що імітують складні загрози.

У майбутньому така методика може бути розширена шляхом використання адаптивних порогів, які враховуватимуть динаміку зміни середніх значень у часі, що дозволить зменшити кількість хибних спрацьовувань і підвищити надійність системи.

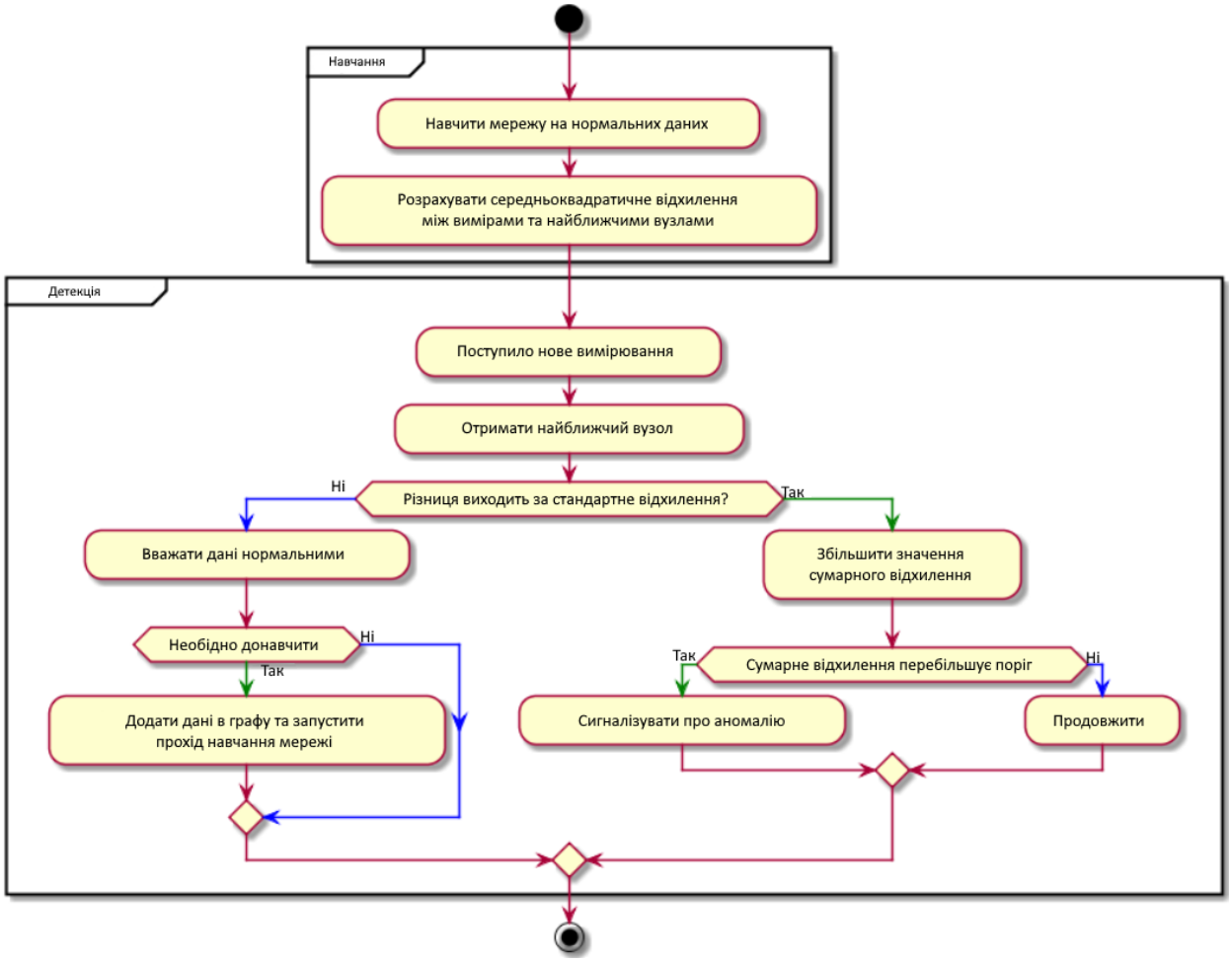


Рис. 4.4 Реалізація алгоритму

У процесі побудови системи виявлення аномалій важливим завданням є вибір механізму кластеризації даних, що здатний не лише відображати багатовимірну структуру трафіку, а й динамічно підлаштовуватися під зміни у мережі. Одним із

найбільш перспективних підходів є застосування адаптивних алгоритмів, що поєднують властивості навчання без учителя з елементами контролю.

У запропонованому алгоритмі реалізується адаптивна підбудова, яка функціонує подібно до методів навчання з частковим залученням учителя. Головна ідея полягає в тому, що система аналізує не окремі відхилення, а середні відхилення для груп нейронів, які найточніше описують вхідні дані. Це дозволяє зменшити вплив випадкових шумів та зробити класифікацію більш стійкою.

Для кожного кластера додатково обчислюється корінь із середнього арифметичного всіх відхилень, що відповідають нейронам у його складі. Коли новий зразок даних надходить на перевірку, система знаходить найближчий до нього нейрон і визначає відстань між ними. Ця відстань порівнюється із середньоквадратичним відхиленням, розрахованим для відповідного кластера. Якщо відстань перевищує встановлений поріг, то зразок класифікується як аномальний.

Таким чином, у роботі використовується гібридний метод класифікації, який поєднує:

- підхід за середнім відхиленням (класифікація даних на основі порівняння з типовим профілем кластера);
- підхід за мінімальною відстанню (визначення найбільш близького нейрона у просторі ознак).

Це дозволяє спочатку вибрати «найкращий» нейрон для нового зразка, а потім оцінити ступінь його відхилення від усереднених параметрів кластера.

Слід зазначити, що інший підхід — класифікація за принципом більшості (коли клас нейрона визначається найбільшою кількістю точок у його області Вороного) — у даному випадку не використовується, оскільки він є менш ефективним для задачі виявлення аномалій у складних і динамічних мережах.

Ядро розробленого детектора побудоване на основі алгоритмів сімейства «зростаючого нейронного газу» (Growing Neural Gas, GNG). Вони навчаються на даних мережевої активності у режимі нормального функціонування мережі

передачі даних (МПД) та здатні самостійно визначати кількість кластерів, що значно підвищує адаптивність системи.

До основних варіацій цього підходу належать:

- **GNG (Growing Neural Gas)** – класичний алгоритм, який формує мережу нейронів шляхом поступового додавання нових вузлів і ребер, відображаючи структуру даних. Він широко застосовується для кластеризації та виявлення прихованих закономірностей у високорозмірних просторах.
- **IGNG (Incremental Growing Neural Gas)** – інкрементальна модифікація класичного алгоритму. Вона забезпечує швидшу збіжність і краще пристосовується до динамічно змінюваних потоків даних. У ряді робіт його також називають «швидкоростучим GNG». Його характерною особливістю є розподіл нейронів на зрілі (фіксовані) та зародкові (нефіксовані), що дає змогу більш детально відстежувати процес еволюції кластерної структури.

Використання GNG та його модифікацій має низку переваг для систем IDS. По-перше, вони дозволяють динамічно адаптуватися до змін у мережевій поведінці без потреби повного перенавчання. По-друге, такі алгоритми здатні ефективно працювати з потоковими даними у режимі реального часу, що є критично важливим для задач кібербезпеки. По-третє, вони забезпечують можливість поєднання детекції за середніми профілями та відхиленнями, що значно знижує кількість хибних спрацьовувань.

4.3 Робота модуля та інтерпритація результатів

Розроблений модуль зберігає дані, використані під час навчання, у вигляді структури нейронної мережі, що відображає сформовані кластери та зв’язки між ними. Це забезпечує можливість не тільки відтворювати «еталонну» модель нормальної поведінки, але й поступово її уточнювати у процесі подальшої експлуатації. У робочому режимі модуль безперервно отримує нові дані від агентів і підсистем IDS, інтегрує їх у загальну структуру й виконує оцінку на предмет відхилень від нормальних профілів.

На виході система повинна постійно формувати інтерпретований результат у зручному для аналізу вигляді. Залежно від обраної конфігурації це може бути:

- бінарний висновок — «аномалія виявлена» або «аномалія відсутня»;
- розширений висновок із зазначенням ймовірності виникнення аномальної активності, що дозволяє оператору приймати більш гнучкі рішення;
- багатокласова класифікація, коли система визначає не лише факт наявності аномалії, але й її тип (наприклад, DoS, сканування портів, підозрілий доступ до сервісів тощо).

Важливо, що такий підхід забезпечує безперервний моніторинг і дозволяє виявляти як явні порушення, так і малопомітні відхилення, які потенційно можуть свідчити про початкові етапи атаки.

Для верифікації ефективності обраного підходу застосовується візуалізація кластеризації даних, яка наочно демонструє процес формування й стабілізації кластерів. На рисунку наведено приклад роботи алгоритму Growing Neural Gas (GNG), побудованого на обмеженій підмножині даних з метою кращої ілюстрації. На графіку чітко простежується, як алгоритм поступово сходиться, формуючи структуру, що відображає закономірності вхідних даних.

Подібна візуалізація є не лише ілюстративним інструментом, але й корисним засобом для діагностики роботи алгоритму. Вона дозволяє оцінити швидкість збіжності, адекватність сформованих кластерів і здатність моделі розділяти нормальні та аномальні сценарії. У практичному застосуванні це допомагає розробникам налаштовувати параметри системи, а адміністраторам безпеки — краще інтерпретувати результати, отримані від модуля.

Таким чином, модуль виконує не лише автоматизовану обробку даних, але й створює інструменти для наочного представлення процесу виявлення, що підвищує прозорість його роботи та довіру до результатів аналізу.

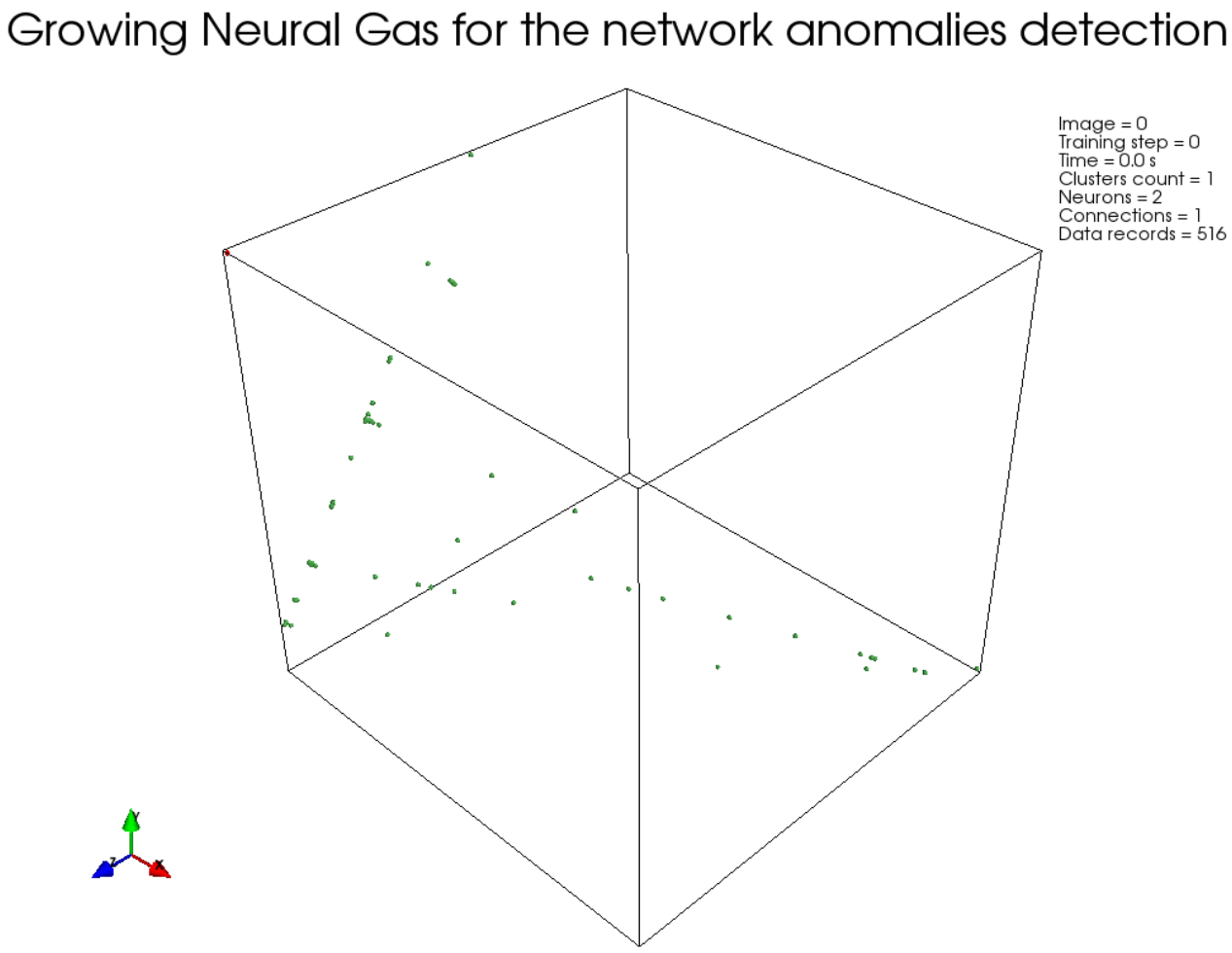


Рис. 4.5 Візуалізація роботи модуля

Реалізація модуля

Сьогодні існує низка реалізацій алгоритму GNG (Growing Neural Gas) різними мовами програмування. Серед них можна виділити:

- реалізацію на MQL-5. Синтаксис цієї мови подібний до C++, що спрощує її розуміння, проте стиль кодування видається дещо громіздким і малопридатним для подальшої модифікації;
- реалізації на R та паралельно на C++, що часто використовуються у наукових роботах;
- реалізацію на MATLAB, яка зручна для моделювання та візуалізації;
- реалізацію на Go, орієнтовану на високу швидкодію;
- реалізацію на JavaScript, яка використовується переважно для інтерактивних симуляцій алгоритмів;

- кілька незалежних реалізацій на Python, серед яких для роботи було обрано другу, оскільки вона коректно запрацювала одразу після встановлення необхідних бібліотек.

Враховуючи доступність бібліотек та універсальність мови, для реалізації прототипу було обрано Python. Натомість з алгоритмом IGNG (Incremental Growing Neural Gas) ситуація виявилася складнішою: жодної відкритої реалізації знайти не вдалося, хоча самі автори у своїх роботах отримували результати на основі цього методу. Тому алгоритм IGNG довелося розробляти з нуля, що додало дослідницькій роботі елемент оригінальності.

Структура коду та основні компоненти

Точкою входу програми виступає функція main(), яка визначає початкові параметри та викликає функцію test_detector(), що відповідає за запуск і тестування детектора.

У test_detector() виконується попереднє завантаження даних, далі модуль проходить фазу навчання через метод train(), після чого здійснюється перевірка на різних підмножинах даних за допомогою методу detect_anomalies().

Завантаження здійснюється функцією read_ids_data(), яка читає записи з бази NSL-KDD за допомогою стандартного модуля csv. Тип вибірки (нормальна, аномальна або повна) задається параметром *activity_type*. За необхідності викликається функція generate_host_activity(), яка додає згенеровану активність хостів до даних про мережу. На виході функція формує ненормалізований масив NumPy, який згодом нормалізується засобами Scikit-Learn (preprocessing.normalize()). Усі подальші операції виконуються лише з нормалізованими даними.

На базі цих даних створюється ненаправлений граф у вигляді об’єкта класу Graph бібліотеки NetworkX за допомогою функції create_data_graph(). Цей граф використовується здебільшого для візуалізації структури даних і не бере участі у кластеризації.

У конструктор класу детектора передаються всі вхідні дані, які зберігаються у внутрішньому атрибуті. Також створюється порожній граф, куди під час навчання

додаються нові нейрони та зв’язки. Основними класами детектора є GNG та IGNG, що наслідують абстрактний клас NeuralGas із реалізованими базовими методами.

Метод train() багаторазово викликає алгоритм кластеризації, а також періодично виконує збереження проміжних результатів у вигляді зображень через метод __save_img(). Під час кластеризації для кожного вхідного вектора у багатовимірному просторі обираються найближчі нейрони поточного графа. Якщо наявні нейрони не відображають дані адекватно, алгоритм створює нові нейрони та встановлює відповідні зв’язки між ними.

Метод __save_img() відповідає за побудову візуалізацій. Для цього використовуються функції draw_dots3d() та draw_graph3d(), які двічі викликаються послідовно: спочатку для відображення вхідних даних у вигляді точок, а потім для накладання поверх них побудованої нейронної мережі. Візуалізація реалізується на базі бібліотеки Mayavi, зокрема за допомогою виклику mayavi.points3d().

Після завершення навчання всі збережені зображення об’єднуються у GIF-анімацію за допомогою модуля ImageIO. Це дозволяє наочно продемонструвати, як алгоритм поступово сходиться та формує кластерну структуру, яка відображає закономірності вхідних даних.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було проведено комплексне дослідження методів виявлення аномалій у телекомунікаційних мережах, а також розроблено концептуальний підхід до побудови модулів, здатних виявляти відхилення від нормальної поведінки системи у режимі реального часу.

Насамперед було розглянуто існуючі класифікації систем виявлення вторгнень (IDS), що поділяються на мережеві, хостові та системи на основі оцінки вразливостей. Було детально проаналізовано їх сильні та слабкі сторони, а також методи збору даних (пасивні, активні та змішані). Встановлено, що для підвищення ефективності сучасні IDS повинні мати комбіновану архітектуру, поєднуючи сигнатурні підходи з методами поведінкового та статистичного аналізу.

Особливу увагу приділено методам інтелектуального аналізу трафіку з використанням алгоритмів машинного навчання. Досліджено використання кластеризаційних підходів, зокрема самоорганізовуваних карт Кохонена та алгоритмів «зростаючого нейронного газу» (GNG та IGNG), які дозволяють формувати адаптивні моделі нормальної активності без необхідності наперед визначати кількість кластерів. Продемонстровано, що такі моделі можуть ефективно виявляти як відомі, так і нові аномальні сценарії, підвищуючи точність системи та зменшуючи кількість хибних спрацьовувань.

Було розроблено прототип модуля виявлення аномалій на мові Python, який використовує дані з еталонної бази NSL-KDD, а також додатково моделює активність хостів. Модуль реалізує алгоритми кластеризації та методи оцінки відхилень від середніх значень, а також візуалізацію процесу збіжності за допомогою бібліотеки Mayavi. В ході тестування було підтверджено працездатність підходу та його здатність коректно розділяти нормальні та аномальні сценарії.

У роботі також окреслено ключові труднощі: проблема навчання виключно на «чистих» даних, необхідність значного обсягу даних для формування надійної моделі, ризики перенавчання та складність інтерпретації змін у структурі кластерів.

Для їх подолання запропоновано використання адаптивного підлаштування моделей та багаторівневих механізмів контролю.

Отже, результати дипломної роботи доводять, що ефективне виявлення аномалій у телекомунікаційних мережах можливе лише за умови інтеграції кількох підходів: сигнатурних методів, поведінкового аналізу та адаптивних алгоритмів машинного навчання. Перспективним напрямом подальших досліджень є розробка гібридних IDS, здатних до самонавчання у реальному часі, інтеграція з системами візуалізації та автоматизованого реагування, а також застосування глибоких нейронних мереж для аналізу високовимірних даних у динамічно змінюваних мережевих середовищах.

Таким чином, поставлені у дипломній роботі завдання виконано: проведено аналіз існуючих методів, розроблено і реалізовано прототип модуля виявлення аномалій, а також сформульовано пропозиції щодо вдосконалення систем IDS у контексті сучасних телекомунікаційних мереж.

ПЕРЕЛІК ПОСИЛАНЬ

1. Network Traffic Anomaly Detection and Prevention: Concepts, Techniques, and Tools. Monowar H. Bhuyan, Dhruba K. Bhattacharyya, Jugal K. Kalita. Springer, 2017. – огляд методів аномалій, профайлінг, інструменти запобігання.
2. Network Anomaly Detection: A Machine Learning Perspective. Dhruba K. Bhattacharyya, Jugal K. Kalita. CRC Press, 2013. – ML-підходи до виявлення аномалій у мережевому трафіку.
3. Anomaly Detection Principles and Algorithms. Kishan G. Mehrotra, Chilukuri K. Mohan, HuaMing Huang. Springer, 2019. – базові принципи і алгоритми для різних типів аномалій.
4. Robust Regression and Outlier Detection. Peter Rousseeuw, Annick M. Leroy. Wiley, 1987. – статистичні методи для виявлення викидів та аномалій.
5. Fault, Attack, and Anomaly Detection in Telecommunications Standard Practices. ITU / IEEE Document, 2022. – офіційна документація щодо стандартів виявлення аномалій у телеком мережах.
6. Cybersecurity Monitoring and Anomaly Detection in Modern Networks. Wiley, 2024. – моніторинг, SIEM, поведінкові методи виявлення аномалій.
7. Big Data Analytics for Anomaly Detection in Communication Networks. Elsevier, 2023. – великі дані + методи виявлення аномалій, масштабованість.
8. Applied Network Security Monitoring. Chris Sanders, Jason Smith. No Starch Press, 2016. – практичні аспекти моніторингу та виявлення підозрілих подій у мережах.
9. A Comprehensive Survey on Network Anomaly Detection. G. Fernandes Jr. et al., 2019. – класифікація методів, типів аномалій, систем, відкриті проблеми.
10. Artificial Intelligence Advances in Anomaly Detection for Telecom Networks. Edozie, E. et al., 2025. – AI / DL методи, реальні випадки, виклики.
11. A Survey on Anomaly Detection in IoT: Techniques, Challenges, and Recent Trends. ZA Haider et al., 2025. – специфіка аномалій в IoT, адаптивні методи.

12. Network Anomaly Detection: A Survey and Comparative Analysis of Stochastic and Deterministic Methods. Jing Wang et al., 2013. – порівняння статистичних проти детерміністичних підходів.
13. Dynamic Network Anomaly Detection System by Using Deep Learning Techniques. Peng Lin; Kejiang Ye; Cheng-Zhong Xu, 2019. – використання глибокого навчання для часових мережевих аномалій.
14. Wireless Sensor Networks Anomaly Detection using Machine Learning: A Survey. Haque, A. et al., 2023. – методи для WSN, врахування ресурсів та обмежень.
15. Real Anomaly Detection in Telecommunication Multidimensional Data Using Data Mining Techniques. Choroś, K. et al., 2010. – аналіз багатовимірних даних, виявлення змін.
16. Anomaly Detection and Community Detection in Networks. Hadiseh Safdari, Caterina De Bacco, 2022. – використання моделей спільнот і латентних змінних для структурних аномалій.
17. EdgeCentric: Anomaly Detection in Edge-Attributed Networks. Neil Shah et al., 2015. – аномалії з урахуванням атрибутів ребер у мережевих графах.
18. SpecAE: Spectral AutoEncoder for Anomaly Detection in Attributed Networks. Yuening Li et al., 2019. – автоенкодери + спектральні методи для складних мереж.
19. Anomaly Detection on Network Traffic. Danfeng Daphne Yao et al., (chapter). 2018. – аналіз трафіку, PAYL, McPAD, SVM-підходи.
20. Anomaly Detection in Dynamic Networks of Varying Size. Timothy La Fond et al., 2014. – аномалії в динамічних мережах, змінні об’єми та часові масштаби.
21. Network Anomaly Detection: A Machine Learning Perspective. Bhattacharyya, D. K.; Kalita, J. K., 2013. – ML підходи, ознаки, оцінка.
22. Anomaly Detection Using Neural Networks: A Review. Author Unknown, 2020. – огляд NN-методів: RNN, LSTM, CNN, AE.
23. Anomaly Detection in Network Logs via Time-Series Analysis. 2022. – виявлення аномалій за часовими рядами, лог-аналіз.
24. Unsupervised Anomaly Detection for Network Security: Clustering and Density-based Methods. 2021. – методи без навчання, DBSCAN, LOF тощо.

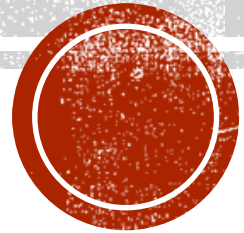
25. Hybrid Approaches to Network Anomaly Detection Combining Statistical and ML Methods. 2023. – комбіновані підходи, мінімізація хибнопозитивів.
26. Online Anomaly Detection in Streaming Telecom Data. 2024. – методи для потокових даних, real-time детекція.
27. GAN-based Anomaly Detection in Telecom Traffic. 2023. – використання генеративних моделей для виявлення незвичних патернів.
28. Anomaly Detection in 5G Networks: Use of Deep Learning to Detect Unusual Traffic Patterns. 2022. – патерни аномального трафіку у 5G, перевантаження, латентність.
29. Evaluating Datasets for Network Anomaly Detection: Challenges and Best Practices. 2020. – огляд наборів даних, проблема imbalance, репрезентативність.
30. Comparative Study of Feature Selection Techniques for Anomaly Detection in Telecom Networks. 2024. – які ознаки краще підходять, зменшення розмірності, інформативність ознак.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ТЕЛЕКОМУНІКАЦІЙ

КВАЛІФІКАЦІЙНА РОБОТА

МЕТОДИ ВИЯВЛЕННЯ АНОМАЛІЙ У ТЕЛЕКОМУНІКАЦІЙНИХ МЕРЕЖАХ



Виконав:
студент групи ТСДМ-63
Євгеній СОРОКА

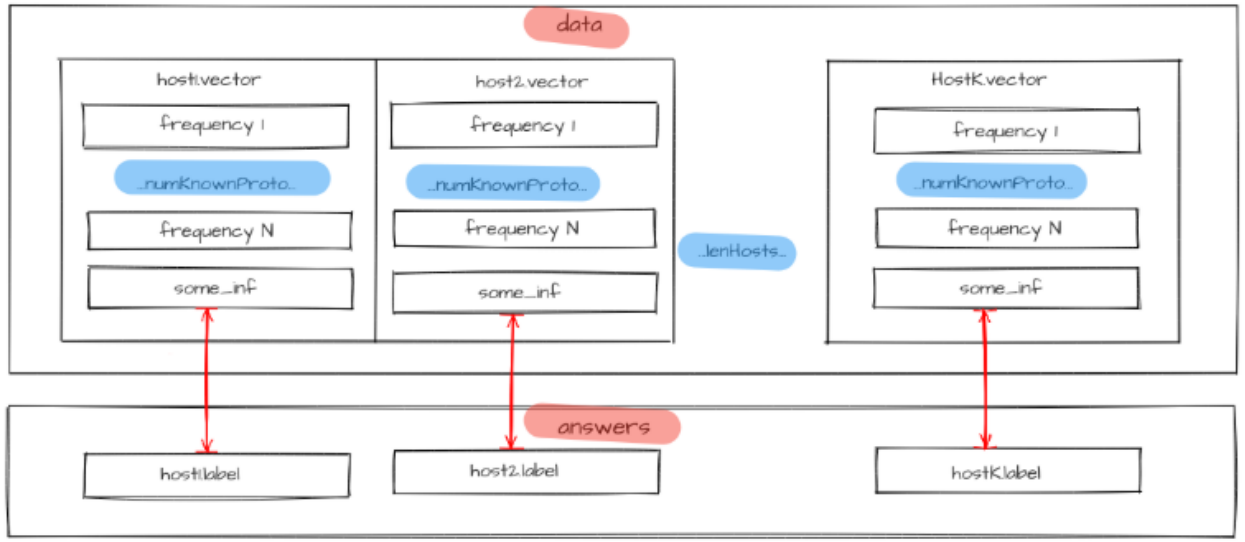
МЕТА РОБОТИ ТА АКТУАЛЬНІСТЬ

- Мета роботи – розроблення та дослідження методу автоматизованого виявлення аномальної активності у телекомунікаційних мережах на основі аналізу мережевого трафіку з використанням алгоритмів машинного навчання, що забезпечують підвищення точності класифікації мережевих вузлів і своєчасне виявлення кіберзагроз у реальному часі.
- Об’єкт дослідження – процеси передавання та аналізу даних у телекомунікаційних мережах, що впливають на їхню безпеку та стабільність функціонування.
- Предмет дослідження – методи та алгоритми машинного навчання і кластеризації, призначені для виявлення аномалій і класифікації вузлів телекомунікаційних мереж за їхнім профілем трафіку.

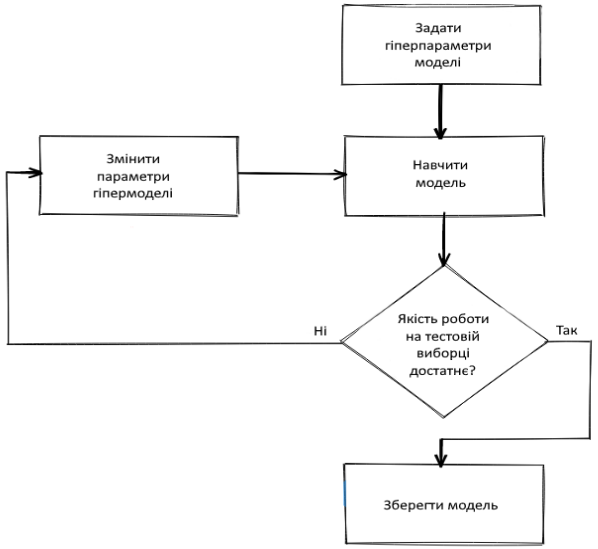
Актуальність.

- Стрімкий розвиток телекомунікаційних мереж, збільшення кількості підключених пристроїв та зростання обсягів переданих даних супроводжуються різким підвищенням рівня кіберзагроз. Традиційні сигнатурні системи захисту виявляються малоефективними проти нових або модифікованих атак, що не мають опису у базах даних. В умовах цифровізації промислових, транспортних і державних інфраструктур зростає потреба у розумних системах моніторингу, здатних автоматично виявляти аномалії у мережевому трафіку без активного втручання в роботу мережі. Застосування методів машинного навчання та кластеризації (SOM, GNG, DBSCAN, AutoML) є перспективним напрямом, що дозволяє створювати адаптивні системи виявлення загроз у реальному часі.

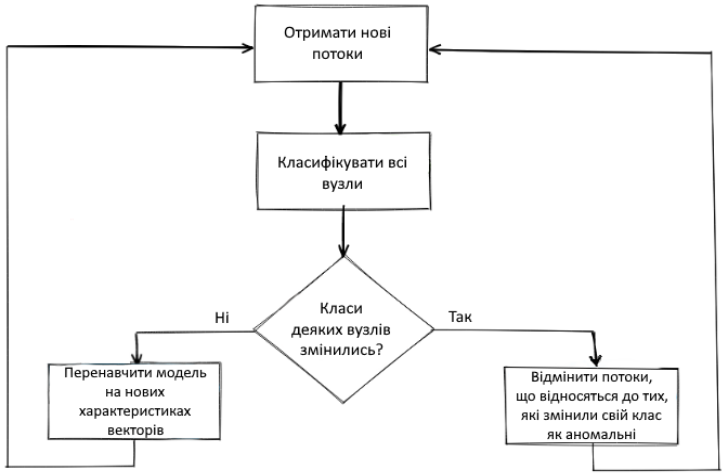
МАТЕМАТИЧНА МОДЕЛЬ



Дані для навчання моделі



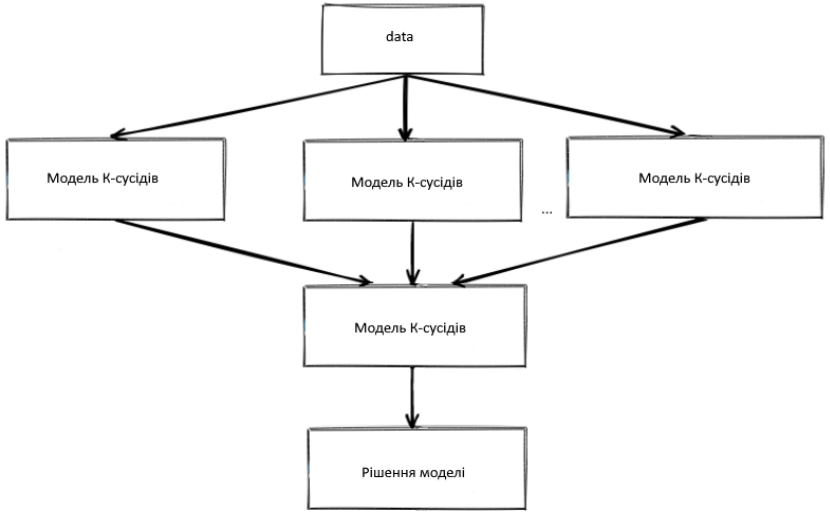
Процес створення моделі



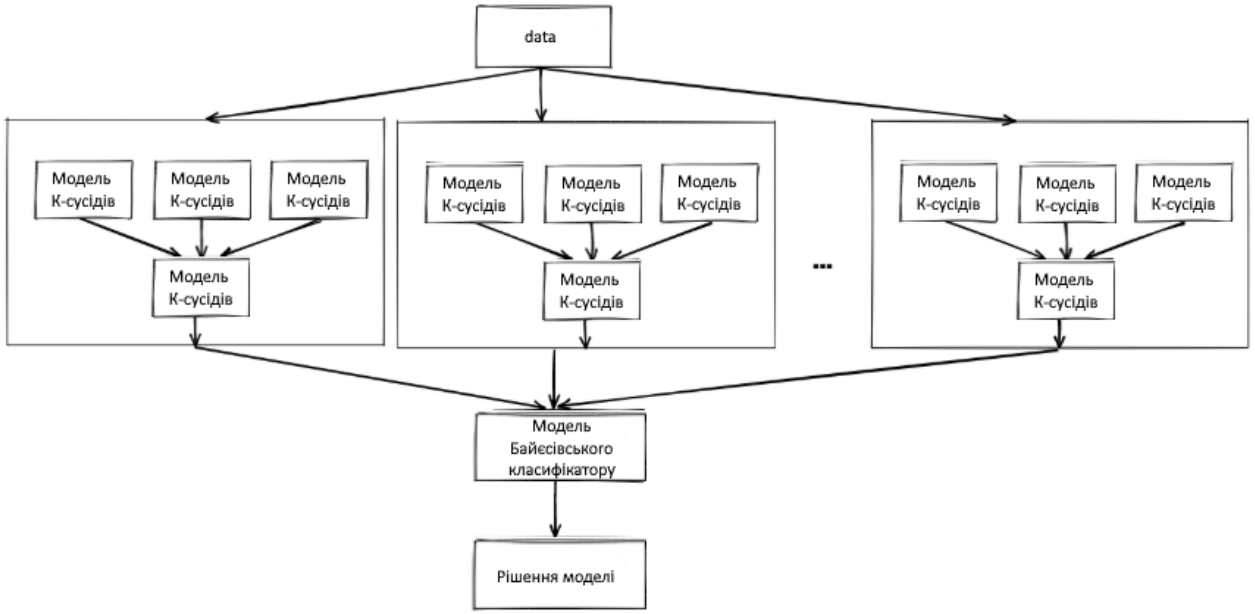
Виявлення аномальних потоків

- Модель класифікує вузли за мережевим трафіком. Потоки (**Flow**) описують пакети між джерелом і приймачем з однаковим протокольним стеком, а набір потоків вузла (**Flows**) формує вектор характеристик (**host.vector**), що включає частоту протоколів, метадані та зв'язки з іншими вузлами.
- Для навчання використовуються розмічені дані (**host.labels**), поділені на тренувальну та тестову вибірки. Модель навчається, оцінюється метриками (**Accuracy, Precision, Recall, F1-score**) і коригується за потреби.
- Аномалії визначаються зміною кластерної мітки вузла (**host.label ≠ host.new_label**) та оцінюються через ієрархічну структуру. Модель автоматично класифікує вузли, виявляє відхилення та адаптується до змін мережі.

РЕАЛІЗАЦІЯ МОДЕЛІ ДЕТЕКЦІЇ



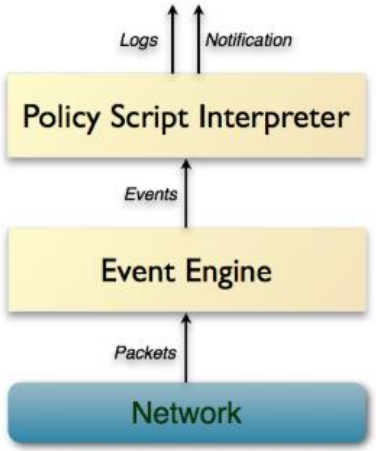
Перший блок моделі



Підсумкова модель

- Мережевий трафік оброблявся через **DPI** CyberLympha DATAPK, що дозволяє виділяти протокольні потоки. Характеристичні вектори вузлів генерувалися Python-скриптом для автоматизації та уніфікації даних.
- Оптимізація моделей проводилась через **AutoML (TPOT)**, сформовано ансамбль: перший блок – KNN-класифікатори, другий – байєсівський класифікатор. Стекінг підвищує точність і стійкість моделі.
- Тестування на незалежних даних показало **F1 ≈ 80 %**, а використання механізму альтернативних класів підвищило **F1 до 95–98 %**.
- Поєднання DPI, AutoML і ансамблевих методів забезпечує ефективну і масштабовану систему для ідентифікації вузлів та виявлення аномалій у мережевому трафіку.

ТИПИ IDS, ДІЮЧІ СИСТЕМИ IDS



Приклад роботи ядра Bro

- Для спрощення аналізу системи варто звести до основних типів:

Типи 0–2 – IDS вузлового рівня: сигнатурні, поведінкові та гібридні; схожі механізми реалізують антивірусні програми.
Типи 3–5 – сканери вразливостей вузлового рівня, часто інтегровані в антивірусні пакети.
Типи 6–8 – гібридні IDS вузлового рівня, що поєднують пасивний моніторинг і активне сканування.
Типи 9–11 – IDS мережевого рівня, що працюють сигнатурно, на аномаліях або гібридно.
Типи 12–14 – мережеві сканери вразливостей, здатні виявляти потенційні точки проникнення.
Типи 15–17 – гібридні IDS мережевого рівня для комплексного захисту мережі.
Тип 26 – гібридна система на основі агентів, що забезпечує масштабованість та інтеграцію кількох функцій.

- **Honeypot** – вузли-пастки для відстеження атак, збору даних, навчання поведінкових систем і нейромереж, від простих сервісів до складних honeynet.
- **Діючі системи IDS:**
- **Snort** – сигнатурна мережна IDS, розширювана через власні модулі, може працювати гібридно.
- **Suricata** – сумісна зі Snort, підтримує аналіз прикладних протоколів, Lua-скрипти, IP reputation, комплексний моніторинг.
- **Bro/Zeek** – платформа гібридного аналізу на транспортному, мережевому та прикладному рівнях; обробляє події, аналізує аномалії та використовується для моніторингу і діагностики мережі.

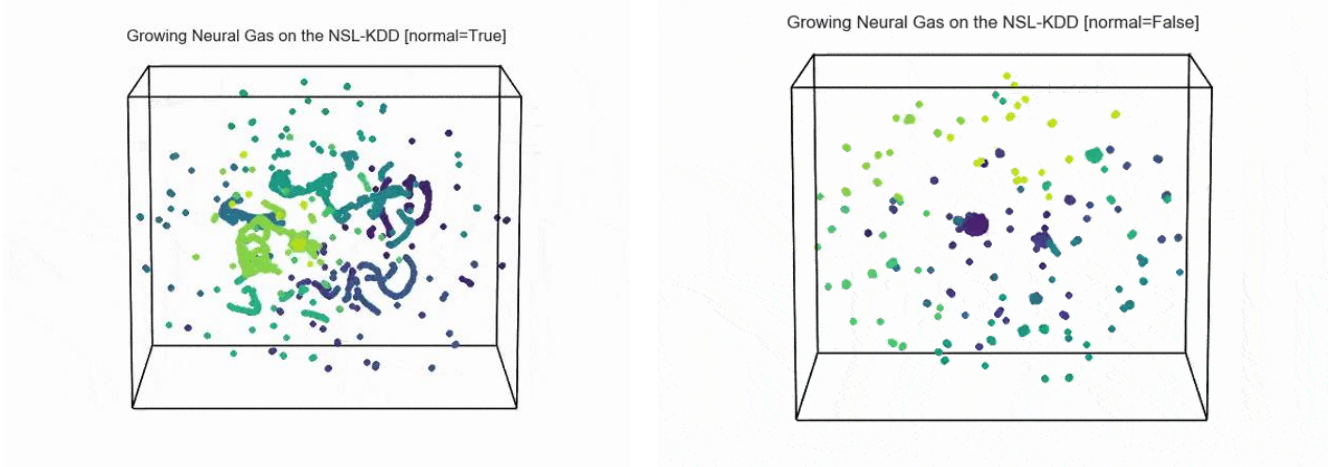
ОПИС ТИПОВОЇ АРХІТЕКТУРИ IDS



Структура IDS

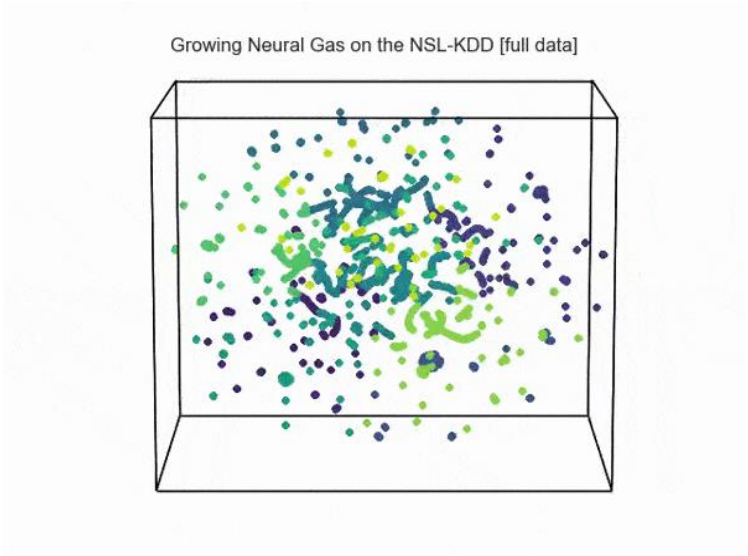
- Система виявлення вторгнень (IDS) — це комплекс програмних і апаратних засобів, що забезпечує виявлення несанкціонованої активності в мережі або на хостах. Її архітектура складається з трьох основних частин: **датчиків, підсистеми аналізу** та **підсистеми реагування**.
- Датчики збирають дані з різних рівнів — прикладного, хостового та мережевого. Підсистема аналізу обробляє інформацію, використовуючи сигнатурний і поведінковий аналіз, а також алгоритми машинного навчання. Підсистема реагування відображає результати, генерує сповіщення й автоматично блокує підозрілу активність, забезпечуючи комплексний захист мережевої інфраструктури.

ПОБУДОВА НЕЙРОМЕРЕЖЕВОГО АНАЛІЗАТОРУ



Нормальна активність в мережі

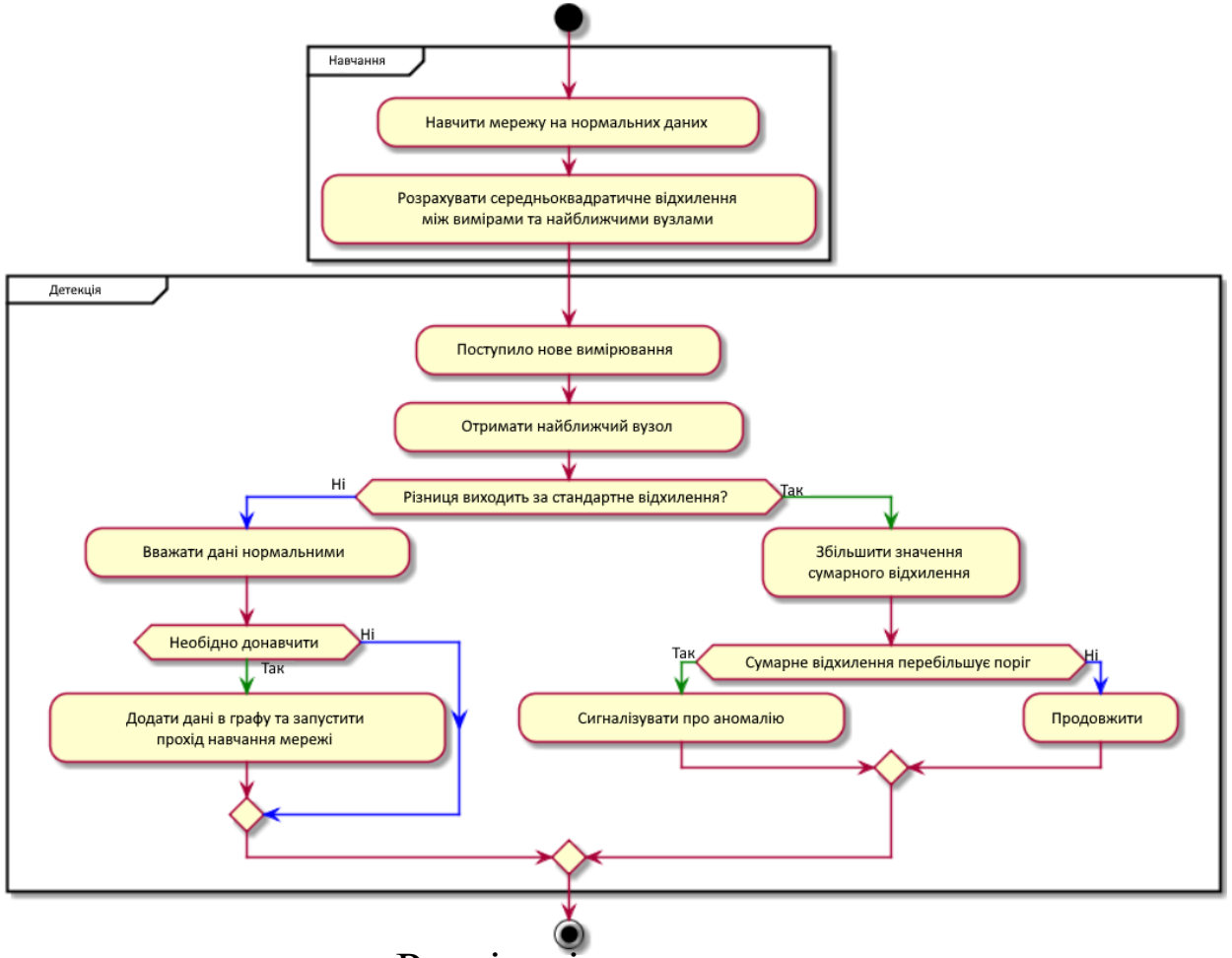
Аномальна активність в мережі



Накладення аномальної поведінки в мережі, при взломі, на нормальну

- Найефективнішими є комбіновані системи виявлення вторгнень, які поєднують сигнатурні, поведінкові та машинні методи. Такий підхід дозволяє виявляти як відомі, так і нові атаки. У дослідженні створено концептуальний модуль виявлення аномалій, що працює з даними NSL-KDD у двокласовому режимі («аномалія/норма»). Використано нейромережеві алгоритми кластеризації, зокрема самоорганізовувані карти Кохонена (SOM) і Growing Neural Gas (GNG), які формують еталонну модель нормальної поведінки. Адаптивне оновлення кластерів дозволяє системі реагувати на зміни мережі та знижує кількість хибних спрацьовувань.

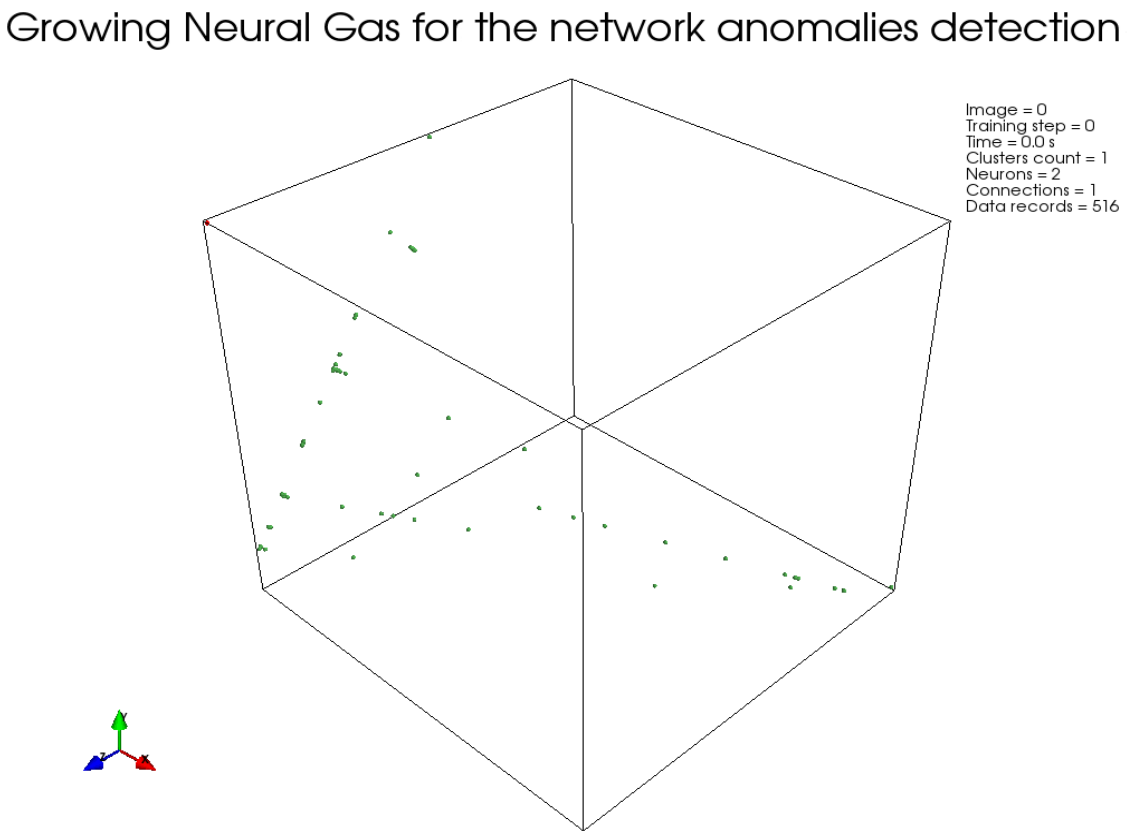
АРХІТЕКТУРА МОДУЛЯ



Реалізація алгоритму

- У роботі розглянуто процес збору та обробки даних у системах виявлення вторгнень, де агенти передають показники з різних рівнів інфраструктури до центральної системи. На основі даних NSL-KDD формуються вектори характеристик вузлів і трафіку, що створюють єдиний профіль поведінки мережі. Для класифікації використано гібридний метод, який поєднує оцінку середнього відхилення та мінімальної відстані між нейронами. Ядром системи є алгоритми Growing Neural Gas (GNG) і Incremental GNG, які динамічно адаптуються до змін трафіку та ефективно виявляють аномалії у реальному часі.

РОБОТА МОДУЛЯ ТА ІНТЕРПРИТАЦІЯ РЕЗУЛЬТАТІВ



Візуалізація роботи модуля

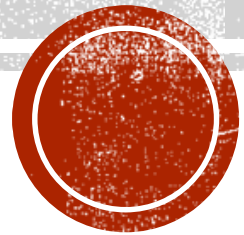
- Розроблений модуль виконує автоматичне виявлення аномалій у телекомунікаційних мережах, зберігаючи результати навчання у вигляді нейронної структури кластерів. Він отримує нові дані від IDS-агентів, оновлює модель і формує висновки про наявність або тип аномалії (наприклад, DoS чи сканування портів). Реалізацію виконано мовою Python з використанням алгоритмів Growing Neural Gas (GNG) та Incremental GNG, бібліотек NumPy, NetworkX і Mayavi для візуалізації. Модуль забезпечує безперервний моніторинг, адаптацію моделі під час експлуатації та наочне представлення процесу кластеризації у вигляді графічних структур і GIF-анімацій.

ВИСНОВКИ

- У кваліфікаційній роботі проведено дослідження методів виявлення аномалій у телекомунікаційних мережах і розроблено підхід до створення модулів, здатних виявляти відхилення від норми в реальному часі.
- Проаналізовано класифікації IDS, методи збору даних та поєднання сигнатурних і поведінкових підходів.
- Особливу увагу приділено машинному навчанню, зокрема алгоритмам кластеризації (SOM, GNG).
- Створено прототип модуля Python, що підтвердив ефективність виявлення аномалій.
- Робота доводить доцільність інтеграції сигнатурних і адаптивних методів, а також перспективність гібридних IDS, здатних до самонавчання у реальному часі.



ДЯКУЮ ЗА УВАГУ!



Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Євгеній СОРОКА

Назва: МЕТОДИ ВИЯВЛЕННЯ АНОМАЛІЙ У ТЕЛЕКОМУНІКАЦІЙНИХ МЕРЕЖАХ

Координатор: Вікторія КОРЕЦЬКА

Підрозділ: ННІТ

Коефіцієнт подібності 1:1.97

Коефіцієнт подібності 2:1.84

Тривога: 0

Після аналізу Звіту подібності констатую наступне:

☐ виявлені в роботі запозичення є сумлінними і не мають ознак плагіату. Тому робота визнається самостійною і допускається до захисту;

☐ виявлені в роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і самостійності її автора. Роботу направити на доопрацювання;

☐ виявлені в роботі запозичення є недобросовісними і мають ознаки плагіату або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень. У зв'язку з чим, робота не допускається до захисту.

…… ……

Дата *Підпис Наукового керівника*