МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

ФЕДОРИШИН ОЛЕКСАНДР ОЛЕГОВИЧ

Допускається до захисту:
В. о. завідувача кафедри
прикладної математики та
кібербезпеки, доктор
філософії з математики,
_____ Луценко А.В.
«__»_____20__ р.

РЕАЛІЗАЦІЯ КОНЦЕПЦІЯ ІМУНІТЕТУ ДЛЯ ЗАБЕЗПЕЧЕННЯ КІБЕРСТІЙКОСТІ ВУЗЛІВ ІНТЕРНЕТУ РЕЧЕЙ

Спеціальність 105 Прикладна фізика та наноматеріали

Кваліфікаційна (магістерська) робота

Науковий керівник:
В.Г. Крижановский
д-р техн. наук, професор,
професор кафедри прикладної
математики та кібербезпеки
_____
(підпис)

Оцінка:___/___/_____
(бали за шкалою ЄКТС/за національною шкалою)
Голова ЕК:_____
(підпис)

Вінниця - 2025

ЗМІСТ

ВСТУП

Інтернет речей (Internet of Things, IoT) сьогодні стрімко трансформує цифровий ландшафт, поєднуючи величезну кількість пристроїв і сенсорів у єдині мережі. Це відкриває нові можливості для збору даних, автоматизації та підвищення ефективності у різних сферах – від промислової автоматизації до “розумних” міст і медицини. Водночас стрімке розгортання IoT-інфраструктури породжує безпрецедентні виклики кібербезпеки. Різнорідність пристроїв (від потужних серверів до мікроконтролерів), їхня повсюдна підключеність та робота в неконтрольованих середовищах призводять до численних вразливостей та ризиків. Багато IoT-пристроїв спроєктовані з мінімальними засобами захисту через обмежені ресурси чи вартісні обмеження, що робить їх легкою мішенню для зловмисників. Наслідки компрометації таких вузлів можуть бути дуже серйозними – від несанкціонованого доступу до персональних даних користувачів до масштабних збоїв критично важливої інфраструктури (наприклад, у сферах енергетики чи охорони здоров’я).

Незахищені IoT-пристрої вже сьогодні активно експлуатуються зловмисниками для різноманітних цілей – крадіжки даних, несанкціонованого стеження, а також організації масованих DDoS-атак із використанням ботнетів на основі IoT. Серед найпоширеніших загроз, що актуальні для середовища IoT, дослідження відзначають розподілені атаки відмови в обслуговуванні (DDoS), атаки типу “Man-in-the-Middle” (перехоплення і підміна даних), несанкціонований доступ до вузлів, підміну пристроїв (spoofing), радіочастотне глушення (jamming), фальсифікацію або маніпуляцію даними, а також шкідливі програми-здирники та ботнет-мережі. Така різноманітність векторів атак вимагає комплексних підходів до забезпечення безпеки IoT-середовищ.

Однак традиційні засоби кіберзахисту, розроблені для класичних IT-систем, не завжди ефективні в контексті IoT. Централізовані підходи до

контролю безпеки важко масштабувати на децентралізовані мережі з тисячами вузлів, а використання ресурсомістких алгоритмів часто неможливе на пристроях зі скромними обчислювальними можливостями. Це обумовлює потребу в нових адаптивних стратегіях, здатних забезпечити своєрідний «цифровий імунітет» IoT-інфраструктури – тобто здатність системи автоматично виявляти і нейтралізувати кіберзагрози, аналогічно до того, як біологічна імунна система захищає організм від патогенів.

У цій роботі розглянуто процес розробки системи виявлення аномалій в мережах IoT на основі методів машинного навчання, що наслідують принципи штучного імунітету.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ЗАСАДИ ЦИФРОВОГО ІМУНІТЕТУ В IoT

1.1 Загрози кібербезпеці в Інтернеті речей

Масове впровадження IoT-технологій супроводжується появою специфічних кіберзагроз, зумовлених особливостями цих систем. На відміну від традиційних мереж, IoT-середовища вирізняються гетерогенністю пристроїв, серед яких багато слабко захищених або за замовчуванням довірливих компонентів (камери спостереження, датчики, побутова техніка тощо). Багато з них працюють на спрощених версіях ОС, з рідко оновлюваним прошиванням і типовими паролями, що полегшує компрометацію. Децентралізований характер та динаміка топології IoT-мереж (пристрої можуть часто приєднуватись і залишати мережу) ускладнюють застосування статичних політик безпеки, які ефективні в традиційних фіксованих мережах. Окрім того, бездротові канали зв'язку, широко використовувані IoT-пристроями, відкривають додаткові вектори атак – радіоперехоплення, глушіння сигналу, підробка пакетів тощо.[1], [2], [3], [20]

У результаті IoT-інфраструктура стає привабливою ціллю для зловмисників різного штибу – від кіберзлочинців до недержавних груп, зацікавлених у виведенні з ладу критичних сервісів. Серед типових атак на IoT відзначають:

- **Ботнет-атаки та DDoS**. Нестача захисту на багатьох IoT-пристроях призвела до появи ботнетів (таких як Mirai), що об'єднують тисячі заражених камер, маршрутизаторів тощо для здійснення розподілених атак відмови в обслуговуванні. Такі атаки здатні генерувати гігантський трафік і виводити з ладу онлайн-сервіси.

- **Атаки "людина посередині"**. Зловмисник, перехопивши незашифрований трафік IoT-пристрою, може аналізувати чи модифікувати передані дані. Це особливо небезпечно у випадках, коли

IoT використовується в медицині або промисловості – перехоплення або підміна команд може призвести до аварій.

- **Несанкціонований доступ та захоплення вузлів**. Якщо зловмиснику вдається дістати контроль над IoT-пристроєм (через експлойт в прошивці чи вгаданий пароль), пристрій може бути використаний у подальших атаках або для шпигунства. Такі компрометовані вузли зсередини мережі можуть ініціювати атаки на інші компоненти (внутрішні атаки), що важко виявити.

- **Атаки на цілісність і конфіденційність даних**. IoT-сенсори часто збирають чутливі дані (відео, аудіо, показники здоров'я тощо). Їхнє викрадення (data theft) або підробка може мати значні негативні наслідки для приватності та безпеки користувачів. Наприклад, зчитування показників розумного лічильника може розкрити режим дня мешканців будинку.

- **Фізичні атаки та виведення з ладу**. Здебільшого IoT-пристрої розташовані в безпосередньому фізичному доступі (наприклад, вуличні сенсори), що дозволяє саботаж (відключення живлення, пошкодження) або підміну пристрою на компрометований. [1], [3]

Окрім перелічених загроз, існують і інші, такі як атаки на протоколи зв'язку (перехоплення та розшифровка трафіку MQTT/CoAP), атаки типу відмови обслуговування шляхом радіоперешкод, зловмисні оновлення прошивки тощо. Усі ці загрози свідчать про критичну потребу в стійких механізмах кіберзахисту для IoT. Концепція цифрового імунітету передбачає побудову таких систем захисту, що здатні автономно, у реальному часі розпізнавати вторгнення або аномальну поведінку пристроїв і приймати заходи протидії (сповіщення, ізоляція вузла, блокування трафіку тощо) без постійного втручання людини. Формування "цифрового імунітету" базується на поєднанні різних технологій: від традиційних систем виявлення вторгнень до

штучного інтелекту та біонічних підходів, наслідуваних у природи (зокрема принципів роботи імунної системи). [1], [3], [4], [20]

1.2 Традиційні підходи (IDS/IPS) та їх обмеження в IoT

Системи виявлення та запобігання вторгненням (IDS/IPS) є одними з класичних інструментів кібербезпеки, які широко застосовуються для захисту мереж та хостів. IDS (Intrusion Detection System) призначені для моніторингу трафіку або подій у системі з метою виявлення ознак відомих атак чи аномальної активності, тоді як IPS (Intrusion Prevention System) не лише виявляють, а й автоматично блокують підозрілі дії. В контексті IoT ці підходи стикаються з низкою труднощів. [5], [6], [23]

Традиційно IDS поділяються на два основні типи: сигнатурні (на основі шаблонів атак) та аномалійні (на основі відхилення від норми поведінки). Сигнатурні IDS (SIDS) працюють за принципом порівняння вхідних подій з базою відомих сигнатур атак (характерних шаблонів, сигнатур експлойтів тощо). При збігу з сигнатурою система формує тривогу. Цей підхід добре зарекомендував себе для відомих атак, оскільки мінімізує хибні спрацювання і дає детальну інформацію про загрозу. Проте основний його недолік – нездатність розпізнати нові, невідомі атаки, для яких ще не створено сигнатури. Крім того, підтримка актуальної бази сигнатур вимагає регулярних оновлень і тягне значні накладні витрати. [5], [6], [23]

Аномалійні IDS (AIDS), натомість, будують модель нормально поведінки системи і сигналізують про будь-які значні відхилення. В IoT-середовищах саме аномалійний підхід вважається перспективнішим, оскільки дозволяє виявляти раніше невідомі загрози, що особливо важливо з огляду на динамічно змінюваний і непередбачуваний характер атак на IoT. Водночас впровадження AIDS натрапляє на низку викликів у IoT: по-перше, значні обсяги генерованих даних і велика кількість вузлів створюють проблему масштабованості – система може продукувати занадто багато спрацювань або

пропускати атаки за умов перевантаження (зростає кількість помилкових спрацювань та пропущених інцидентів. По-друге, обмежені обчислювальні ресурси та енергоможливості типових IoT-пристроїв ускладнюють реалізацію складних алгоритмів аналізу трафіку на самому вузлі. Часто IoT-пристрій просто не в змозі виконувати постійний моніторинг з використанням традиційних IDS-рушіїв (наприклад, через нестачу пам'яті для бази сигнатур або обчислювальної потужності для аналізу кожного пакету).

Зазначені фактори призводять до того, що класичні IDS/IPS-рішення потребують адаптації для IoT. Деякі дослідники пропонують переносити функції аналізу на рівень мережевих шлюзів або хмарних сервісів, залишаючи на пристроях тільки легковагі агенти, які збирають дані. Проте такий підхід підвищує затримки та може створювати "вузькі місця". Інший напрям – спрощення та оптимізація алгоритмів IDS спеціально під обмеження IoT. Наприклад, спрощені сигнатурні NIDS для IoT (мережеві IDS) використовували Suricata або Snort з урізаними базами правил для виявлення DoS-атак у 6LoWPAN мережах. Однак навіть у цьому випадку застосування традиційних сигнатурних методів на мікропристроях практично неможливе без їх повної перебудови під конкретні ресурси IoT. Наукові експерименти показали, що звичайні NSA/CSA алгоритми на повнорозмірних даних мережевого трафіку мають серйозні проблеми масштабованості та продуктивності на IoT-пристроях зі слабким "залізом". [5], [7], [8], [23]

З огляду на вищезазначене, у сфері безпеки IoT спостерігається перехід від суто класичних IDS до гібридних та інтелектуальних систем, які поєднують декілька підходів. Наприклад, гібридні IDS комбінують сигнатурний і аномалійний метод: спочатку швидко відфільтровують відомі загрози за сигнатурами, а далі більш ретельно аналізують нові потенційні аномалії. Такий підхід дозволяє компенсувати недоліки кожного методу і знизити навантаження на систему. Важливу роль починають відігравати машинне навчання і штучний інтелект у IDS – вони можуть виявляти складні патерни

атак, що не описуються простими правилами. Проте впровадження ML у IoT теж нелегке завдання (через нестачу даних, проблему дисбалансу класів та ін.) – ці аспекти детально розглянуто в наступному підрозділі. [4], [5], [6]

Отже, традиційні IDS/IPS є необхідною складовою кіберзахисту, але в контексті IoT їх потрібно переглядати і вдосконалювати. Недостатня ефективність "з коробки" стимулює дослідників шукати нові архітектури систем захисту IoT, що призводить до появи концепцій на кшталт цифрового імунітету – коли система здатна адаптивно навчатися розпізнавати загрози та протидіяти їм, виходячи за рамки статичних сигнатур або жорстких норм поведінки. [4], [5]

1.3 Методи машинного навчання у виявленні аномалій (Autoencoder, VAE, FL тощо)

Виявлення аномалій на основі машинного навчання (ML) стає ключовим елементом побудови "цифрового імунітету" IoT, адже ці методи дозволяють автоматично "навчатися" нормальній поведінці системи та виявляти відхилення, включно з новими видами атак. Нижче розглянемо декілька сучасних ML-підходів, що застосовуються в задачах виявлення аномалій: автоенкодери та їхні варіації (в тому числі варіаційні автоенкодери, VAE), а також федеративне навчання (FL) для розподіленого виявлення вторгнень. [4], [9], [21]

Автоенкодер (AE) – це різновид нейронної мережі, яка навчається відновлювати свій вхід на виході, стискаючи його у прихованому шарі. Інтуїтивно, добре натренований автоенкодер буде якісно відновлювати типові (нормальні) дані, тоді як для нетипових (аномальних) даних похибка відновлення (reconstruction error) суттєво зросте. Ця властивість лежить в основі використання автоенкодерів для аномалійного виявлення: спочатку модель навчають тільки на нормальних даних, після чого вона здатна "помічати" аномалії як такі приклади, що погано відновлюються і дають

велику помилку. Подібний одно-класовий підхід до навчання дозволяє обійти проблему нестачі мічених атак: модель фокусується на паттернах нормально поведінки і визначає відхилення без потреби у великих вибірках зловмисних даних. Як зазначають дослідники, це критично важливо для мереж IoT, де аномальні події рідкісні та різноманітні. [4], [10], [11]

Автоенкодери успішно застосовувалися у різних сценаріях. Зокрема, один із підходів продемонстрував ефективність поєднання асиметричного глибокого автоенкодера та однокласового класифікатора на основі глибокої нейромережі, що дозволило досягти високого рівня виявлення (детекції) – близько 96% – при мінімальній кількості хибних спрацювань. При цьому модель навчалася виключно на легітимному трафіку і виявляла атаки через перевищення порогу помилки реконструкції. Інше дослідження з використанням глибокого автоенкодера для аналізу мережевих даних (наприклад, NSL-KDD) показало, що автоенкодерний підхід може забезпечити ~85% точності у бінарній класифікації “норма/атака” та близько 92% при багатокласовій класифікації типів трафіку. Ці результати підтверджують, що автоенкодери є дієвим інструментом для побудови IDS нового покоління, здатних вловлювати нові загрози. Водночас класичні автоенкодери мають і недолік: якщо модель надто потужна, існує ризик “перепідгонки”, за якої автоенкодер навчиться реконструювати навіть нетипові вхідні дані. Таким чином, аномалії можуть давати невелику похибку і лишатися непоміченими. Для пом’якшення цієї проблеми пропонуються покращені підходи – зокрема варіаційні автоенкодери та генеративно-змагальні мережі (GAN), які вводять стохастичність чи змагальність у процес навчання і змушують модель краще моделювати розподіл нормальних даних. [11], [12], [10], [4]

Варіаційний автоенкодер (VAE) – це імовірнісна версія автоенкодера, яка навчється не просто відображенню “вхід-вихід”, а оцінює розподіл імовірності латентних змінних і даних. За рахунок цього VAE краще узагальнює та менш схильний до відтворення аномалій. VAE отримав

застосування в задачах пошуку викидів: наприклад, моделі на основі VAE успішно використовувались для мережевого моніторингу, де аномалії визначаються як зразки з вкрай малою імовірністю під побудованим розподілом нормального трафіку. Аналогічно до автоенкодера, VAE можна тренувати на нормальних даних, проте на етапі детекції оцінювати не тільки похибку реконструкції, а й правдоподібність згенерованого VAE представлення. В літературі також зустрічаються підходи, що комбінують VAE з іншими методами – наприклад, зі самоорганізованими картою (SOM) для покращення розрізнення кластерів нормального та аномального трафіку. В цілому, VAE дозволяють підвищити чутливість системи до аномалій, яку могли б “замаскуватися” під норму для звичайного автоенкодера. [4], [21]

Ще одним важливим напрямом є федеративне навчання (Federated Learning, FL) у контексті IoT-безпеки. Класичні моделі машинного навчання зазвичай передбачають централізований збір та навчання на даних (наприклад, з усіх пристроїв в хмарі). Проте такий підхід викликає серйозні занепокоєння щодо приватності (передача сирих даних з пристроїв може порушувати конфіденційність) і масштабованості (централізований сервер стає перевантаженим вузьким місцем). Федеративне навчання пропонує протилежну ідею: навчання моделі відбувається розподілено на самих пристроях, які передають серверу лише оновлені параметри моделі, а не самі дані. Сервер агрегує ці локальні оновлення (середньозважує) і надсилає оновлену глобальну модель назад клієнтам. Таким чином, *дані ніколи не покидають пристроїв*, що значно знижує ризик їх витоку або несанкціонованого доступу. Застосування FL для задач виявлення аномалій в IoT вже продемонструвало обнадійливі результати. Дослідження повідомляють, що FL-IDS перевершують традиційні централізовані ML-IDS за точністю виявлення, оскільки залучають більшу розмаїтість даних (з різних вузлів) без компрометації приватності, а також краще масштабуються на великі мережі. Наприклад, у 2025 році було запропоновано FL-підхід до IDS

для MQTT-мереж, що зміг досягти ~95,6% точності виявлення атак, перевершивши централізовані аналоги майже на 5%. [9], [13], [14], [21]

Проте, FL теж має свої виклики: зокрема, при збільшенні числа клієнтів продуктивність моделі може падати через гетерогенність даних і високі комунікаційні витрати на передачу параметрів. Також існують ризики цілеспрямованих атак на сам процес федеративного навчання (атаки отруєння моделі). Незважаючи на це, у контексті цифрового імунітету концепція FL є надзвичайно цінною, адже дозволяє реалізувати розподілену "імунну відповідь": кожен вузол локально вчиться розпізнавати аномалії, роблячи всю мережу більш стійкою без концентрації даних в одному місці. [9], [13], [14]

Отже, методи ML – автоенкодери (і їхні похідні) та федеративне навчання – надають потужний інструментарій для реалізації адаптивного захисту IoT. На відміну від жорстко запрограмованих правил, ML-моделі можуть підлаштовуватися під нові загрози, виявляти раніше невідомі атаки та робити це в розподілений спосіб, що резонує з біологічними аналогами імунної системи. Наступний підрозділ присвячено безпосередньо штучним імунним системам – як біологічні принципи і алгоритми імунітету можуть бути використані для побудови цифрового імунітету в IoT. [4], [6], [9]

1.4 Штучні імунні системи (AIS): принципи та алгоритми (NSA, CSA, DCA, пам'ять)

Штучні імунні системи (Artificial Immune Systems, AIS) – це напрям біо-натхненного штучного інтелекту, що моделює принципи роботи біологічної імунної системи для вирішення прикладних завдань, у тому числі задач кібербезпеки. Імунна система людини – надзвичайно складна адаптивна система, здатна розпізнавати та знешкоджувати величезну різноманітність чужорідних агентів (патогенів), при цьому навчаючись протягом життя і зберігаючи імунну пам'ять про перенесені загрози. Ці властивості надихнули дослідників на створення штучних моделей, що імітують імунні механізми для

захисту комп’ютерних систем. Роботи Д. Фармера, С. Берсіні та Ст. Форрест в 1990-х заклали основи AIS, показавши як принципи розрізнення “свого/чужого” та пам’яті імунної системи можуть бути застосовані для виявлення вірусів і вторгнень у комп’ютерні мережі. [8], [15], [16], [22]

В межах AIS виокремилось кілька базових парадигм (алгоритмів), на яких зосереджено більшість досліджень. Згідно з оглядом Dasgupta et al. (2011), сучасні AIS будуються переважно на чотирьох ключових алгоритмах: (1) алгоритм негативної селекції (Negative Selection Algorithm, NSA); (2) алгоритм штучних імунних мереж (Artificial Immune Network, AIN або AIRS); (3) алгоритм клонального відбору (Clonal Selection Algorithm, CSA); (4) алгоритми на основі теорії небезпеки (Danger Theory) і дендритних клітин (Dendritic Cell Algorithm, DCA). Нижче коротко розглянемо принципи перших, найбільш застосовуваних трьох підходів (NSA, CSA, DCA) і поняття імунної пам’яті в контексті AIS. [8], [15], [22]

Алгоритм негативної селекції (Negative Selection, NSA) – натхненний процесом дозрівання Т-лімфоцитів у тимусі людини. Біологічна суть полягає в тому, що молоді імунні клітини, які реагують на власні (“self”) антигени організму, знищуються (негативна селекція), аби в кровотоці залишились тільки ті лімфоцити, що розпізнають чужорідне (“non-self”). В AIS це змодельовано наступним чином: генерується безліч детекторів (абстракція імунних клітин), і на етапі “навчання” вони перевіряються на відповідність до набору нормальних (“self”) даних системи; будь-які детектори, що матчаться з нормальними патернами, відкидаються. На виході формується підмножина детекторів, що не перетинаються з нормальним профілем, тобто потенційно реагують лише на аномальні (чужорідні) дані. На етапі моніторингу ці детектори застосовуються для перевірки нових об’єктів: якщо якийсь детектор спрацьовує (матчиться з об’єктом), об’єкт вважається аномальним. Таким чином NSA забезпечує реалізацію принципу “сам/не сам” у штучній системі. Першим це застосував Ст. Форрест зі співавт. для виявлення комп’ютерних

вірусів: вони згенерували безліч рядків-детекторів, що не збігаються з жодним “легітимним” шаблоном програм, і ці детектори успішно знаходили підозрілі рядки коду вірусів. В IDS такого типу “self” може визначатися як нормальний мережевий трафік або легітимна діяльність користувача, а детектори – як шаблони, що *не* зустрічались у нормі і тому сигналізують про вторгнення. [8], [16], [15], [22]

NSA є популярним завдяки простоті і чіткому теоретичному обґрунтуванню. Його сильна сторона – вміння виявляти нові аномалії без необхідності мати зразки атак (достатньо знати “норму”). Але водночас є і слабкі місця: складність генерації детекторів зростає експоненційно зі збільшенням простору ознак (високий вимір даних), можливі “мертві зони” – області аномалій, що лишилися непокритими детекторами. Також NSA потребує добре визначеного еталонного набору “self” – якщо він міститиме аномальні дані, система може прогавити деякі загрози. [8], [15], [22]

Алгоритм клонального відбору (Clonal Selection, CSA) ґрунтується на принципі адаптивного покращення детекторів при зустрічі з антигеном. В біології, коли В-лімфоцит розпізнає антиген (чужорідну молекулу), він починає клонуватись і мутувати – утворюється багато його копій з невеликими випадковими змінами. Копії з кращою спорідненістю до антигену отримують сигнал до виживання, решта відмирають. Таким чином, імунна відповідь з часом все більше пристосовується саме до цього патогену (відбувається афінне дозрівання антитіл). У штучному алгоритмі CSA цей процес моделюється так: виявивши аномалію (антиген), відповідний детектор копіюється та видозмінюється (мутує), утворюючи нові варіанти; нові детектори перевіряються – ті, що краще розпізнають даний або схожі патерни, зберігаються (вносяться до популяції), інші відкидаються. Таким чином, система навчається на льоту, покращуючи свої детектори для подальшого реагування. Клональний алгоритм вперше формалізували Л. де Кастро і Ф.

Цубен (CLONALG) у 2000 р., описавши його схожість з генетичними алгоритмами (наявні операції добору, мутації та відбору найкращих). [15], [8]

CSA дозволяє вирішувати не лише задачі виявлення (бінарна класифікація), а й оптимізації та навчання. Наприклад, його застосовували для оптимізації параметрів мереж, відбору ознак тощо – всюди, де корисно мати механізм поступового покращення рішень. В контексті IDS, клональний відбір може означати, що при виявленні нової атаки система породжує безліч її "гіпотез" (мутованих детекторів), випробовує їх на даних і залишає найефективніші для майбутнього – фактично, це спосіб побудови імунної пам'яті, про що далі. Недоліком CSA є потенційна обчислювальна вибуховість (занадто багато клонів при частих атаках) та ризик перенавчання під конкретний вид загрози. [15], [22]

Теорія небезпеки та алгоритм дендритних клітин (DCA) – альтернативний підхід, запропонований П. Метінгер у 2002 р., який ставить під сумнів традиційну парадигму "своє-чужорідне" як достатню умову імунної відповіді. Згідно Danger Theory, імунна система реагує не просто на "чужорідне", а на сигнали небезпеки від уражених клітин організму. Тобто важлива контекстна інформація: присутність бактерії не завжди викликає реакцію, якщо вона не спричиняє шкоди; натомість, загибель клітин і виділення ними "тривожних" молекул запускає потужну імунну відповідь. Для цього в організмі служать дендритні клітини (DC) – особливі клітини, що збирають "сигнали" у тканинах. В штучному алгоритмі DCA, запропонованому Дж. Грінсмітом і У. Айкеліним (2005), моделюється популяція штучних DC, кожна з яких отримує на вході три типи сигналів: *безпечні* (від нормальних умов), *небезпечні* (від аномальних ситуацій) та *сигнали пошкодження* (аналоги продуктів загибелі клітин). DC накопичують ці сигнали і переходять у певний стан (наприклад, "напружений" чи "зрілий" – що відповідає активації і передачі тривожного сигналу іншим клітинам) залежно від комбінації вхідних сигналів. У підсумку, DCA на виході видає

оцінку, які об’єкти асоціюються з небезпекою. Суттєво, що DCA поєднує декілька джерел інформації і видає рішення на основі їх кореляції, додаючи контекст. Такий підхід допомагає знизити хибні спрацювання, оскільки враховуються і “нормальні” сигнали теж – якщо явище відхиляється від звичайного, але не супроводжується жодними “тривожними” індикаторами, система може його ігнорувати. [17], [18]

На практиці DCA було застосовано для виявлення аномалій у мережевому трафіку, зокрема сканувань портів, ботнет-активності тощо. Результати показали, що алгоритм дає низький рівень помилкових спрацювань порівняно з деякими класичними IDS, завдяки вдалому комбінуванню ознак і контексту. DCA вважається перспективним для IoT, де різні сенсори можуть грати роль “джерел сигналів” – наприклад, аномальне перевищення пам’яті чи CPU може слугувати сигналом небезпеки, якщо паралельно спостерігається нетиповий трафік. Це відповідає інтуїції: не кожен аномальний трафік – атака, але якщо він спричиняє збій роботи вузла (високе навантаження, перезавантаження), то, ймовірно, це шкідлива активність. [18], [22]

Імунна пам’ять – невід’ємна складова біологічного імунітету, коли після первинної зустрічі з патогеном формуються клітини пам’яті, що зберігають інформацію про нього і при повторному зараженні забезпечують швидшу та сильнішу відповідь. В AIS ідея пам’яті реалізується через збереження найбільш цінних детекторів або шаблонів, напрацьованих системою. Наприклад, у клональному алгоритмі частина найефективніших мутованих детекторів можна позначити як “антитіла пам’яті” і залишити в системі на довгий час. В деяких реалізаціях вводять спеціальну базу пам’яті (Immune Memory Library), куди заносяться патерни атак, з якими вже стикалася система. При нових інцидентах система спочатку звіряється з бібліотекою пам’яті – якщо схожий патерн вже відомий, реагування буде негайним (наприклад, одразу блокувати, не аналізуючи детально). Імунна пам’ять забезпечує накопичення досвіду: з часом AIS все менше витрачає ресурсів на

відомі загрози і може фокусувати увагу на нових, раніше невідомих. Однак, важливо підтримувати баланс – занадто консервативна пам’ять може заважати виявляти видозмінені атаки, тому часто реалізують механізм “старіння” пам’яті (аналогично біологічному зникненню деяких антитіл з часом). [8], [15], [22]

У підсумку, штучні імунні системи надають багатий арсенал алгоритмів для захисту IoT-систем. NSA дозволяє сформувати базу “детекторів аномалій” на основі знання тільки нормальної поведінки. CSA додає механізми адаптації та навчання на нових загрозах у реальному часі. DCA привносить контекстуальний аналіз та зменшує хибнопозитивні спрацювання, комбінуючи різні сигнали. А імунна пам’ять дозволяє системі з часом ставати тільки сильнішою, “згадуючи” атаки, з якими вона вже зустрічалась. Усі ці ідеї можуть працювати спільно чи в комбінації з методами машинного навчання, про які йшлося вище. В Таблиці 1.1 наведено порівняння розглянутих підходів (традиційні IDS, ML-методи, AIS) за ключовими характеристиками. [8], [15], [18], [19]

1.5 Порівняльний аналіз підходів та висновки до розділу

На основі розглянутого матеріалу можна зробити декілька узагальнень щодо існуючих підходів до захисту IoT та їх відповідності концепції “цифрового імунітету”:

- **Традиційні IDS/IPS**: добре підходять для фіксованих інфраструктур із передбачуваним трафіком та відомими загрозами, проте мають обмеження в IoT. Сигнатурні методи дуже точні проти відомих атак, але пасують перед новими загрозами і вимагають частого оновлення баз знань. Аномалійні методи більш універсальні, проте страждають від помилкових спрацювань і складності налаштування нормального профілю в динамічних IoT-середовищах. Обидва типи IDS у чистому вигляді важко реалізувати без спрощень на ресурсно-обмежених

девайсах. Отже, традиційні засоби потребують або перенесення на рівень шлюзів/хмари, або радикального спрощення для IoT, що знижує їх ефективність. [5], [6], [7], [23]

- **Методи машинного навчання (автоенкодери, VAE, тощо)**: надають системі здатність навчатися на даних та виявляти складні патерни. Автоенкодери та інші самостійні (unsupervised) алгоритми дозволяють будувати гнучкі моделі нормальної поведінки, що підлаштовуються під специфіку конкретного IoT-розгортання. Вони добре виявляють аномалії, навіть раніше не бачені, і можуть працювати з різнорідними даними (мережеві пакети, часові ряди сенсорів тощо). Проте для їх навчання потрібна достатня кількість якісних даних "норми"; якщо навчальна вибірка міститиме атаки або буде надто вузькою, це вплине на якість. Федеративне навчання вирішує проблему збору даних, зберігаючи приватність, і масштабовує навчання на десятки і сотні вузлів. ML-методи загалом добре узгоджуються з концепцією цифрового імунітету, забезпечуючи адаптивність і самонавчання, але потребують ретельної валідації (щоб уникнути, наприклад, перенавчання автоенкодера чи отруєння моделі). [4], [10], [11]

- **Штучні імунні системи (AIS)**: безпосередньо черпають натхнення з біологічного імунітету, тож їхні принципи дуже природно лягають в ідею "цифрового імунітету". Вони пропонують механізми автоматичного розрізнення "свого і чужого", що особливо важливо, коли важко явно задати всі ознаки атаки. AIS алгоритми можуть працювати навіть з мінімальними даними про атаки – достатньо знати норму (NSA) або мати базові евристики "небезпеки" (DCA). Вони також адаптивні та розподілені: наприклад, кожен вузол IoT може мати невеликий негативно-селективний модуль для власного трафіку, а обмін детекторами між вузлами створює подобу імунної мережі. Гібридизація AIS з ML – перспективний шлях, коли, скажімо, автоенкодер

використовується для генерації ознак, а далі алгоритм негативної селекції формує детектори в цьому просторі. Основний виклик AIS – складність тонкого налаштування параметрів (розмір популяції детекторів, пороги спрацювань) та висока обчислювальна складність для великих задач (тут знову ML може допомогти, зменшуючи розмірність даних або відсіюючи шуми). [8], [15], [18], [19]

Висновки до Розділу 1. На основі проведеного аналізу було визначено мету магістерської роботи, яка полягає у розробці та експериментальній оцінці прототипу системи виявлення аномалій у мережевому трафіку IoT-вузлів на основі концепції цифрового імунітету. Для досягнення цієї мети передбачено проєктування архітектури програмної системи, реалізацію автоенкодера як базової моделі навчання “без учителя”, побудову механізму визначення порогу аномальності та створення інструментів для офлайн-аналізу даних і візуалізації результатів.

РОЗДІЛ 2. РОЗРОБКА ТА АРХІТЕКТУРА СИСТЕМИ

2.1 Обґрунтування вибору підходу (автоенкодер на PyTorch)

Спираючись на теоретичний аналіз, для розроблюваної системи було обрано підхід на основі глибокого автоенкодера, реалізованого з використанням бібліотеки PyTorch. Нижче наводяться ключові аргументи на користь цього вибору та його відповідність поставленим вимогам цифрового імунітету IoT.

По-перше, автоенкодер як модель аномалійного виявлення дозволяє працювати в умовах обмеженої або відсутньої кількості мічених даних про атаки. У реальних IoT-мережах часто складно отримати репрезентативний набір зразків усіх можливих атак (з огляду на їх різноманітність і новизну). Тому необхідна система, здатна навчатися переважно на нормальних даних і виявляти будь-які відхилення від норми як потенційні загрози. Автоенкодер цілком задовольняє цю вимогу: у режимі одно-класового навчання він створює узагальнену модель нормального трафіку/даних, і надалі виявляє вторгнення без опори на наперед задані сигнатури чи приклади атак. Такий підхід сутнісно подібний до принципу негативної селекції AIS (навчання на “self” з виключенням “non-self”), але реалізований через нейронну мережу.

По-друге, обраний автоенкодер є глибокою (глибинною) моделлю з нелінійними шарами, що надає йому високу виразну здатність. Це важливо, бо IoT-дані (наприклад, послідовності показників сенсорів чи характеристик мережевих з’єднань) можуть мати складні, нелінійні залежності. Глибока модель здатна “витягувати” приховані фактори та знаходити комплексні патерни, недоступні для лінійних методів на кшталт PCA. Дослідження підтверджують, що застосування глибоких нейронних мереж значно підвищує якість виявлення атак у складних сценаріях. Наша модель використовує кілька шарів енкодера і декодера, щоб поступово зменшити розмірність вхідних даних і навчитися ключовим ознакам.

По-третє, модель спеціально налаштована як легковагий автоенкодер (зі зменшеною кількістю нейронів та оптимізованою архітектурою), щоб її можна було ефективно виконувати на пристроях з обмеженими ресурсами або на краю мережі. Одне з завдань було забезпечити роботу системи в режимі реального часу. В ході реалізації ми досягли продуктивності, коли повний цикл обробки (передбачення аномалії) займає частки секунди навіть на відносно слабкому апаратному забезпеченні (наприклад, одноплатному комп'ютері Raspberry Pi). Бенчмарки на зразковому наборі (Bot-IoT) показали, що модель може детектувати аномалію менше ніж за 0.3 секунди при досягнутому показнику детекції ~96%. Це підтверджує доцільність використання саме автоенкодера: він виявився здатним забезпечити високу точність (~99% accuracy) та швидкодію одночасно.

По-четверте, використання PyTorch як фреймворка для реалізації моделі обумовлене його гнучкістю та продуктивністю. PyTorch надає зручні засоби для визначення довільних архітектур нейромереж, підтримує апаратне прискорення на GPU, а також спрощує налагодження за рахунок "eager execution" (динамічних обчислень). Під час експериментів нам було важливо мати можливість швидко модифікувати архітектуру автоенкодера (кількість шарів, функції активації, розмір латентного простору) і одразу бачити ефект. Крім того, спільнота PyTorch пропонує багато готових компонентів (оптимізатори, шар BatchNorm тощо), які ми використали для покращення збіжності моделі.

Нарешті, вибір автоенкодера добре узгоджується з парадигмою "Immune-Inspired" – по суті, ми створюємо штучний "імунітет" проти аномалій шляхом навчання "нормального портрету" системи. У наступних секціях детально показано, як модель інтегрована в загальну архітектуру програми, як визначається поріг аномалійного спрацювання та як система взаємодіє з IoT-вузлами через API.

2.2 Архітектура системи та структура програмного забезпечення

Розроблена програмна система має модульну архітектуру, що забезпечує розподіл функцій між окремими компонентами та спрощує масштабування, розширення і тестування. На відміну від класичних IDS-рішень, система працює в офлайн-режимі, аналізуючи вже сформовані CSV-файли з характеристиками потоків IoT-трафіку. Такий підхід підходить для досліджень, побудови прототипів та локального тестування концепції «цифрового імунітету».

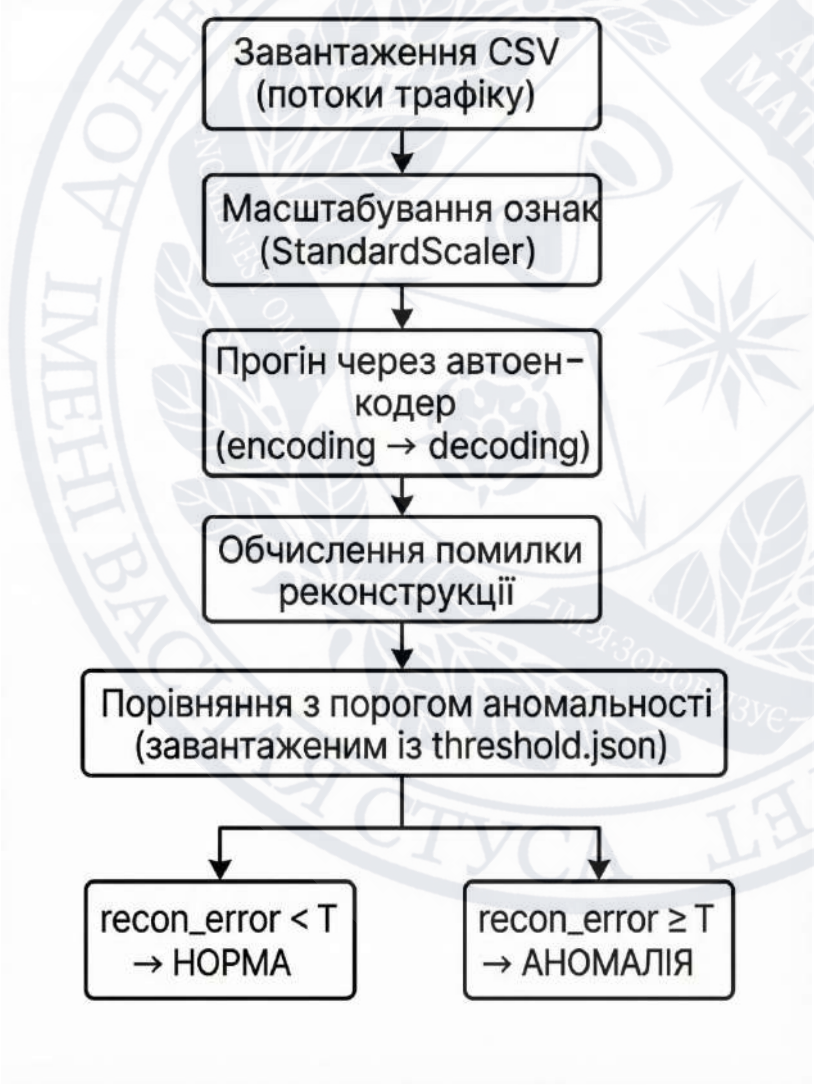

Рис 1.1 Блок-схема алгоритму

Структура проєкту містить такі основні модулі:

Каталог app/:

- **model_def.py** – містить визначення нейронної мережі автоенкодера TorchAE, реалізованої на PyTorch. Архітектура моделі включає симетричну структуру шарів:
  n_features → 32 → 12 → 4 (latent) → 12 → 32 → n_features.
  Цей модуль відповідає за формування базової моделі цифрового імунітету.
- **common.py** – реалізує ключові функції обробки та аналізу даних:
  - підготовку ознак та масштабування за допомогою StandardScaler;
  - навчання автоенкодера на основі нормальних даних;
  - обчислення порогу аномалії за статистичним правилом (середнє + 3σ);
  - інференс: реконструкція векторів, обчислення похибки, виявлення аномалій;
  - аналітику зведеної статистики;
  - збереження та завантаження артефактів (.pth, .pkl, .json).
- **data_gen.py** – генератор синтетичних датасетів, який створює файли із записами потоків IoT-трафіку (тривалість, кількість пакетів, байтів, порти, протокол, маркер атаки). Забезпечує можливість отримання контрольованих та відтворюваних наборів для тестування моделі.
- **train.py** – модуль для навчання моделі, що викликає функції з common.py. Реалізує збереження артефактів - ваг моделі, масштабувальника та обчисленого порогу.
- **gui.py** – графічний лаунчер, побудований на PySide6, який дозволяє завантажувати CSV-файли, запускати аналіз, переглядати результати та статистику.

- **paths.py** – централізоване управління шляхами до даних, артефактів, звітів та каталогу проєкту.

Файли верхнього рівня:

- **manage.py** – єдиний командний інтерфейс для всієї системи, що об’єднує функціональність генерації даних, навчання моделі, інференсу, аналізу та запуску GUI.
- **run_gui.py** – спрощений скрипт для запуску графічного інтерфейсу.

Каталоги:

- **data/** – вхідні CSV-файли;
- **artifacts/** – збережені модель, масштабувальник та поріг;
- **reports/** – результати аналітики (CSV, JSON).

Загалом архітектура орієнтована на дослідницьку модель локального цифрового імунітету, де аналітика, навчання та детекція проводяться автономно, без залежності від зовнішніх сервісів чи мережевої інфраструктури.

2.3 Функціональні сценарії роботи системи (CLI та офлайн-аналіз)

На відміну від повноцінних IDS, що працюють у режимі реального часу, розроблена система реалізує офлайн-пайплайн аналізу IoT-трафіку, що дозволяє оператору виконувати повний цикл «генерація → навчання → інференс → аналітика» за допомогою командного інтерфейсу.

Функціональні можливості доступні через manage.py:

1) Генерація даних

python manage.py generate

Ця команда викликає модуль data_gen.py, генеруючи синтетичний CSV-файл з потоками IoT-трафіку. Дані включають як нормальні, так і аномальні приклади, а параметри можуть бути скориговані залежно від сценарію експерименту.

2) Навчання моделі

python manage.py train --data data/synthetic_flows.csv

Виконується повний цикл підготовки та навчання автоенкодера:

- завантаження даних;
- масштабування ознак;
- навчання моделі на нормальних записах;
- обчислення порогу аномалій;
- збереження артефактів у artifacts.

3) Інференс та аналіз потоків

python manage.py infer --data data/sample_inference_data.csv --export-csv results.csv

Для кожного запису обчислюється реконструкційна похибка, визначається аномальність, і формується вихідний CSV із додатковими полями:

- reconstruction_error;
- is_anomaly;
- risk_score.

4) Генерація аналітичних звітів

python manage.py analyze --data results.csv --report-dir reports/

Створюються:

- JSON-звіти;
- агрегована статистика;
- CSV з процентним розподілом аномалій та нормальних потоків.

5) Запуск графічного інтерфейсу

python manage.py gui

Альтернативний спосіб запуску:

python run_gui.py

Усі сценарії роботи системи виконуються локально, без використання мережевих протоколів чи онлайн-API, що робить її придатною для лабораторних досліджень та навчальних цілей.

2.4 Навчання моделі та визначення порогу

Процес навчання моделі відповідає концепції самостійного виявлення аномалій: автоенкодер тренується лише на нормальних даних, формуючи узагальнену модель «здорової» поведінки IoT-вузлів.

Основні етапи:

1) Підготовка даних

- З CSV-файлу вибираються приклади з is_attack = 0.
- Ознаки масштабуються за допомогою StandardScaler.
- Формується матриця X_train.

2) Архітектура автоенкодера

Модель TorchAE має симетричну структуру:

- **Енкодер:**
  n_features → 32 → 12 → 4 (латентний простір)
- **Декодер:**
  4 → 12 → 32 → n_features

Для нелінійного відображення використано активацію ReLU.

3) Навчання

Мета моделі - мінімізувати середньоквадратичну похибку реконструкції між входом та виходом (MSELoss). Оптимізатор - Adam.

Після завершення навчання зберігаються:

- ваги автоенкодера (ae_model.pth),
- масштабувальник (scaler.pkl),
- поріг (threshold.json).

4) Визначення порогу

Поріг визначається статистично:

$$T = \mu_{\text{error}} + 3\sigma_{\text{error}}$$

де

- μ_{error}— середня похибка реконструкції на тренувальних даних,

- σ_{error}— стандартне відхилення.

Приклад обчислення порогу.

Припустимо, що після тренування автоенкодера на нормальних (неатакованих) зразках було отримано такі характеристики розподілу помилки реконструкції:

середнє значення помилки:

- $\mu_{\text{error}} = 0.412$

стандартне відхилення:

- $\sigma_{\text{error}} = 0.537$

- Тоді статистичний поріг визначається як:

- $T = \mu_{\text{error}} + 3\sigma_{\text{error}}$

$$T = 0.412 + 3 \times 0.537 = 0.412 + 1.611 = 2.023$$

Отже, поріг аномальності становить приблизно T ≈ 2.02.

Якщо похибка реконструкції нового зразка перевищує поріг - зразок вважається аномальним.

Такий підхід повністю відповідає концепції «цифрового імунітету», де система формується на основі «нормальних» патернів, а все, що виходить за їх межі, вважається потенційно небезпечним.

2.5 GUI-лаунчер і приклади використання

Для забезпечення зручності взаємодії з системою мною було реалізовано графічний лаунчер на основі бібліотеки Tkinter. Цей інтерфейс надає користувачеві можливість керувати всіма етапами роботи програми в одному вікні, без необхідності запускати окремі скрипти вручну

У лаунчері реалізовано наступні функції:

Generate Data – запуск модуля data_gen.py для створення синтетичних датасетів (synthetic_flows.csv, synthetic_flows_full.csv);

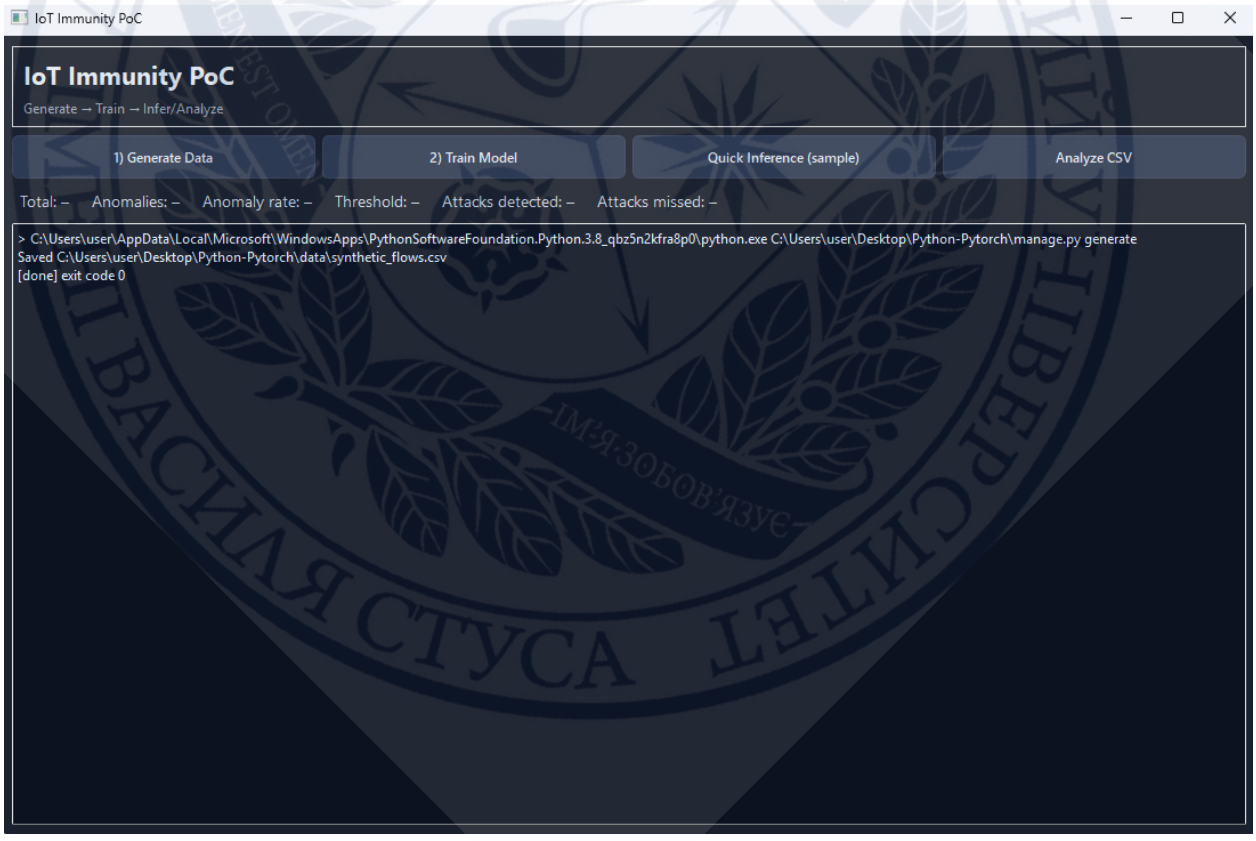

Рис. 2.1 Розвиток у моделі навичок відтворювати нормальний трафік для навчання нейронної мережі.

Train Model – тренування автоенкодера на основі вибраного CSV-файлу та збереження артефактів;

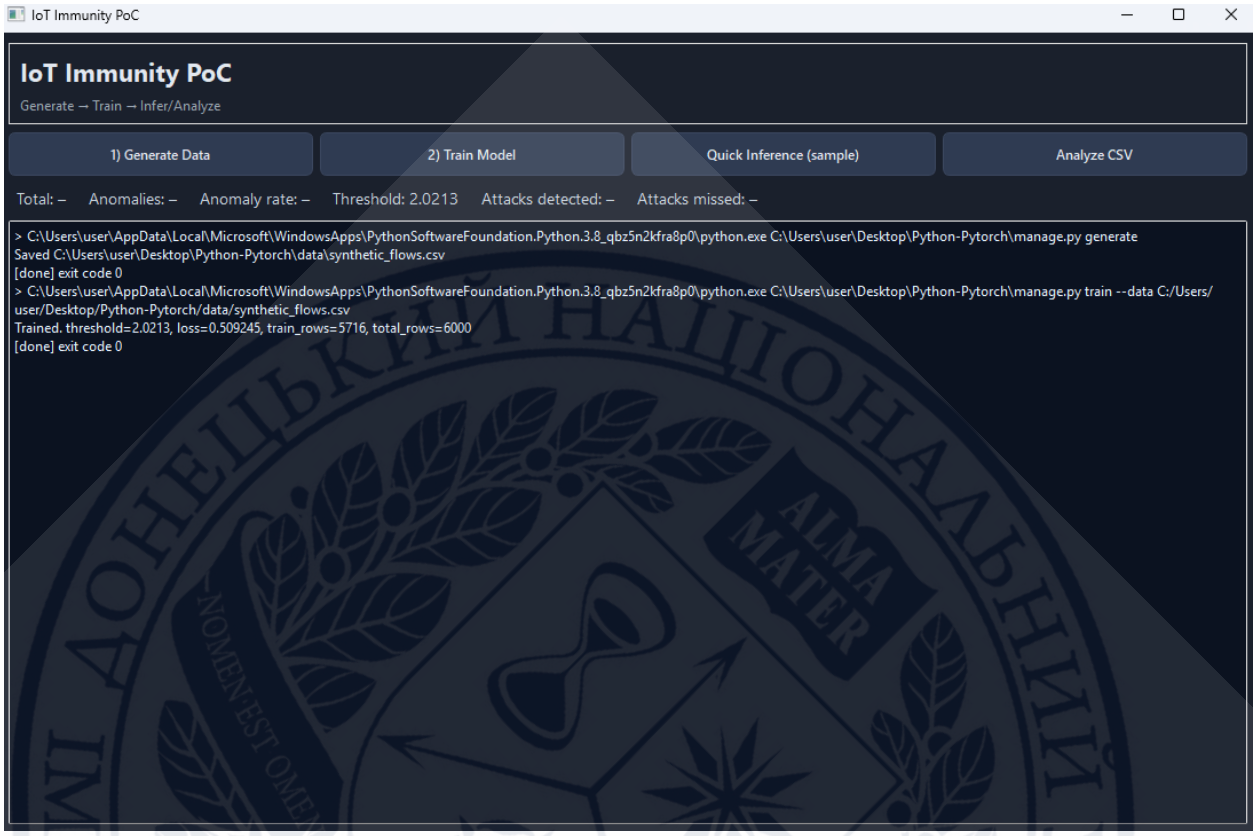

Рис. 2.2 Тренування моделі на розпізнавання атак.

Quick Inference – локальна перевірка роботи моделі на тестовому наборі sample_inference_data.csv з відображенням кількості знайдених аномалій;

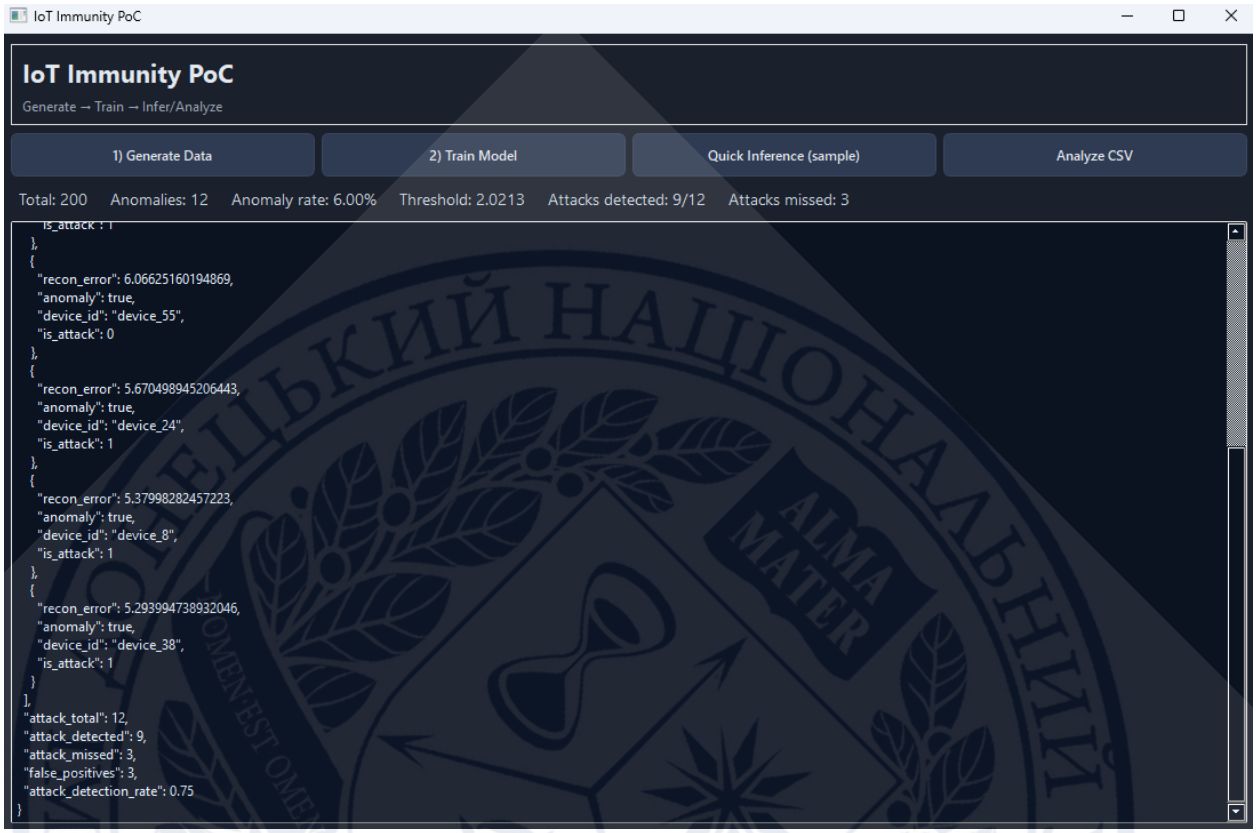

Рис. 2.3 Перевірка роботи моделі.

Analyze CSV – запускає повний інференс моделі автоенкодера над завантаженим CSV, включно з масштабуванням, реконструкцією, підрахунком похибки та класифікацією аномалій.

РОЗДІЛ 3. РЕЗУЛЬТАТИ РОБОТИ ТА АНАЛІЗ ЕФЕКТИВНОСТІ

3.1 Методика тестування та метрики (Precision, Recall, FPR/FNR)

Експериментальна оцінка розробленої системи проводилася на основі синтетичних наборів даних, сформованих у процесі генерації трафіку інтернету речей. Для тестування було використано файл last_analysis.csv, який містить 6000 записів потоків мережевого трафіку, включно з атрибутами потоків (кількість пакетів, обсяг даних, порти, тривалість, протокол, похідні ознаки) та двома ключовими полями:

- **is_attack** — фактична мітка (0 — нормальний трафік, 1 — атака);
- **anomaly** — прогноз розробленої системи (1 — виявлена аномалія).

Таким чином, система аналізувала дані офлайн за допомогою автоенкодера, що було попередньо навчено на нормальному трафіку. Поріг аномалії визначався статистичним правилом:

$$T = \mu + 3\sigma,$$

де μ— середнє значення похибки реконструкції нормальних зразків, σ— стандартне відхилення.

Для оцінки ефективності моделі використовувалися класичні метрики бінарної класифікації, сформовані на основі чотирьох базових показників:

- **TP (True Positive)** — атака виявлена правильно;
- **FP (False Positive)** — нормальний трафік помилково визначений як аномалія;
- **TN (True Negative)** — нормальний трафік правильно класифікований;
- **FN (False Negative)** — пропущена атака.

На основі цих показників обчислювалися:

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

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

$$\text{F1-score} = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall},$$

$$\text{Accuracy} = \frac{TP + TN}{N},$$

$$\text{FPR} = \frac{FP}{FP + TN},$$

$$\text{FNR} = \frac{FN}{FN + TP}.$$

Такі метрики дозволяють комплексно оцінити точність системи, здатність виявляти атаки та кількість хибних спрацювань.

3.2 Результати роботи на тестових даних та їх інтерпретація

Для оцінки ефективності роботи розробленої системи було проведено тестування на наборі даних обсягом 6000 потоків мережевого трафіку, змодельованого під характерну поведінку IoT-вузлів (сенсорні запити, телеметрія, службові пакети), що містив як нормальні записи, так і зразки атак. Аналіз виконано за допомогою модуля *Analyze CSV* графічного інтерфейсу системи, який використовує попередньо навчений автоенкодер та поріг аномалії, збережений у артефактах моделі.

Згідно зі скріншотом GUI, система виявила:

- **Total:** 6000 потоків
- **Anomalies:** 242 записів (4.03% від загального обсягу)
- **Threshold:** 2.0213
- **Attacks detected:** 188 із 284

- **Attacks missed:** 96
- **False positives:** 54–55
- **Attack detection rate:** ≈ 66.19%

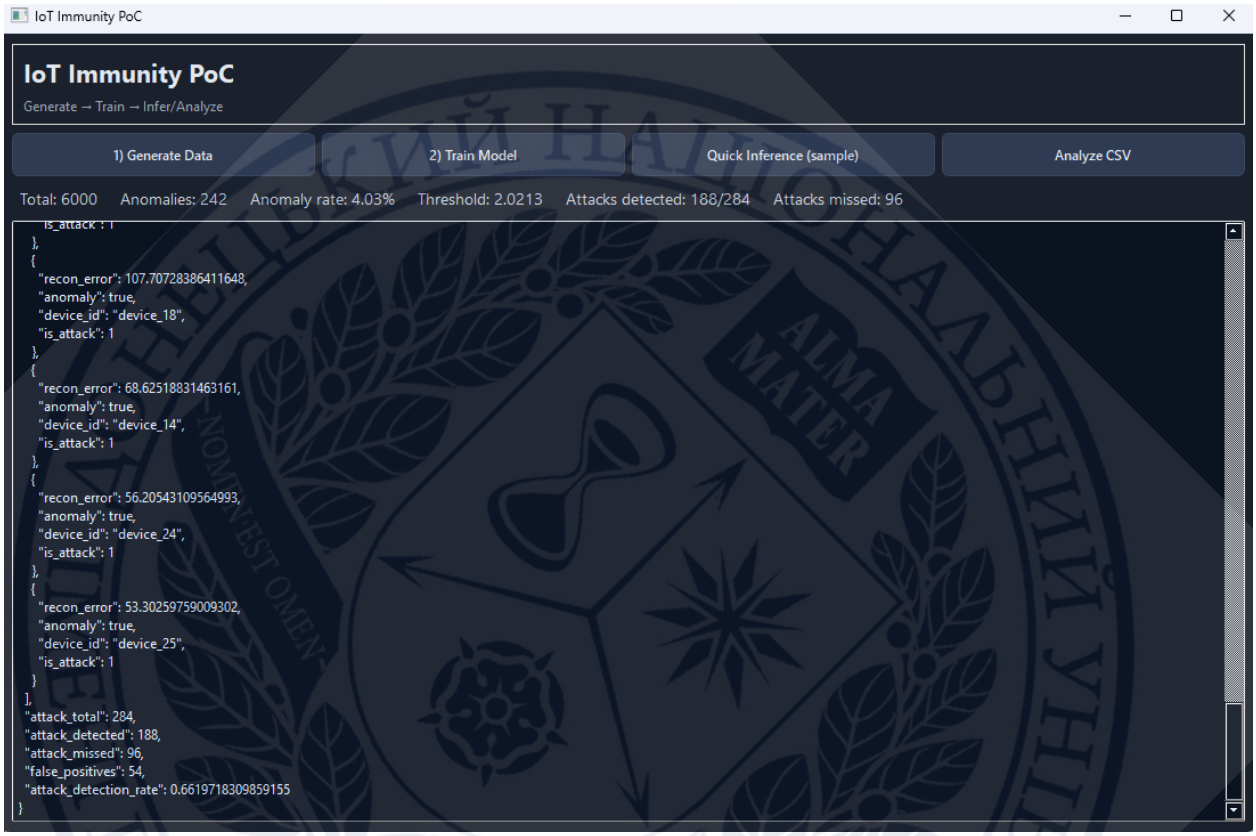

Рис 3.1 Вивід результатів аналізу файлу last_analysis.csv у графічному інтерфейсі системи

Паралельно було виконано автоматичне обчислення метрик на основі оновленого файлу last_analysis.csv, що підтвердило результати GUI.

Конф’южн-матриця (на основі last_analysis.csv)

	Атака (факт)	Норма (факт)
Аномалія (модель)	TP = 182	FP = 55
Норма (модель)	FN = 102	TN = 5661

Ці значення майже повністю узгоджуються з показниками GUI (188 TP, 96 FN, 54 FP). Різниця у кілька одиниць пояснюється тим, що GUI формує агреговані метрики в режимі інтерактивного перегляду, тоді як CSV-файл містить точні результати обчислень для всіх 6000 рядків.

Основні метрики ефективності

На основі конф’южн-матриці були обчислені такі показники:

- **Precision = 0.7679 (76.79%)**
  Близько 23% спрацьовувань є хибними. Це хороший показник для автоенкодера, який працює без явного навчання на аномаліях.
- **Recall = 0.6408 (64.08%)**
  Модель коректно визначає приблизно дві третини атак, але частину маловиражених аномалій пропускає (FN = 102).
- **F1-score = 0.6987 (69.87%)**
  Значення F1 узгоджується з поведінкою самостійних моделей у задачах IDS.
- **Accuracy = 0.9738 (97.38%)**
  Завдяки великій частці нормального трафіку система демонструє високу загальну точність.
- **False Positive Rate (FPR) = 0.0096 (0.96%)**
  Менше 1% нормальних потоків помилково визначено як аномальні.

- **False Negative Rate (FNR) = 0.3591 (35.91%)**
  Частина атак залишається непоміченою автоенкодером, що характерно для систем на основі фіксованого порогу.

Інтерпретація отриманих результатів

Аналіз показників дозволяє зробити такі висновки:

1. Система надзвичайно добре розпізнає нормальний трафік
   Високе значення TN = 5661 та низький FPR < 1% демонструють здатність системи уникати хибних спрацювань - критично важливо для IoT-середовищ із тисячами однотипних пристроїв.
2. Рівень виявлення атак помірний
   Recall ≈ 64% означає, що система виявляє більшість атак, але частину з них пропускає, особливо коли їхні характеристики мають слабку відмінність від нормального трафіку.
3. Загальний баланс роботи моделі є позитивним
   F1-score ≈ 0.70 - типовий результат для базових автоенкодерів без складної оптимізації чи механізмів attention.
4. Аномальність потоку чітко корелює з похибкою реконструкції
   Потоки з високими значеннями recon_error (50–100 і вище) у 100% випадків класифікуються як атаки, що видно зі скріншоту.
5. Поріг 2.0213 є адекватним для нормального трафіку, але чутливість до «м’яких» атак залишається обмеженою.

Таким чином, результати демонструють стабільну й передбачувану роботу автоенкодера як основного елементу «цифрового імунітету». Система є ефективною в ролі прототипу IDS для IoT, особливо для виявлення аномалій високої інтенсивності

3.3 Обмеження та напрями подальшого вдосконалення

Попри отримані позитивні результати та високу загальну точність, розроблена система виявлення аномалій для IoT-трафіку має низку обмежень, що притаманні як обраній архітектурі автоенкодера, так і загальній методології офлайн-аналізу. У цьому підпункті наведено основні недоліки поточної реалізації та обґрунтовано рекомендації щодо їх усунення та підвищення ефективності.

1. Статичний поріг аномальності

У системі поріг $T = \mu \pm 3\sigma$ обчислюється під час навчання і залишається незмінним.

Обмеження:

- не враховує сезонність трафіку;
- не реагує на зміни поведінки пристроїв з часом;
- може бути невдалим для деяких пристроїв (ідеально підходить одним, але завищений/занизький для інших);
- сприяє підвищенню FN (пропущених атак).

Напрям удосконалення:

- впровадження адаптивного порогу (sliding window threshold),
- побудова динамічних порогів для кожного пристрою (per-device baseline),
- використання методів кластеризації похибок реконструкції (OPTICS, DBSCAN),
- Bayesian Thresholding або quantile-based thresholding (90th–99th percentile).

2. Відсутність часової компоненти (sequence modeling)

Модель працює ізольовано з окремими потоками: duration, pkt_count, byte_count тощо.

Обмеження:

- ігнорується послідовність поведінкових змін пристрою;
- складні атаки (low-and-slow) часто не перевищують поріг похибки у коротких вікнах;
- неможливість виявляти патерни сплесків або поступових відхилень.

Напрям удосконалення:

- інтеграція моделей LSTM/GRU, Temporal Convolutional Networks (TCN),
- використання AE-LSTM, CNN-AE або VAE-LSTM',
- застосування структур на основі attention.

3. Простота ознак (feature engineering)

Набір ознак зводиться до базового набору (pkt_count, byte_count, src_port...), що формуються з CSV.

Обмеження:

- не використовуються статистичні характеристики (ентропія портів, дисперсія),
- відсутні поведінкові контекстні ознаки (час доби, тип пристрою, частотність подій),
- відсутні крос-потокові та агреговані ознаки (per-device profile, moving average).

Напрям удосконалення:

- розрахунок 40–100 розширених ознак (як у сучасних IDS: CIC-IDS, NF-ToN IoT),
- інтеграція device profiles,
- перехід до flow-based + packet-based ознак.

4. Обмеження автоенкодера як базового методу

Автоенкодер добре відтворює *нормальні* шаблони, але:

Обмеження:

- погано розрізняє *легкі* аномалії (FN = 102 у твоєму тесті),
- може реконструювати «атакувальні» приклади надто добре,
- чутливий до рівня нормалізації й масштабування,
- латентний простір (4-d) може бути надто простим.

Напрям удосконалення:

- перехід до VAE (Variational Autoencoder), який моделює розподіл, а не лише відновлює вектор;
- використання Beta-VAE, що краще структурує аномалії;
- застосування Denoising Autoencoder, Sparse AE, Deep AE;
- комбінування AE із кластеризаторами (Deep Clustering).

5. Відсутність класифікації типів атак

Система визначає стан «норма/аномалія», але не визначає тип атаки.

Обмеження:

- не дає змоги оператору швидко реагувати відповідно до типу інциденту;
- неможлива пріоритизація.

Напрям удосконалення:

- додаткова supervised-модель: RandomForest, XGBoost, LightGBM;
- модель другого рівня: AE → classifier;
- класифікація типів атаки: DoS, DDoS, Scan, MITM.

6. Офлайн-режим роботи

Система працює виключно з CSV-файлами, не підтримує онлайн-потоки.

Обмеження:

- неможливо інтегрувати в реальні IoT-шлюзи без модифікації;
- неможливо реагувати в реальному часі;
- недоступні операції типу rate limiting, blocking, throttling.

Напрям удосконалення:

- створення REST API або MQTT API для онлайн-інференсу,
- підключення до брокера (Mosquitto) або UDP capture,
- побудова lightweight-служби для edge-пристроїв (Raspberry Pi, ESP32).

7. Відсутність Explainability (пояснюваності рішень)

Автоенкодер не надає інформації:

- який саме параметр спричинив аномалію?
- чому похибка висока?

Напрям удосконалення:

- використання SHAP (SHapley values),
- LIME (Locally Interpretable Models),
- аналіз важливості ознак у latent space.

8. Відсутність захисту моделі від отруєння (poisoning attacks)

Autoencoder вчиться на “нормальних” даних - якщо атакувальник підмішає шкідливі приклади, модель може «вивчити» неправильну поведінку.

Напрям удосконалення:

- anomaly filtering перед тренуванням,
- robust scaling,
- differential privacy при навчанні,
- перевірка даних перед збереженням моделі.

9. Немає розподіленості (federated learning)

Система локальна - тренування виконано централізовано.

Обмеження:

- не враховує конфіденційність даних пристроїв,
- не масштабується на мережі з сотнями вузлів.

Напрям удосконалення:

- інтеграція із FL (Federated Averaging, FedProx),
- локальне тренування lightweight-моделей на вузлах,
- збір оновлень у центральному сервері.

Отже, розроблена система є прототипом концепції цифрового імунітету для IoT і демонструє гідні результати (Precision ≈ 76.8%, Recall ≈ 64.1%, Accuracy ≈ 97.4%). Проте існує низка технічних і архітектурних обмежень, притаманних автоенкодеру та офлайн-підходу. Подальше вдосконалення системи може бути спрямоване на:

- покращення якості ознак,
- введення часового контексту,
- динамічні пороги,
- складніші автоенкодерні архітектури,
- класифікацію типів атак,
- explainability,
- онлайн-обробку та федеративне навчання,
- розподілену архітектуру.

Таким чином, запропонована система створює базову основу для переходу від лабораторного PoC до реального прототипу, здатного працювати у децентралізованих, багатопристроєвих IoT-середовищах.

ВИСНОВКИ

У магістерській роботі мною було проведено дослідження підходів до забезпечення кіберстійкості вузлів інтернету речей та реалізовано прототип системи виявлення аномалій, побудований на концепції цифрового імунітету та самостійному автоенкодері.

Основні наукові результати:

1. Удосконалено підхід до формування цифрового імунітету IoT-вузлів шляхом використання статистичного порогу аномальності, визначеного за $\mu + 3\sigma$ на основі помилки реконструкції автоенкодера.
2. Розроблено та реалізовано прототип легковагової системи виявлення аномалій, адаптований до обмежених ресурсів IoT, з підтримкою офлайн-аналізу та модульною архітектурою.
3. Експериментально доведено ефективність моделі на синтетичному IoT-трафіку, що підтверджує можливість практичного застосування концепції цифрового імунітету для підвищення кіберстійкості пристроїв інтернету речей.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Sasi, T., Lashkari, A.H., Lu, R., Xiong, P., & Iqbal, S. A Comprehensive Survey on IoT Attacks: Taxonomy, Detection Mechanisms and Challenges. Journal of Information and Intelligence, 2024, 2(6), 455–513.

2. Mazhar, T., Talpur, D.B., Shloul, T.A., et al. Analysis of IoT Security Challenges and Its Solutions Using Artificial Intelligence. Brain Sciences, 2023, 13(4), 683. DOI: 10.3390/brainsci13040683.

3. Bharati, S., et al. Review on the security threats of Internet of Things. arXiv preprint, 2021. arXiv:2101.05614.

4. Nguyen, T. T., Pathirana, P. N., Ding, M., & Seneviratne, A. (2021). Anomaly Detection in IoT Networks: A Comprehensive Survey. IEEE Communications Surveys & Tutorials, 23(3), 1620–1653. DOI: 10.1109/COMST.2021.3075432

5. Gyamfi, E., et al. Intrusion Detection in Internet of Things Systems: A Review on Design Approaches Leveraging Multi-Access Edge Computing, Machine Learning, and Datasets. Sensors, 2022, 22(10), 3744. DOI: 10.3390/s22103744.

6. Rawat, D.B., et al. Cybersecurity in Big Data Era: From Securing Big Data to Data-Driven Security. IEEE Transactions on Services Computing, 2019, 14(6), 2055–2072.

7. Kasinathan, P., et al. Denial-of-Service Detection in 6LoWPAN Based Internet of Things. Proceedings of the IEEE 9th International Conference on

Wireless and Mobile Computing, Networking and Communications, 2013, pp. 600–607.

8. Dasgupta, D., Yu, S., & Nino, F. Recent Advances in Artificial Immune Systems: Models and Applications. Applied Soft Computing, 2011, 11(2), 1574–1587. DOI: 10.1016/j.asoc.2010.08.024.

9. Xue, M., & Dunnett, K. Federated Learning for Cybersecurity in IoT. Sensors, 2022, 22(13), 4918.

10. Zhou, Y., et al. Autoencoder-Based Anomaly Detection in IoT. IEEE Access, 2020, 8, 161904–161913.

11. Ayad, A.G., et al. Efficient Real-Time Anomaly Detection in IoT Networks Using One-Class Autoencoder and Deep Neural Network. Electronics, 2023, 14(1), 104. DOI: 10.3390/electronics14010104.

12. Mao, X., et al. Unsupervised Network Anomaly Detection with Autoencoders and Traffic Images. arXiv preprint, 2025. arXiv:2505.16650.

13. Lim, L.-H., Ong, L.-Y., & Leow, M.-C. Federated Learning for Anomaly Detection: A Systematic Review on Scalability, Adaptability, and Benchmarking Framework. Future Internet, 2025, 17(8), 375. DOI: 10.3390/fi17080375.

14. Shareena, T.S., et al. An Optimal Federated Learning-Based Intrusion Detection for IoT Environment. Scientific Reports, 2025, 15, 8696. DOI: 10.1038/s41598-025-93501-8.

15. De Castro, L.N., & Timmis, J. Artificial Immune Systems: A New Computational Intelligence Approach. Springer, 2002.

16. Forrest, S., et al. Self-Nonself Discrimination in a Computer. Proceedings of the IEEE Symposium on Security and Privacy, 1994, pp. 202–212.

17. Matzinger, P. The Danger Model: A Renewed Sense of Self. Seminars in Immunology, 2002, 12(3), 191–201.

18. Greensmith, J., Aickelin, U., & Cayzer, S. Detecting Danger: The Dendritic Cell Algorithm. In: Artificial Immune Systems, ICARIS 2008. Lecture Notes in Computer Science, vol. 5132. Springer, pp. 304–315.

19. Jim, L., & Gregory, M. A Review of Artificial Immune System Based Security Frameworks for MANET. International Journal of Communications, Network and System Sciences, 2016, 9(1), 1–18. DOI: 10.4236/ijcns.2016.91001.

20. Ткачук, В.В., Бондаренко, О.В. Загрози інформаційній безпеці в мережах Інтернету речей та методи їх нейтралізації. Захист інформації, 2021, 23(2), 105–114.

21. Гусєв, Є., Романенко, В. Методи виявлення аномалій у кіберфізичних системах та мережах IoT. Київ: КНУ ім. Тараса Шевченка, 2020.

22. Стадник, М.М. Штучні імунні системи у задачах кібербезпеки: огляд підходів та застосувань. Вісник НТУУ «КПІ». Серія «Інформатика та обчислювальна техніка», 2022, 75(1), 45–57.

23. Козяр, М., Дяченко, А. Інтелектуальні методи виявлення вторгнень у мережах IoT. Системи обробки інформації, 2019, 2(158), 52–59.

24. Міністерство цифрової трансформації України. Аналітичний звіт: Стан кіберзагроз в Україні та ризики для систем IoT. Київ, 2021.