МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

МАХНОВ ЯРОСЛАВ ГЕННАДІЙОВИЧ

Допускається до захисту:
в. о. завідувача кафедри
прикладної математики та
кібербезпеки, доктор філософії з
математики
_____Алла ЛУЦЕНКО
«__»_____20__р.

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ВИЯВЛЕННЯ МЕРЕЖЕВИХ АТАК НА IoT СИСТЕМУ "SMART CITY" З ВИКОРИСТАННЯМ МЕТОДІВ ШТУЧНОГО ІНТЕЛЕКТУ

Спеціальність 105 Прикладна фізика та наноматеріали
Кваліфікаційна (магістерська) робота

Керівник:
Любов ЗАГОРУЙКО,
к.т.н., доцент,
доцент кафедри прикладної
математики та кібербезпеки
_____
(підпис)

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

Вінниця — 2024

АНОТАЦІЯ

Махнов Я. Г. Розробка програмного забезпечення для виявлення мережевих атак на IoT систему "Smart City" з використанням методів штучного інтелекту. Спеціальність 105 «Прикладна фізика та наноматеріали», Освітня програма «Технології інтернету речей». Донецький національний університет імені Василя Стуса, Вінниця, 2024.

У роботі проаналізовано основні підходи та методи виявлення кібератак на IoT системи у контексті інформаційного середовища «Smart City». Обґрунтовано вибір моделі виявлення аномалій з використанням методів штучного інтелекту. Розроблено програмне забезпечення для виявлення мережевих атак, яке забезпечує комплексну оцінку здатності моделі ефективно виявляти атаки та відповідає вимогам щодо надійності й ефективності виявлення відхилень у поведінці IoT-пристроїв.

Ключові слова: Інтернет речей (IoT), штучний інтелект, виявлення аномалій, виявлення атак, виявлення вторгнень, мережеві атаки.

70 с., 1 табл., 30 рис., 1 дод., 50 джерел.

ABSTRACT

Makhnov Y. H. Software development for detecting network attacks on the "Smart City" IoT system using artificial intelligence methods. Specialty 105 "Applied Physics and Nanomaterials", Educational Program "Internet of Things Technologies". Vasyl Stus Donetsk National University, Vinnytsia, 2024.

The qualification (master's) work analyzes the main approaches and methods for detecting cyberattacks on IoT systems in the context of the "Smart City" information environment. The choice of an anomaly detection model using artificial intelligence methods is substantiated. The developed software for detecting network attacks provides a comprehensive assessment of the model's ability to correctly classify instances and meets the requirements for reliability and efficiency in detecting deviations in the behavior of IoT devices.

Key words: Internet of Things (IoT), artificial intelligence, anomaly detection, attack detection, intrusion detection.

70 pp., 1 tabl, 30 figures, 50 sources.

ЗМІСТ

ВСТУП

Забезпечення безпеки інформаційного середовища «Smart City» є критично важливим у сучасних умовах зростання кіберзагроз та підвищеної цифровізації міської інфраструктури. У сучасному розумному місті важливу роль відіграють IoT мережі, які забезпечують взаємодію між різноманітними пристроями та системами. Аналіз звітів компаній, які займаються забезпеченням кібербезпеки, демонструє низький рівень безпеки IoT інфраструктури. Оскільки пристрої Інтернету речей використовуються як в приватному секторі, так і на інших об'єктах, в тому числі об'єктах критичної інфраструктури, проблема безпеки цих пристроїв та безпеки IoT систем набуває важливого значення. Через чисельні вразливості пристрої Інтернету речей часто стають мішенню для кіберзлочинців.

Актуальність роботи полягає у вирішенні важливого завдання – розроблення програмного забезпечення для виявлення мережевих атак на IoT системи у контексті інформаційного середовища «Smart City».

Дослідженню розумних міст присвячено чимало праць зарубіжних та вітчизняних науковців. Зокрема, Д. Кумар, К. Лаудон, А. Маліп, О. Сахно, О. Палка, Л. Дмитроца розглядають методи, засоби та інструменти для формування інформаційно-технологічних платформ гіперскладної системи «Розумне місто». В. Б. Дудикевич, Г. В. Микитин, Л. Л. Бортнік, Т. Р. Стосик висвітлюють проблему безпеки інтернету речей та безпровідних технологій для сегментів інтелектуалізації інфраструктури суспільства. Методи виявлення кібератак в інфраструктурі Інтернету речей на основі машинного навчання широко представлено в роботах К. Бобровнікова, М. Капустян, Д. Денисюк, А. Верма, В. Раві та ін.

Попри наявність сучасних інструментів та методик, кіберзагрози постійно еволюціонують, зловмисники використовують все більш витончені методи для обходу традиційних систем безпеки. Питання інтеграції алгоритмів виявлення мережевих атак на IoT системи у контексті інформаційного середовища «Smart City» на основі машинного навчання залишається недостатньо дослідженим.

Мета роботи - дослідження видів мережевих кібератак на IoT системи «Smart City»; підвищення точності та ефективності їх виявлення на основі моделей виявлення аномалій в поведінці системи та користувачів середовища «Smart City».

Досягнення визначеної мети передбачає вирішення таких **завдань:**

1. Провести аналіз інформаційних та комунікаційних технологій «розумного міста» та визначити місце Інтернету речей в моделі інформаційно-технологічної архітектури «Smart City».
2. Дослідити існуючі методи виявлення мережевих атак на IoT системи та визначити ключові аспекти для захисту в інформаційному середовищі «Smart City».
3. Розробити програмне забезпечення на основі алгоритмів машинного навчання, що дозволить будувати систему IoT для виявлення кібертатак; надати рекомендації щодо підвищення рівня безпеки інформаційного середовища «Smart City».

Об'єкт дослідження – процес взаємодії між різноманітними пристроями та системами розумного міста;

Предмет дослідження – методи та алгоритми виявлення атак на Інтернет речей з використанням технології машинного навчання.

Методи досліджень. Для розв'язання поставлених задач у даній кваліфікаційній роботі використано: методи аналізу та синтезу, метод машинного навчання на основі випадкових лісів, методи об'єктно-орієнтованого програмування, статистичні методи кореляції даних та інтерпретації отриманих результатів.

Наукова новизна одержаних результатів. Удосконалено алгоритми виявлення атак на IoT системи в контексті інформаційного середовища «Smart City» за допомогою методів машинного навчання, а саме, моделей виявлення аномалій в мережевому трафіку інформаційного середовища «Smart City».

Практичне значення отриманих результатів. Розроблено програмне забезпечення для виявлення атак на IoT систему з використанням методу

випадкового лісу та розроблено рекомендації щодо підвищення захисту інформаційного середовища «Smart City».

Результати дослідження було апробовано на Міжнародній науково-практичній конференції «Відновлення України у повоєнні часи: виклики, стратегічні пріоритети, ресурсне забезпечення, потенціал майбутнього розвитку», яка проходила 10-11 жовтня 2024 року на базі Донецького національного університету імені Василя Стуса. (Доповідь на тему «Сучасні загрози та кібератаки на важливі інформаційні структури» https://jrupp.donnu.edu.ua/issue/view/547).

Структура й обсяг кваліфікаційної (магістерської) роботи. Робота викладена на 70 сторінках та складається з анотації, змісту, вступу, трьох розділів, висновків, списку використаних джерел та лістингу програми. У роботі 30 рисунків, 1 таблиця, список використаних джерел містить 50 найменувань, додаток.

РОЗДІЛ 1.
АНАЛІЗ СУЧАСНОГО СТАНУ ПРОБЛЕМ БЕЗПЕКИ ІОТ СИСТЕМ У КОНТЕКСТІ ІНФОРМАЦІЙНОГО СЕРЕДОВИЩА «SMART CITY»

1.1 IoT системи «Smart City»

Для оптимізації різних аспектів міського життя та підвищення якості життя мешканців у сучасних містах активно впроваджуються інформаційно-комунікаційні технології та інноваційні підходи. Інтеграція цифрових технологій, комунікаційних технології та аналітики даних в інфраструктуру міста лежить в основі концепції «Smart City» - розумного міста і має на меті створення ефективного та дієвого сервісного середовища, яке покращує якість життя в місті та сприяє його сталому розвитку. На вектори формування «розумних міст» суттєво впливає динаміка формування потреб громадян, яка лежить в основі показників територіальних та загальносвітових індексів «Smart City».

Значна частина досліджень в царині «розумних міст» базується на технологіях, що сформовані завдяки корпоративним організаціям Amazon, Cisco, GE, IBM, Intel, Microsoft та Oracle [1] тощо, при цьому увага зосереджена на розробці хмарних платформ та застосунків для проектів класу «Smart City».

Базовими інструментами формування системи «Smart City» є інформаційні та комунікаційні технології, які «можуть варіюватися від простих обчислювальних пристроїв, зокрема настільних комп'ютерів, ноутбуків та смартфонів, до більш складних систем, наприклад, мережевих пристроїв та програмного забезпечення, призначеного для конкретних завдань, зокрема аналізу даних, управління проектами або кібербезпеки» [2].

Інформаційні технології, спроможні забезпечити ефективне функціонування «Smart City» зручно представити у вигляді множини [3]:

$$IT = \{IT_{IoT}, IT_{BD}, IT_{BL}, IT_{CL}, IT_{AI}, IT_{ML}, IT_{OD}\}, \quad (1.1)$$

де

- IT_{IoT} – Інтернет речей,
- IT_{BD} – «великі дані»,
- IT_{BL} – блокчейн,
- IT_{CL} – хмарні технології,
- IT_{AI} – штучний інтелект,
- IT_{ML} – машинне навчання,
- IT_{OD} – «відкриті дані».

Важливу роль у системі комунікацій «Smart City» відіграє $IT_{IoT.}$

Завдяки використанню мікроконтролерів, радіомодулів, комунікаційних протоколів IoT-пристроям мають змогу обмінюватись інформацією в інтегрованомуінформаційному середовищі «Smart City» .

Концепція IT_{IoT} спрямована на підвищення ефективності використання ресурсів та підвищення якості послуг при зниженні операційних і управлінських витрат в гіперскладних системах «розумних міст» (рис. 1.1) [4].

До компонентів IT_{IoT} у «Smart City» входять: фізичні пристрої, датчики та виконавчі механізми. Всі вони покликані забезпечувати перебіг процесів збирання та обміну даними через мережу Інтернет.

Використання IT_{BD} для видобування прихованих знань має широкий спектр застосування у «Smart City»: реалізація розподілених реєстрів (Distributed Ledger Technologies, DLTs), супроводження процесів прийняття рішень.

В «розумних містах» IT_{BL} активно впроваджуються в «розумній охороні здоров'я», «розумному транспорті» і «розумних» ланцюгах постачання, забезпечуючи сталість розвитку «розумних міст» завдяки незмінності, доданої до ланцюжків даних, інформації, анонімності та конфіденційності даних.

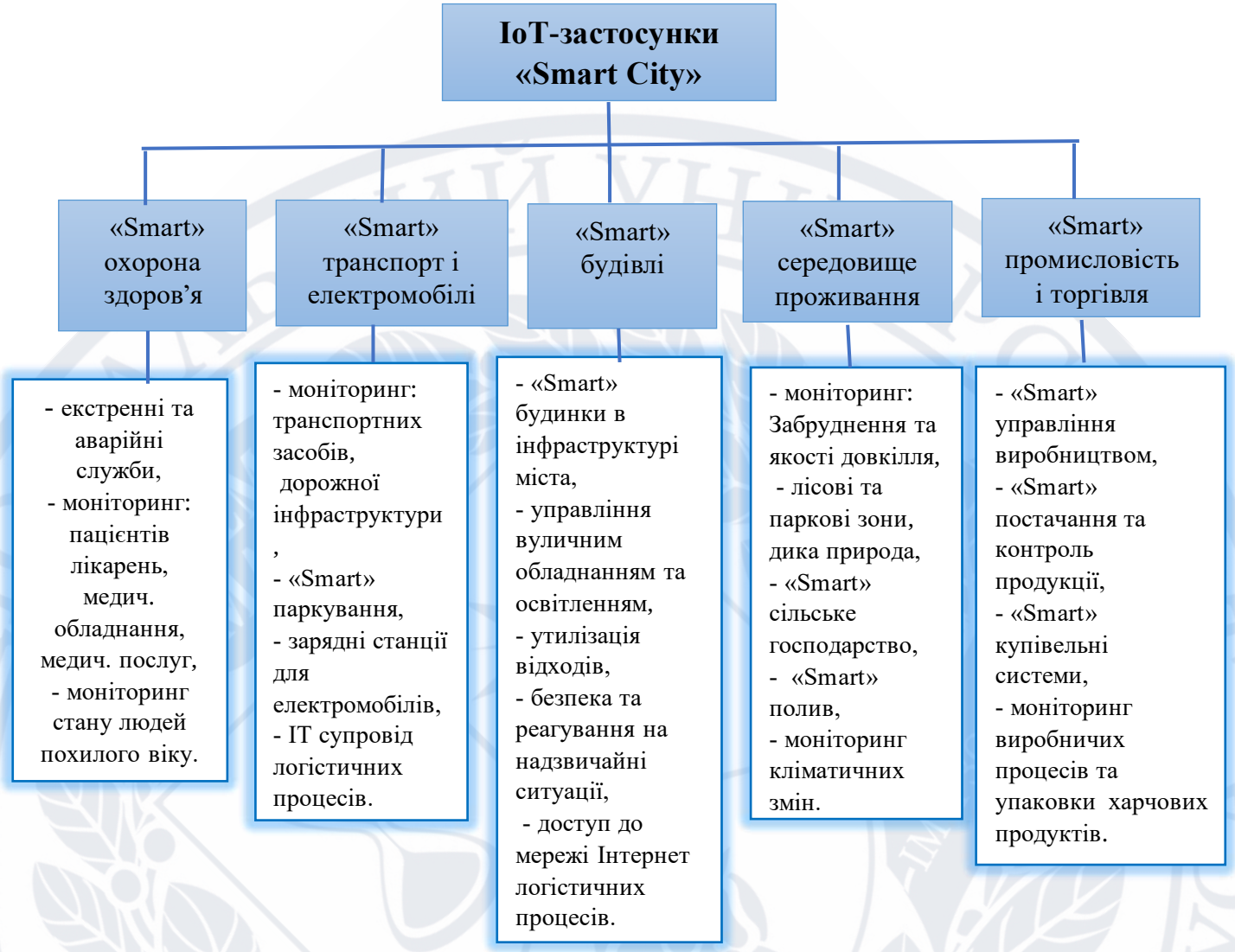

Рис. 1.1. Сфери застосування IoT-пристроїв у «Smart City» [4]

У «Smart City» IT_{BL} активно впроваджуються в «Smart» охороні здоров'я, «Smart» транспорті і ««Smart» ланцюгах постачання, шо сприяє забезпеченню розвитку завдяки незмінності доданої до ланцюжків даних інформації, дотримання анонімності та конфіденційності даних.

Важливо забезпечити кожного учасника «Smart City» (рис. 1.2) єдиним інтерфейсом для збору даних та засобами їх зберігання.

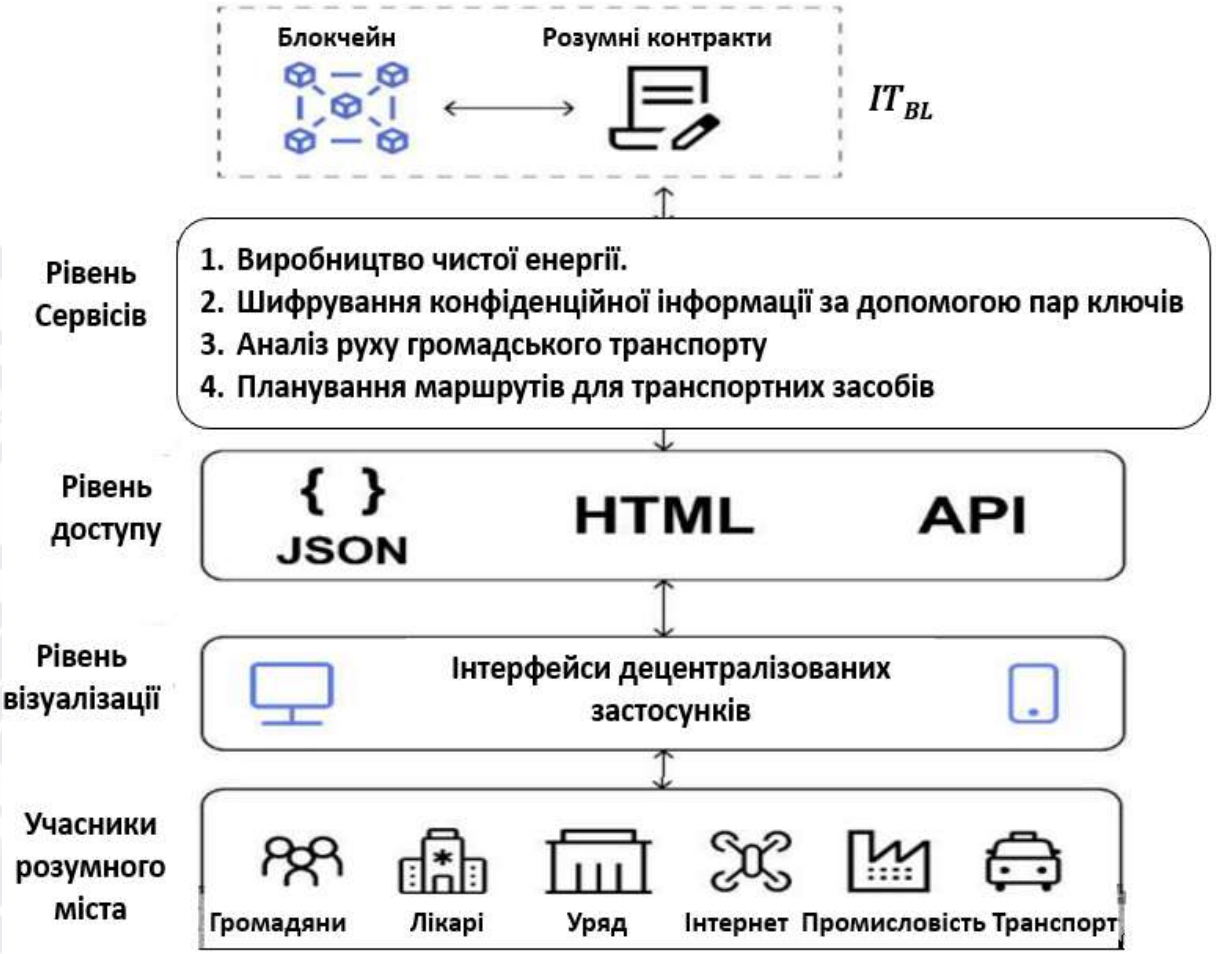

Рис. 1.2. Блокчейн технології в структурі інтерфейсів системи «Smart City»

IT_{BL} сприяє розвитку «розумних міст», підвищуючи продуктивність та безпеку послуг та застосунків.

Розподілена природа IT_{BL} підвищує показники масштабованості та зменшує кількість точок відмови міської інформаційно технологічної архітектури. Водночас покращуються процеси обміну даними завдяки використанню уніфікованих загальних програмних інтерфейсів.

На рис. 1.3 представлена інформаційно-технологічна платформа «Smart City».

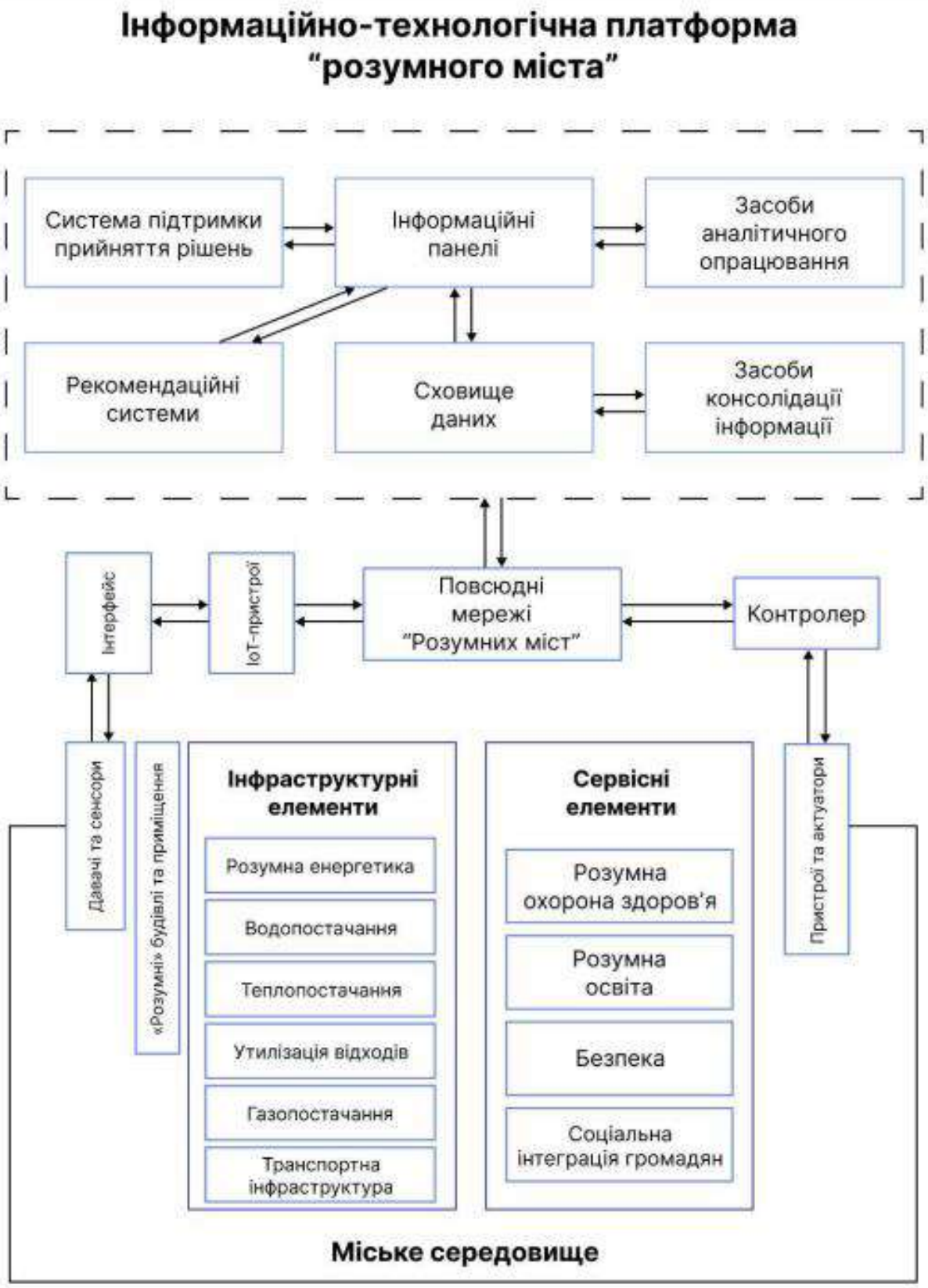

Рис. 1.3. Інформаційні потоки «Smart City» [5]

Подана схема зосереджена на взаємодії міських підсистем між собою

завдяки впровадженню інформаційних та комунікаційних технологій. Представлені в інформаційно-технологічній платформі «потоки даних» як етапи життєвого циклу даних, відображають більшість систем «Smart City».

Система «Smart City» інтегрує інфраструктурні елементи, «Smart» будівлі та приміщення, сервісні елементи.

У складних інтегрованих системах «Smart City» активно впроваджуються та використовуються різні типи давачів та сенсорів на основі IT_{IoT}.

Для збереження, консолідації, аналізу та відображення даних використовуються хмарні інформаційно технологічні платформи. Це забезпечує високоефективний супровід процесів прийняття рішень учасниками, муніципальними адміністраціями, установами та організаціями, провайдерами «розумних» послуг.

Концепція «Smart City» заснована на сучасних передових підходах, високотехнологічних інноваційних технологіях, спрямованих на забезпечення комфортного міського середовища та ефективної реалізації планів розвитку міст.

Модель, яка узагальнює процеси життєвого циклу даних в соціально комунікаційних та фізичних системах гіперскладних систем «Smart City» представлено на рис. 1.4.

До технічних аспектів віднесено: сумісність та структура даних, комунікаційні протоколи, M2M (машино-машинна взаємодія, технологія, яка забезпечує передачу даних між різними пристроями).

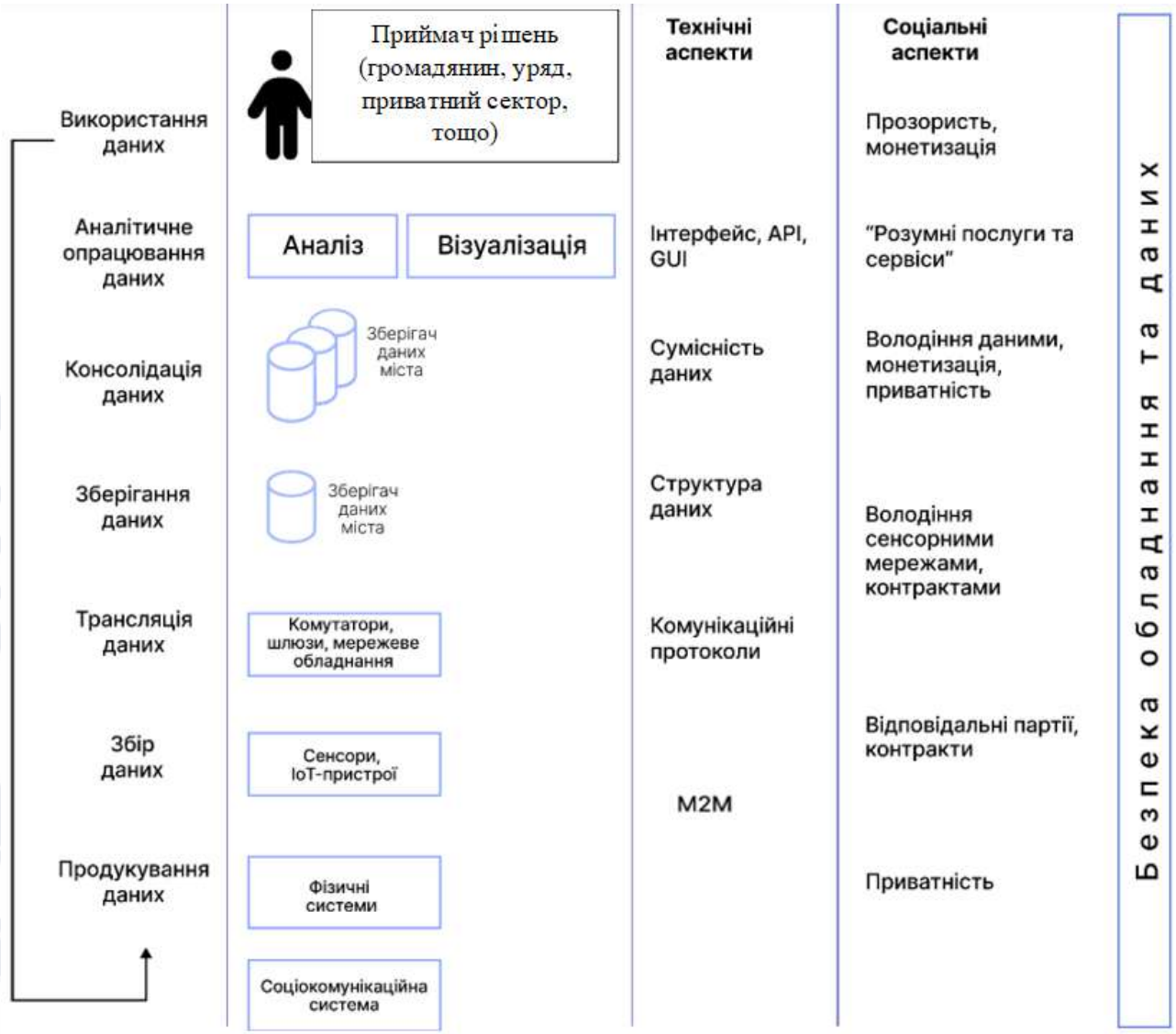

Рис. 1.4. Формування життєвого циклу даних у «Smart City»

Дана модель враховує соціальні аспекти життєвого циклу даних, такі як:

- конфіденційність,
- безпеку,
- прозорість монетизації,
- технічні особливості, зокрема, підтримка інтерфейсів та сумісності процесів.

1.2. Вразливості та загрози безпеці IoT систем в "Smart City"

«Розумне місто» — це концепція, яка передбачає широке використання інформаційно-комунікаційних технологій (ІКТ) для управління електропостачанням, комунальними послугами та транспортною інфраструктурою з метою економії витрат, зменшення шкідливого впливу на навколишнє середовище та прискорення усунення неполадок.

Інтернет речей (IoT) є фундаментальною складовою розвитку розумних міст. IoT системи розвивають свої переваги через ідею взаємодії з інтелектуальними системами за допомогою оснащення електронних об'єктів вбудованими датчиками для збору даних і реагування на них через мережу, в тому числі з використанням різних технологій бездротового зв'язку, хмар і аналізу. Розумні пристрої, які варіюються від побутових приладів до загальноміських інфраструктурних систем, є невід'ємною частиною покращення життя в місті за допомогою технологій. Однак підключення цих пристроїв до мережі також створює нові виклики для безпеки.

Дослідники виділяють два типи найбільш поширених атак на пристрої Інтернету речей [22]. Перший тип – атаки Bashlite, спрямовані на фреймворки Linux, за яких здійснюється передача даних через відкритий telnet. Другий тип – атаки Mirai, спрямовані на знаходження слабких місць в гаджетах IoT, які можуть бути атаковані через їх IP- та Mac-адреси, після чого на зламані пристрої завантажується зловмисне програмне забезпечення.

Наприклад, великого розголосу набула атака ботнету Mirai. На формування бот-мереж суттєво впливають такі особливості інфраструктури інтернету речей як недостатня обчислювальна потужність, обмеження живлення та висока щільність вузлів IoT. Саме через ці специфічні особливості жертвами атаки ботнету Mirai стало безліч недорогих пристроїв, де стояли стандартні паролі [6]. Завершенням стала розподілена відмова в обслуговуванні (DDoS) постачальника системи доменних імен (DNS) Dyn, який є частиною інтернет-інфраструктури багатьох американських гігантів, в результаті чого

клієнти понесли величезні фінансові втрати.

У 2015 році почалися перші російські кібератаки на енергетичну інфраструктуру. Оскільки сьогодні майже всі прилади залежать від електроенергії, її відключення може спричинити справжній хаос у різних галузях діяльності. Ще одним відомим прикладом стало поширення цілеспрямованого шкідливого коду для підриву важливих промислових процесів у 2017 році. Відомі випадки поширення шкідливих програм для прихованого майнінгу на інфікованих системах управління електростанціями.

Російські кібергрупи активізувались незадовго до фізичного вторгнення. Після кібератаки у січні 2022 року в мережі щонайменше двічі з'являлися масиви даних, які містили особисту інформацію українців, отриману шляхом компіляції різних баз приватних компаній.

З початку повномасштабної війни росія щомісяця проводить сотні кібератак, левова частка яких припадає на критичну інфраструктуру. Приклад тому є постійні намагання зруйнувати українську енергетику, і не лише ракетами та дронами, артудари посилюються кібератаками. Тільки в 2023 році, згідно даних Держспецзв'язку, було зафіксовано близько 55 кібератак на енергооб'єкти [7].

Кількість кібероперацій росії зростає, водночас кількість критичних кібероперацій зменшується. У 2022 році Держспецзв'язок реєстрував 2194 кіберінциденти, з яких 1048 були серйозними або критичними. У 2023 році їх було 2554, з яких лише 367 були серйозними. Однак навіть один промах може дорого коштувати. Наприклад, у грудні 2023 року найбільший український оператор «Київстар» зазнав потужної хакерської атаки, внаслідок якої мільйони абонентів втратили доступ до послуг мобільного зв'язку та інтернету [8].

Об'єктами атак кіберзлочинців часто стають пристрої Інтернету речей. Ризики, що пов`язані з інтернетом речей виходять на новий рівень, оскільки через сумісність, автономність прийняття рішень з`являються різні лазівки в безпеці та потенційні вразливості.

Інтернет речей дійшов до того рівня, коли може з'єднати різні простори

(цифровий та фізичний простір, наприклад), де різні датчики взаємодіють з фізичним простором. Ці датчики вже повністю використовуються практично у всьому, від якихось іграшок до систем охорони здоров'я, промислового сегмента. Всі підключені пристрої можуть бути потенційним входом до Інтернет-інфраструктури або особистих даних, і можуть збирати дані про своє оточення та користувачів що робить їх вразливими цілями для кібератак, особливо щодо порушення конфіденційності та несанкціонованого доступу. Дані з таких пристроїв можуть бути проаналізовані, після чого будуть використані. Аналіз цих даних дозволяє створювати невидимі посилання, які можуть бути спрямовані на конфіденційність окремих осіб або організацій.

Інтеграція великої кількості раніше розрізнених інфраструктурних систем в єдине мережеве середовище розширює площу цифрових атак для кожної взаємопов'язаної організації. Ця розширена поверхня атаки збільшує можливості для зловмисників використовувати вразливість для початкового доступу, переміщатися між мережами і спричиняти каскадні, міжгалузеві порушення роботи інфраструктури або іншим чином загрожувати конфіденційності, цілісності та доступності даних, систем і мереж організації.

Крім того, в результаті інтеграції більшої кількості систем і посилення зв'язку між підмережами, мережеві адміністратори і співробітники служби безпеки можуть втратити видимість колективних системних ризиків. Потенційна втрата видимості включає компоненти, що належать і експлуатуються постачальниками, які надають свою інфраструктуру як послугу для підтримки інтеграції.

Вразливості в ланцюгах постачання ІКТ, які виникають через неналежні практики безпеки або можуть бути навмисно створені для зловмисних цілей, призводять до: крадіжки даних та інтелектуальної власності; втрати довіри до цілісності системи «Smart City»; збою системи чи мережі через порушення доступності операційної технології.

Постачальники ІТ-послуг для «Smart City» також можуть мати доступ до величезних обсягів конфіденційних даних з різних громад для підтримки

інтеграції інфраструктурних послуг, включаючи конфіденційну державну інформацію та інформацію, що ідентифікує особу, що може стати привабливою мішенню для зловмисників.

Автоматизація інфраструктурних операцій у середовищі "розумного міста", з одного боку, забезпечує узгодженість і швидкість стандартизованих операцій, з іншого, може також створювати нові вразливості, спричинені збільшенням кількості віддалених точок входу в мережу і мережевих з'єднань, вразливих до компрометації. Обсяг даних і складність автоматизованих операцій можуть зменшити видимість системних операцій і потенційно перешкоджати реагуванню на інциденти в режимі реального часу.

Інтеграція штучного інтелекту та різноманітних методів машинного навчання, складних цифрових систем також може призвести до появи нових векторів атак і додаткових вразливих мережевих компонентів через зниження загальної прозорості роботи мережевих пристроїв, оскільки ці системи приймають і виконують оперативні рішення на основі алгоритмів, а не людських суджень.

Інтеграція IT- та OT-систем у розумних містах створює складний кіберфізичний ландшафт, в якому фізична інфраструктура є вразливою до кібератак. Інтеграція передових цифрових технологій із застарілими системами може створити нові потенційні вектори для атак, що призводить до неузгодженості політик безпеки і вразливостей. Безпека розумних міст ставиться під загрозу через відсутність універсальних стандартів для пристроїв Інтернету речей, що призводить до появи нових вразливостей з кожним пристроєм, доданим до мережі. Забезпечення сумісності при збереженні високих стандартів безпеки вимагає ретельного балансу, модернізації або заміни застарілих систем, які не можуть відповідати сучасним вимогам безпеки.

Для забезпечення захищеного обміну інформацією в багаторівневій архітектурі розумних міст актуальним є питання безпеки безпровідних технологій зв’язку. Комплексна модель безпеки безпровідних технологій зв’язку, представлена на рис. 1.5.

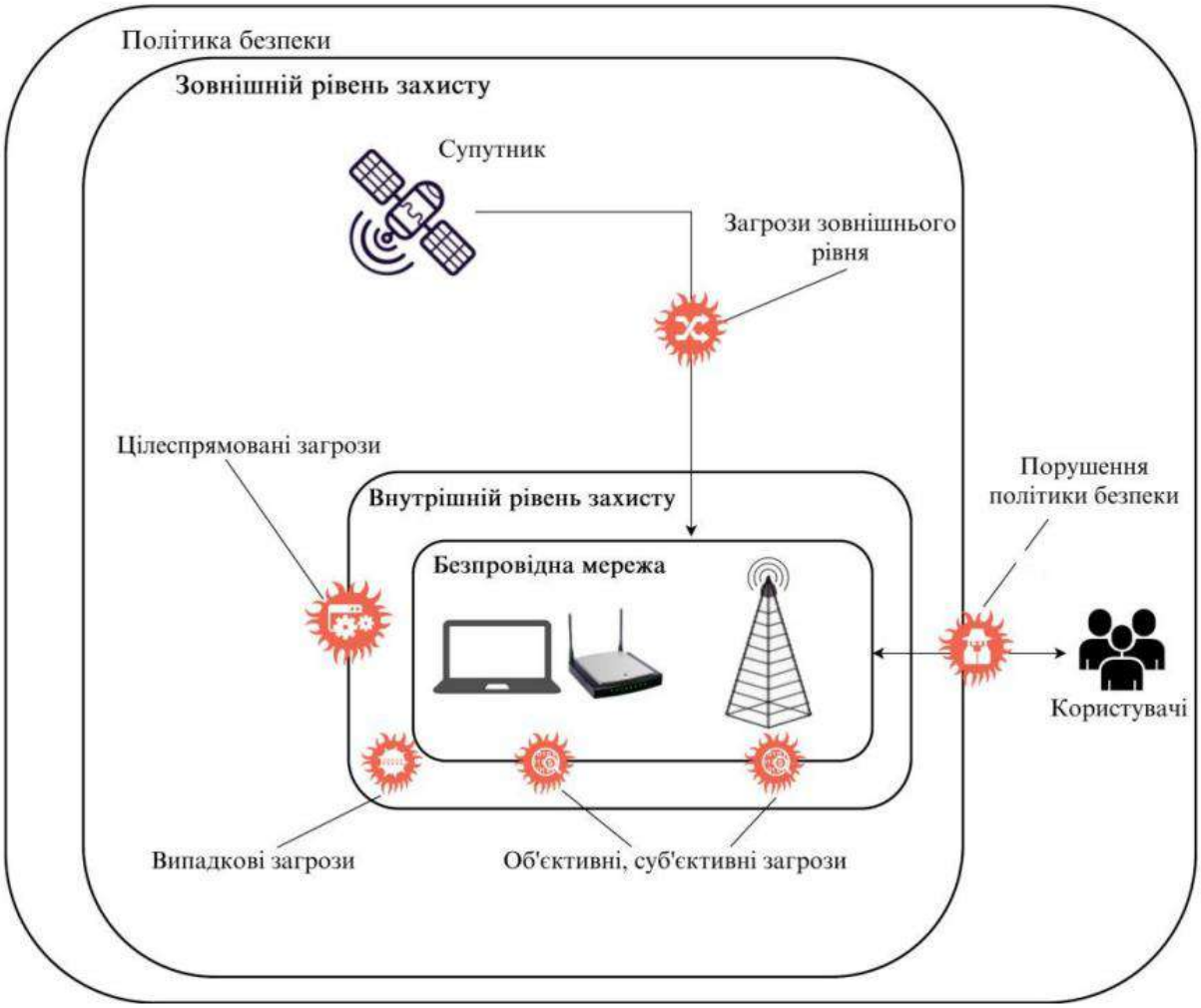

Рис. 1.5. Комплексна модель безпеки безпровідних технологій зв’язку [10]

Модель включає:

1) єдину функціональну структуру захисту інформації, представлену периметрами зовнішньої, внутрішня безпеки та політикою інформаційної безпеки;

2) спеціалізовану структуру комплексної системи безпеки, зумовлену комплексом загроз для об’єктів інфраструктури суспільства: «Smart» екології, «Smart» освіти, «Smart» енергетики, «Smart» транспортної системи [10].

Основні завдання безпеки інтелектуальних технологій пов’язані з протидією цілеспрямованим загрозам. Наразі, немає єдиного рішення, яке могло б захистити всі пристрої IoT від усіх типів загроз. Водночас, дослідники виокремлюють загальні стратегії, націлені на зменшення ризиків, спричинених пристроями Інтернет речей .

Перша стратегія ґрунтується на належному налаштуванні та забезпеченні захищеності всіх пристроїв IoT, що включає налаштування облікових записів користувачів, створення надійних паролів, налаштування брандмауерів, використання та регулярне оновлення антивірусного програмного забезпечення, регулярні встановлення оновлень безпеки.

Друга стратегія передбачає підключення пристроїв IoT до корпоративних або глобальних мереж через безпечні бездротові мережі, що убезпечує системи від атак забезпечує захист конфіденційних даних.

Третя стратегія заснована на постійному моніторингу та обізнаності щодо загроз безпеки пристроїв IoT і впровадженні відповідних профілактичних заходів захисту від атак.

Комплексне використання даних стратегій забезпечить правильне налаштування IoT пристроїв, захищеність від атак, та захист конфіденційної інформації.

1.3. Аналіз методів виявлення мережевих атак на IoT системи у контексті інформаційного середовища «Smart City»

Глобальне зростання кіберзагроз, спрямованих на системи операційних технологій (OT), у поєднанні з їх інтеграцією в системи "розумного міста", розширює сферу атак і збільшує потенційні збитки від порушень. Особливо вразливими є розумні міста до загроз безпеки через розгалужену мережу взаємопов'язаних пристроїв. Ці загрози включають витік даних, несанкціонований доступ, атаки на відмову в обслуговуванні тощо.

Мережевий зв'язок є ключовим елементом у розумних містах. Всі сучасні системи виявлення кібератак проводять моніторинг або головних комп'ютерів, або мережевих з'єднань, щоб зібрати дані про вторгнення.

Програмні або апаратні системи виявлення та запобігання вторгненням, що забезпечують мережну та комп'ютерну безпеку - IDS/IPS (Intrusion Detection and

Prevention System). IDS – це пасивна система виявлення, яка у режимі реального часу аналізує весь трафік і за необхідності повідомляє про можливі загрози. IDS здатна виявити загрозу та сповістити адміністратора, а подальші дії має робити вже він сам. Для ефективної роботи систем виявлення вторгнень (IDS) необхідно зменшити кількість потоків, пов'язаних з IoT.

Виділяють три типи IDS за механізмом аналізу:

1. Сигнатурні IDS – система виявлення вторгнень, що сильно нагадує звичний антивірус – вона також аналізує трафік та зіставляє отримані пакети з базою даних сигнатур. Для забезпечення ефективності роботи цієї системи базу даних сигнатур необхідно регулярно оновлювати. Ключовим недоліком вважається те, що якщо база даних з якихось причин виявиться тимчасово недоступною, то така IDS не зможе забезпечити виявлення загроз.

2. IDS, засновані на аномаліях. У цьому випадку використовується технологія машинного навчання – система аналізуватиме роботу мережі та порівнюватиме її з аналогічним періодом у минулому. Аномалії можуть бути статистичними, протокольними або на рівні трафіку – все це система виявлення загроз здатна виявляти та припиняти, за умови, що у минулому машинному навчанні приділялося достатньо часу та уваги.

3. IDS на правилах. Адміністратор може вручну прописати складні правила IDS, які дозволять виявляти погрози за непрямими чи прямими ознаками. Налаштування такої системи вимагає значно більше часу та експертності, але на практиці дозволяє отримати вкрай високий рівень захисту.

Розглядають і більш сучасні технології аналізу пакетів UTM NGFW та DPI.

UTM – універсальний пакет утиліт, який містить кілька дрібних модулів захисту, від поштових фільтрів та проксі-серверів доVPN та IDS. По суті це комплекс, що гарантує багаторівневе відстеження та ліквідацію загроз.

NGFW та DPI – це вже міжмережевий екран нового покоління. NGFW є платформою мережевої безпеки, у складі якого, крім самого міжмережевого екрану, є також традиційний брандмауер з додатковими можливостями

фільтрації. Що ж до DPI, це технологія глибокого аналізу пакетів, що дозволяє перехоплювати пакети, що містять загрози.

Моделі виявлення на основі підписів (Signature-based Detection) покладаються на базу даних сигнатур, яка містить унікальні ідентифікатори відомих загроз.

Моделі виявлення на основі аномалій (Anomaly-based Detection) вивчають нормальну поведінку системи, мережі або користувача, створюючи статистичну модель або профіль.

Серед ефективних методів захисту від мережевих атак на IoT системи варто виокремити:

- оновлення та впровадження нових стандартів та політик захисту,
- аналітика сучасних загроз IoT: схема зондування на основі політик для IoT (RealAlert); розвідка загроз (Threat intelligence)
- використання штучного інтелекту в якості адаптивного аналізу поведінки в мережі (Behavior Analyst),
- blockchain для безпеки IoT.

Ці методи спрямовані на захист цілісності даних, забезпечення конфіденційності та підтримання загальної безпеки мереж Інтернету речей у розумних містах.

Схема RealAlert може точно оцінити довіру до сенсорних вузлів, а також до даних в IoT, однак дана система надто ресурсо затратна для процесу зберігання та обробки даних. Розвідка загроз (Threat intelligence) надходить з різних джерел і на основі аналізу потоків даних в режимі реального часу, звітів і баз даних, які детально описують поточну діяльність, пов'язану з кіберзагрозами, системи безпеки Інтернету речей можуть впроваджувати більш вдосконалені системи виявлення вторгнень (IDS) і моделі предиктивного аналізу загроз. Таким чином, розвідка загроз підвищує стійкість мереж Інтернету речей до мінливого ландшафту кіберзагроз.

Моделі виявлення на основі машинного навчання (Machine Learning-based Detection) використовують алгоритми машинного навчання для аналізу великих

обсягів даних та виявлення складних типів кібератак. Використовують статистичні тести для виявлення відхилень та застосовують алгоритми машинного навчання для побудови моделей нормальної поведінки. Засоби машинного навчання для виявлення аномалій в мережі та виявлення хибно позитивних спрацювань демонструють досить високу точність.

Blockchain Blockchain не нова технологія в мережевій безпеці, однак вона має потенціал для використання в мережі розумного міста, оскільки пропонує децентралізований підхід до безпеки IoT, забезпечуючи прозорість, цілісність даних і захист від несанкціонованого доступу. Технології blockchain особливо корисні для управління ланцюгом поставок і безпечних систем голосування, де цілісність і прозорість мають першорядне значення. Однак масштабованість і ресурсо-затратність технологій blockchain залишаються проблемами для широкого застосування IoT.

Гібридні моделі виявлення (Hybrid Detection Models) поєднують переваги різних підходів, таких як сигнатурний аналіз, аналіз аномалій, аналіз поведінки та машинне навчання. Вони можуть використовувати різні методи для різних типів даних або різних етапів атаки і мають наступні переваги:

- Підвищення точності та повноти виявлення за рахунок використання різних методів.
- Зниження хибнопозитивних спрацювань шляхом комбінування результатів різних моделей.
- Адаптивність до різних типів атак та змін у поведінці системи.

Багато сучасних систем виявлення кібератак використовують гібридний підхід, поєднуючи різні технології та методи для забезпечення максимального захисту. Складні моделі, такі як моделі на основі глибокого навчання, можуть вимагати значних обчислювальних ресурсів. Необхідно враховувати доступну обчислювальну потужність при виборі моделі. Також потрібно враховувати показники ефективності, такі як рівень виявлення (Detection Rate) - відсоток реальних атак, які успішно виявляються моделлю та рівень хибнопозитивних

спрацьовувань (False Positive Rate) - відсоток легітимної активності, яку модель помилково ідентифікує як атаку.

Важливо вибрати модель з високим рівнем виявлення, щоб мінімізувати ризик пропуску реальних атак. Високий рівень хибнопозитивних спрацьовувань може призвести до перевантаження фахівців з безпеки та відволікання їх від реальних загроз. Модель повинна легко інтегруватися з іншими інструментами безпеки, такими як SIEM-системи, системи управління інцидентами та системи захисту кінцевих точок.

Висновок до розділу І

Проаналізувавши сучасні підходи та методи виявлення мережевих атак на IoT системи дозволив прийти до висновку, що не існує універсальної моделі, оптимальний вибір залежить від конкретної ситуації. Часто найкращим рішенням є комбінація різних моделей для досягнення максимальної ефективності. Також моделі виявлення кібератак повинні постійно оновлюватися та адаптуватися до нових загроз. Для розумних міст з великою інфраструктурою та високим рівнем ризику, захист екосистеми Інтернету речей потребує багатогранного підходу. Хоча кожен метод має свої сильні сторони та обмеження, ефективна інтеграція цих технологій має важливе значення для захисту від загроз, що розвиваються. З метою запобігання різноманітним загрозами, включаючи нові та невідомі атаки, оптимальним вибором може бути гібридна модель, що використовує комбінацію традиційних і передових методів безпеки. Оскільки ініціативи «розумного міста» продовжують зростати, зростатиме потреба в інноваційних і масштабованих рішеннях безпеки, які можуть адаптуватися до складної та динамічної природи середовищ Інтернету речей.

РОЗДІЛ 2
ДОСЛІДЖЕННЯ МЕТОДІВ ВИЯВЛЕННЯ КІБЕРАТАК В ІНФРАСТРУКТУРІ ІНТЕРНЕТУ РЕЧЕЙ НА ОСНОВІ МАШИННОГО НАВЧАННЯ

2.1. Огляд існуючих рішень для захисту IoT систем

Нині існує багато підходів до виявлення кібератак в інфраструктурі Інтернету речей. Розглянемо найбільш поширені.

Моделі виявлення на основі підписів (Signature-based Detection) покладаються на базу даних сигнатур, яка містить унікальні ідентифікатори відомих загроз. Ці сигнатури можуть бути фрагментами коду, специфічними сигнатури мережевого трафіку, хешами файлів або іншими характеристиками, що відрізняють шкідливу активність від легітимної. Коли система виявлення аналізує мережевий трафік, файли або логи, вона порівнює їх з базою сигнатур. Якщо виявляється збіг, це свідчить про можливу кібератаку. Важливим аспектом є регулярне оновлення бази сигнатур, оскільки кіберзлочинці постійно розробляють нові методи атак.

Моделі виявлення на основі аномалій (Anomaly-based Detection) вивчають нормальну поведінку системи, мережі або користувача, створюючи статистичну модель чи профіль. Можуть включати такі параметри, як обсяг мережевого трафіку, частота доступу до файлів, типові команди, що виконуються користувачем, та інші характеристики. Будь-яке відхилення від цієї норми, яке перевищує встановлені пороги, розглядається як потенційна аномалія.

Моделі виявлення на основі поведінки (Behavior-based Detection) фокусуються на аналізі поведінки користувачів, програм або процесів у системі. Вони шукають підозрілі дії, які можуть свідчити про кібератаку, навіть якщо ці дії не відповідають жодній відомій сигнатурі. Наприклад виконання команд, які рідко використовуються звичайними користувачами, отримання доступу до

файлів або директорій, які зазвичай не використовуються даним користувачем або програмою, виявлені спроби підключення до невідомих або підозрілих IP-адрес або доменів. Такі моделі застосовують евристичні правила для виявлення підозрілої поведінки, проводять аналіз журналів подій для виявлення незвичайних дій, та можуть виконувати підозрілі файли або програми у ізольованому середовищі для аналізу їхньої поведінки.

Моделі виявлення на основі машинного навчання (Machine Learning-based Detection) використовують алгоритми машинного навчання для аналізу великих обсягів даних та виявлення складних сигнатур, які можуть бути пов'язані з кібератаками. Вони можуть навчатися на історичних даних, виявляючи кореляції та аномалії, які неможливо виявити за допомогою традиційних методів.

Багато сучасних систем виявлення кібератак використовують гібридний підхід (Hybrid Detection Models), поєднуючи різні технології та методи для забезпечення максимального захисту, таких як сигнатурний аналіз, аналіз аномалій, аналіз поведінки та машинне навчання. Вони можуть використовувати різні методи для різних типів даних або різних етапів атаки. Наприклад, сигнатурний аналіз може використовуватися для швидкого виявлення відомих загроз, тоді як машинне навчання може застосовуватися для виявлення нових, невідомих атак.

При виборі найбільш ефективної моделі виявлення кібератак фахівці повинні керуватися цілим рядом факторів, які враховують специфіку організації, її інфраструктуру, характер загроз та доступні ресурси.

Якщо основним джерелом даних є мережевий трафік, важливо враховувати його обсяг, швидкість та протоколи. Для високошвидкісних мереж можуть знадобитися апаратні рішення або моделі, оптимізовані для обробки великих потоків даних. Аналіз логів може включати логи безпеки, системні логи, логи додатків тощо. Важливо визначити, які типи логів є найбільш релевантними для виявлення кібератак та які моделі найкраще підходять для їх аналізу. Якщо необхідно аналізувати поведінку користувачів, слід враховувати такі фактори, як кількість користувачів, їх ролі та типова (аномальна) поведінка та підозріла

активність. Моделі на основі машинного навчання можуть бути ефективними для виявлення аномалій у поведінці користувачів. Аналіз файлів може включати виявлення шкідливого програмного забезпечення, аналіз вмісту файлів та виявлення підозрілих змін у файловій системі. Для виявлення відомих загроз, таких як віруси, трояни, черв'яки та інші типи шкідливого програмного забезпечення, найефективнішими є моделі на основі сигнатур. Вони забезпечують швидке та точне виявлення загроз, які вже відомі та описані у базі сигнатур, але для виявлення нових, невідомих загроз, які ще не мають сигнатур, необхідні моделі на основі аномалій або машинного навчання. Ці моделі можуть виявляти підозрілу активність, навіть якщо вона не відповідає жодній відомій сигнатурі. APT-атаки характеризуються тривалим перебуванням зловмисника в системі, використанням легітимних інструментів та складними методами обходу захисту. Для виявлення APT-атак ефективними є моделі на основі поведінки, які аналізують ланцюжки подій та виявляють підозрілі послідовності дій.

Важливо враховувати показники ефективності, такі як рівень виявлення (Detection Rate) - відсоток реальних атак, які успішно виявляються моделлю та рівень хибнопозитивних спрацьовувань (False Positive Rate) - відсоток легітимної активності, яку модель помилково ідентифікує як атаку.

Доцільно вибрати модель з високим рівнем виявлення, щоб мінімізувати ризик пропуску реальних атак. Високий рівень хибнопозитивних спрацьовувань може призвести до перевантаження фахівців з безпеки та відволікання їх від реальних загроз.

Модель повинна легко інтегруватися з іншими інструментами безпеки, такими як SIEM-системи, системи управління інцидентами та системи захисту кінцевих точок.

Не існує універсальної моделі, оптимальний вибір залежить від конкретної ситуації. Часто найкращим рішенням є комбінація різних моделей для досягнення максимальної ефективності. Також моделі виявлення кібератак повинні постійно оновлюватися та адаптуватися до нових загроз.

«Smart City» характеризується великою інфраструктурою та високим рівнем ризику, гіперскладна система «Smart City» стикається з різноманітними загрозами, включаючи нові та невідомі атаки. Тому оптимальним вибором може бути гібридна модель, що поєднує сигнатурний аналіз для швидкого виявлення відомих загроз, аналіз аномалій для виявлення незвичайної активності та машинне навчання для виявлення складних патернів.

Підхід з використання хмарних технологій IT_{CL} та парадигми програмно визначених мереж (SDN) для виявлення та пом'якшення DDoS-атак в бездротових мережах Інтернету речей Н. Раві та С. Шаліні пропонують розглядати на базі дворівневої децентралізованої SDN [19]. В хмарному середовищі розташовано універсальний контролер, підключений до локальних контролерів, які містяться в кожному домені підмережі. Весь трафік мережі контролюється локальними контролерами, які збирають трафік і витягують з нього множину ознак, що можуть вказувати на наявність DDoS-атак. З метою пом'якшення DDoS-атак ефективне використання стратегій окремо для рухомих та нерухомих пристроїв бездротового Інтернету речей. А для виявлення DDoS-атак використовують часткове навчання, extreme learning machine та нейронну мережу прямого поширення.

2.1 Машинне навчання для виявлення аномалій в мережевому трафіку.

Штучний інтелект активно використовується у сучасних містах для управління «розумними» службами та послугами. Потужною галуззю штучного інтелекту IT_{AI} є машинне навчання $IT_{ML.}$ Використання алгоритмів машинного навчання отримують широкий спектр можливостей завдяки впровадженню різноманітних цифрових давачів на базі IoT-пристроїв, зростання наборів та колекцій «великих даних». Завдяки IT_{ML} «розумні» системи мають можливість покращувати «Smart» системи та послуги, приймати обґрунтовані рішення на основі аналітичної обробки та видобування знань з «великих даних», які

отримуються з різних джерел середовища «розумного міста», таких як давачі, соціальні та муніципальні бази даних тощо [11].

Перевагами застосування методів машинного навчання є вища точність виявлення та менша кількість хибних спрацювань, можливість виявлення аномалій та нових ознак атак. Застосунки на базі IT_{AI} дають можливість автоматизувати процеси, скоротити час очікування надання послуг та реагування на запити, адміністративні витрати загалом [12].

Класифікація застосунків, що використовуються у сучасних «розумних містах» на базі штучного інтелекту, представлено на рис. 2.1 [13].

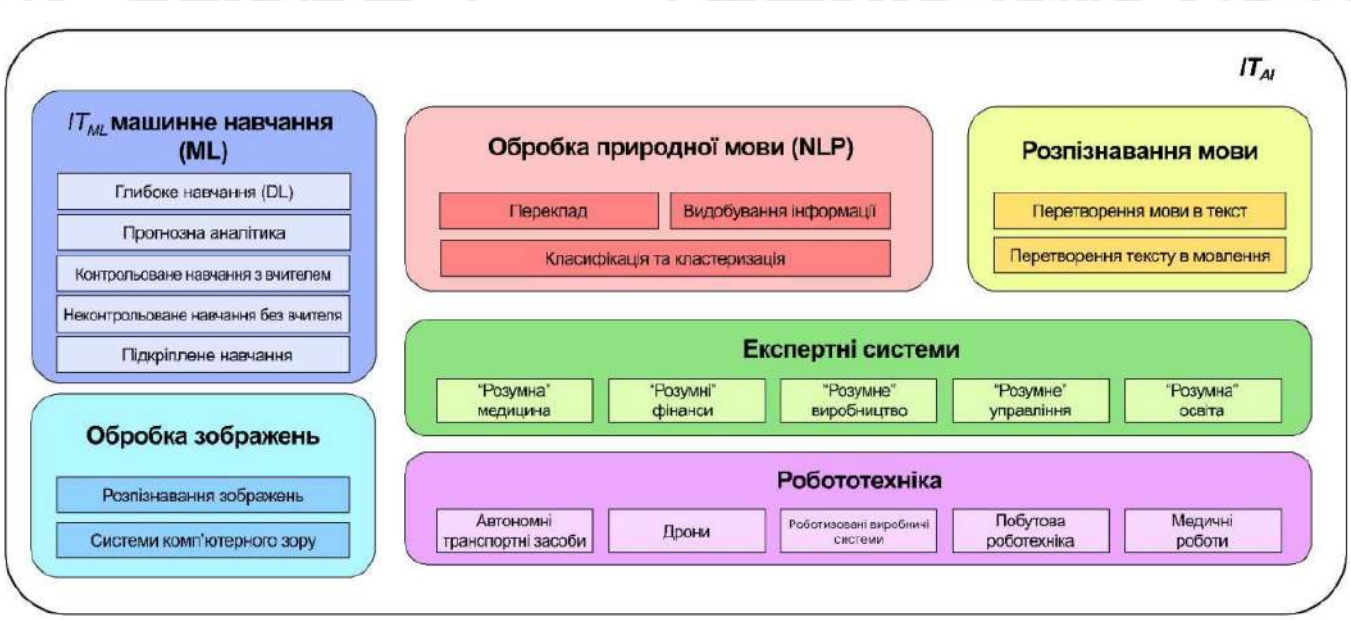

Рис. 2.1. Використання застосунків на базі машинного навчання у «Smart City» [13]

Розрізняють наступні способи навчання алгоритмів IT_{ML} [14]:

- контрольоване навчання з вчителем (для навчання алгоритмів IT_{ML} використовуються попередньо відібраних, так званих мічених, наборів даних, з відповідними помітками "нормальний" або "атака"), зіставляючи вхідні значення з відповідними вихідними метриками, у системах «Smart City» цей підхід зазвичай використовується для задач класифікації та регресії [15]),
- неконтрольоване навчання без вчителя (застосування алгоритмів IT_{ML} для потреб «Smart City» на базі навчання без контролю базується на

дослідженні немаркованих наборів та колекцій даних «Smart City» з метою виявлення прихованих закономірностей і структур даних та подальшою кластеризацією і зменшенням розмірності).

- підкріплене навчання (навчання алгоритмам взаємодії з обчислювальним середовищем та формування оптимальних послідовностей дій на основі винагороди та зворотного зв'язку з метою прийняття у «розумних застосунках» послідовних рішень).

Методи глибокого навчання (deep learning) на основі аналізу даних дозволяють, наприклад, збільшувати виробництво «зеленої» енергії. Нейронні мережі ефективні в управлінні транспортом. У системах «Smart City» IT_{ML} ефективно використовуються для виявлення закономірностей, кореляції та тенденцій у наборах та колекціях даних, забезпечуючи обґрунтування процесів прийняття рішень та проактивного керування на основі даних [16].

Зокрема, застосування алгоритмів IT_{ML} до наборів та колекцій на основі даних «Smart City» дає змогу отримати корисну інформацію та здійснити прогнозування в галузях «розумного» енергоуправління, «розумного» транспорту, «розумної» утилізації відходів, громадській безпеці, залученні містян до процесів міського управління та соціальної інтеграції «розумних» спільнот та громад. Програмно-алгоритмічні комплекси та застосунки на основі алгоритмів IT_{ML} розширюють можливості цифрової трансформації традиційних міських систем в «розумні» адаптивні мережі продукування, постачання та використання ресурсів, що покращують якість надання послуг та якість життя мешканців загалом.

Виявлення вторгнень із застосуванням методів глибокого навчання (DL-IDS) потребує попередньої обробки трафіку IoT середовища та усунення невизначеностей, нормалізації набору даних, прибирання надмірності та заміни відсутніх значень. З цією метою дослідники пропонують вимірювання подібностей у наборі даних проводити з використанням відстані Мінковського [20]. Потім необхідно замінити відсутні значення атрибутів у даних на обчислені значення найближчого сусіда. Це допоможе уникнути зміщення результату

класифікації в бік більш частих записів. Визначаються найближчі сусіди за Евклідовою відстанню, середнім значенням одержаних даних замінюють відсутнє значення.

Алгоритм оптимізації spider monkey (SMO) дозволяє відібрати такі ознаки трафіку, які можуть свідчити про факт вторгнення в IoT-середовище.

Stacked–deep polynomial network (SDPN) надає можливість класифікувати вхідні дані як нормальні або аномальні (можуть вказувати на факти вторгнення, зокрема DoS атаки, атаки user–to–root U2R, атаки probe, атака remote–to–local R2L.

Серед ефективних методів виявлення трафіку атак в мережах Інтернету речей варто розглянути метод, що використовує підхід CorrAUC. Підхід дозволяє: відбирати ознаки, що несуть достатню для виявлення трафіка атак інформацію та на основі цього розробляти новий алгоритм CorrAUC точної фільтрації найбільш ефективних ознак для вибраного алгоритму машинного навчання, який складається з оцінки атрибутів кореляції і поєднується з метрикою Area Under Roc Curve; застосувавши інтегровані методи мультикритеріального аналізу рішень та Shannon Entropy. створювати бієктивні м'які набори для перевірки вибраних ознак для ідентифікації трафіку атак у мережі IoT [21].

Однак середовище «Smart City» є гіперскладним, динамічним, що підвищує рівень вимог до конфіденційності та безпеки даних, потребує забезпечення коректності етичних аспектів використання IT_{ML}. Тому для розкриття потенціалу IT_{ML} для захисту IoT систем у контексті інтегрованих складних систем «Smart City» необхідне формування додаткових переліків, категорій та засобів оцінювання «розумності», адаптивності та відповідності мінливим запитам та потребам жителів.

2.3 . Вибір моделі виявлення кібератак

Однією з важливих проблем, пов’язаних з безпекою в мережах Інтернету речей, є те, що більшість IoT пристроїв мають обмежене живлення й обмежені обчислювальні можливості. IT_{ML} також потребують значних обсягів додаткових обчислювальних ресурсів, що вимагає врахування доступної обчислювальної потужностіь при виборі моделі. До недоліків використання IT_{ML} також слід віднести: зменшення швидкості обробки даних, і як наслідок, значний час відгуку систем виявлення кібератак; неспроможність виявлення та адаптивного реагування на атаки нульового дня та мультивекторні атаки.

Для захисту від кібератак важко застосувати шифрування та аутентифікацію. Саме тому важливу роль у захисті IoT систем відіграє виявлення мережевих проникнень на основі аномалій. Перевага підходу полягає в тому, що коли відбувається атака нульового дня (zero-day attacks), сигнатура атаки не буде розпізнана, проте подальша поведінка мережі буде відхилятися від нормальних моделей трафіку, що дозволить IDS виявити аномалію [23].

У рамках даного дослідження методів виявлення кібератак, було обрано модель виявлення аномалій як основний об'єкт дослідження з кількох причин. Кіберзагрози постійно еволюціонують, тому актуальне використання моделей, які дозволяють виявляти нові, невідомі типи атак, які ще не мають сигнатур. На відміну від моделей на основі сигнатур, які ефективні лише проти відомих загроз, модель виявлення аномалій може виявляти будь-яку активність, що відхиляється від нормальної поведінки системи. Це робить її універсальним інструментом, здатним виявляти широкий спектр кібератак, включаючи атаки нульового дня.

Модель виявлення аномалій здатна адаптуватися до змін у поведінці системи та користувачів, що робить її ефективною навіть у динамічних середовищах. Вона може виявляти аномалії, які виникають внаслідок змін у конфігурації системи, оновлення програмного забезпечення або зміни в поведінці користувачів.

Хоча моделі виявлення аномалій можуть мати вищий рівень false positive спрацьовувань порівняно з моделями на основі сигнатур, сучасні методи машинного навчання та аналізу даних дозволяють знизити цей рівень, підвищуючи точність виявлення. Модель виявлення аномалій може виявляти складні, багатоетапні атаки, які можуть бути непомітними для інших моделей. Вона здатна аналізувати послідовності подій та виявляти підозрілі патерни, які можуть свідчити про кібератаку. Така модель є активною областю досліджень, і нові методи та алгоритми постійно розробляються. Це робить її перспективним напрямком для подальших досліджень та вдосконалень.

Розглянемо модель виявлення вторгнень, яка базується на методах класифікації та попереднього аналізу даних, представлену схемою, яка зображена на рис. 2.2.

Запропонована модель складається з декількох основних етапів:

• Попередня підготовка данних (Preprocessing) – цей етап включає попередній аналіз даних, підготовку їх до роботи з алгоритмами та відображення символьних атрибутів через числові. IoT-пристрої генерують величезну кількість різноманітних даних, які можуть бути використані для виявлення аномалій. Це можуть бути дані сенсорів (температура, вологість, тиск тощо), дані про стан пристроїв (рівень заряду батареї, використання процесора, пам'яті), мережевий трафік (кількість пакетів, протоколи, IP-адреси), логи подій (час входу, виконані команди, помилки) та інші. Дані можуть збиратися безпосередньо з пристроїв, з хмарних платформ, де вони зберігаються, або з мережевих шлюзів.

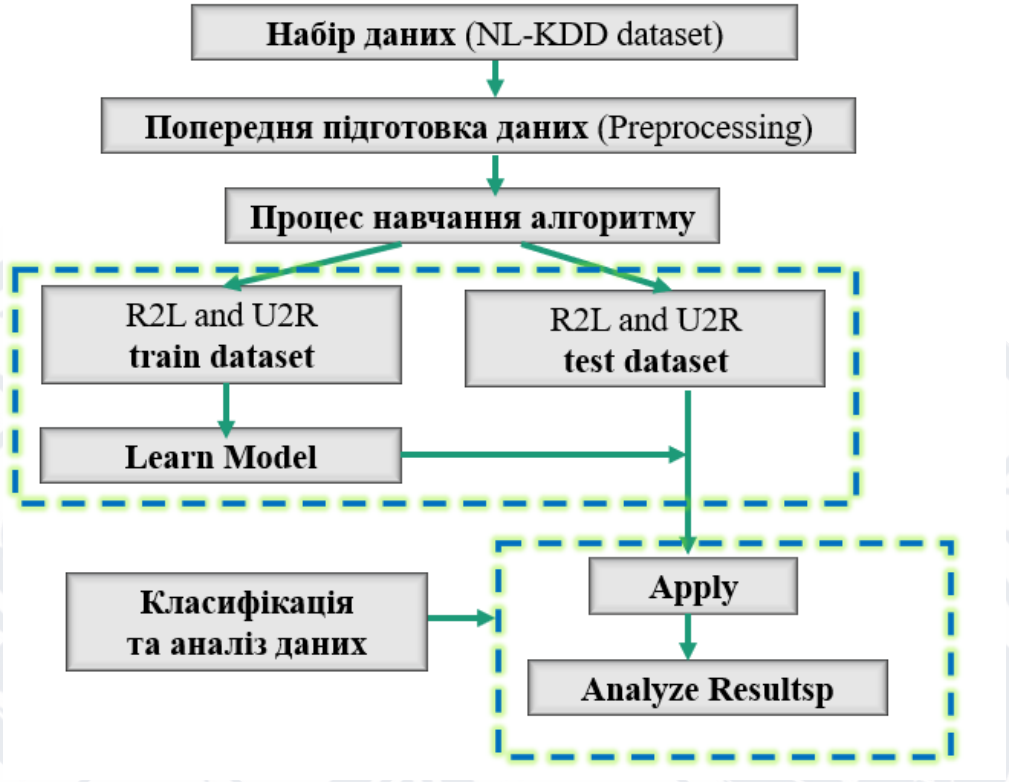

Рисунок 2.2 - Модель виявлення вторгнень

Для збору даних можуть використовуватися різні протоколи, такі як MQTT, CoAP, HTTP тощо. Попередня обробка даних необхідна для того, щоб підготувати їх до подальшого аналізу та навчання моделі. Це може включати такі кроки як очищення даних - видалення шуму, пропусків, дублікатів та інших неточностей, нормалізація - приведення даних до єдиного масштабу або діапазону значень, виділення ознак - вибір найбільш інформативних ознак, які будуть використані для навчання моделі, зменшення розмірності - зменшення кількості ознак для спрощення моделі та прискорення навчання.

Для попередньої обробки даних можуть використовуватися різні методи, такі як фільтрація, агрегація, перетворення, кодування тощо.

• Процес навчання алгоритму (Learn Model, train dataset, test dataset) – на цьому етапі відбувається розбиття даних на тестований та навчальний набір даних, до якого в подальшому застосовується алгоритм навчання. Навчена модель аналізує вхідні дані в режимі реального часу та порівнює їх з навченою моделлю нормальної поведінки. Якщо виявляються відхилення, які

перевищують встановлені пороги, модель генерує алерт. Для виявлення аномалій можуть використовуватися різні методи, такі як порогові значення, статистичні тести, кластеризація, аналіз відстаней тощо. Важливо правильно інтерпретувати результати моделі та відрізняти справжні аномалії від false positive спрацьовувань.

• Класифікація та аналіз даних (Apply, Analyze Results) – застосування алгоритму класифікації та аналіз отриманих результатів.

Модель виявлення аномалій повинна постійно адаптуватися до змін у поведінці пристроїв та нових типів загроз. Це досягається шляхом регулярного налаштування та оновлення моделі.

Дослідження запропонованої моделі виявлення вторгнень здійснено з використанням наступних алгоритмыв класифікації: K–nearest Neighbors (KNN), Gaussian naive Bayes (GNB), Random Forest Classifier (RFC), Support V ector Machine (SV M) дозволило отримати результати, представлені у табл. 2. 1

Дані алгоритми були вибрані в силу того, що в них достатньо різні механізми класифікації й відповідно до цього різні результати аналізу.

Результати роботи алгоритмів подано в метриках Precision, Recall та F1–score, для двох класів – Normal, який відповідає «нормальній» поведінці користувачів, та Attacks, який відповідає двом класам атак R2L та U2R.

Таблиця 2.1 – Результати роботи алгоритмів класифікацій

	Normal			Attacks		
	precision	recall	F1–score	precision	recall	F1–score
KNN	0.99	0.99	0.99	0.95	0.91	0.94
GNB	0.98	0.97	0.97	0.08	0.09	0.08
RFC	0.99	0.99	0.99	0.99	0.93	0.96
SV M	0.93	0.89	0.91	0.82	0.76	0.81

Проаналізуємо представлені алгоритми та оцінимо той, який показав кращі результати і найбільше підходить для вирішення поставлених задач. Зазначимо,

що алгоритм GNB продемонстрував високі результати у виявленні Normal класу, проте у випадку класу Attacks, алгоритм спрацював гірше та виявив незначну кількість атак, про що свідчать показники Precision і Recall, тобто кількість помилок FP та FN достатньо великі, відносно кількості атак. Зазначимо, що дані не збалансовані, тому метрики Precision та Recall є набагато інформативнішими, ніж показник Accuracy алгоритмів.

Найбільш ефективним для виявлення атак R2L та U2R виявився алгоритм Random Forest. Цей алгоритм показав, як добрі результати Precision та Recall, 99% та 93% відповідно, так і маленький відсоток помилок FP та FN, 0.04 % та 0.07 % відповідно. Наступними за оцінкою якості відокремимо K–nearest Neighbors та Support Vector Machine.

Висновок до розділу II

Зловмисники постійно знаходять нові методи для обходу традиційних систем безпеки. Розглянута модель виявлення аномалій дозволяє відслідковувати та виявляти нові, невідомі типи атак, які ще не мають сигнатур. Попри те, що такі моделі можуть мати більше помилкових спрацьовувань порівняно з моделями на основі сигнатур, сучасні методи машинного навчання дозволяють зменшувати відсоток помилок, одночасно підвищуючи точність виявлення атак. Водночас, система виявлення аномалій сама по собі може стати ціллю кібератак. Тому важливо забезпечити її захист від несанкціонованого доступу та маніпуляцій. Модель повинна бути інтерпретованою, щоб фахівці з безпеки могли зрозуміти, чому певна подія була класифікована як аномалія. Використання моделей машинного навчання для виявлення аномалій повинно здійснюватися з дотриманням етичних принципів та з урахуванням можливих наслідків для конфіденційності та безпеки користувачів. Модель виявлення аномалій на основі машинного навчання актуальна універсальністю, адаптивністю до динамічного середовища «Smart City».

РОЗДІЛ 3
РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ РЕАЛІЗАЦІЇ МОДЕЛІ ВИЯВЛЕННЯ КІБЕРАТАК

3.1 Розробка програмного забезпечення на основі алгоритмів машинного навчання

Дерева рішень є одними з кращих і універсальних методів виявлення мережевих кібератак. Підхід передбачає аналіз мережевих даних для виявлення потенційних атак шляхом виявлення кореляції між різними змінними; мінімізацію кореляції між деревами з метою уникнення перенавчання.

Алгоритм машинного навчання Random Forest дозволяє будувати систему IoT для виявлення кібертатак з використанням парадигми хмарних обчислень в інфраструктурі розумного міста. Це дозволяє на основі аналізу нормальної та аномальної поведінки виявляти скомпрометовані пристрої IoT у розподілених туманних вузлах. Здійснюючи моніторинг мережного трафіку, що проходить через кожен такий вузол, після виявлення атак на рівні хмарних, система повідомляє хмарні служби безпеки про необхідність аналізу та оновлення системи [16].

Поєднання алгоритмів Random Forest і Decision Tree може забезпечити достатньо високий рівень точності виявлення аномалій та атак на пристрої Інтернету речей.

На основі проведених експериментальних досліджень, в роботі [20] здійснено комплексний аналіз застосування алгоритмів машинного навчання для виявлення аномальної поведінки в мережах Інтернету речей. Результати оцінки підтверджують, що алгоритм Random Forest досягає найкращої продуктивності з точки зору Accuracy, Precision, Recall, F1-Score and Receiver Operating Characteristic (ROC) curves для всіх застосованих наборів даних. Інші алгоритми машинного навчання працюють з близькою до Random Forest ефективністю,

тому рішення стосовно вибору алгоритму машинного навчання залежить від даних, які підлягають аналізу.

Для аналізу було використано набір даних UNSW-NB15, зібраних Університетом Нового Південного Уельсу (Австралія). Ці дані включали в себе записи різних типів кібератак:

- Fuzzers - атака, яка передбачає надсилання випадкових даних до системи для перевірки її стійкості та виявлення будь-яких вразливостей,
- Analysis - тип атаки, який передбачає аналіз системи для виявлення її слабких місць і потенційних цілей для використання,
- Backdoors - така, яка передбачає створення прихованої точки входу в систему для подальшого використання зловмисником,
- DoS (відмова в обслуговуванні) - атака, мета якої порушити нормальне функціонування системи, зробивши її недоступною для користувачів,
- Exploits - атака, яка використовує вразливість у системі для отримання несанкціонованого доступу або контролю,
- Generic (загальний) - всеохоплююча категорія, яка включає різноманітні типи атак, які не підходять до інших категорій,
- Reconnaissance (розвідка) - атака, яка передбачає збір інформації про цільову систему, наприклад про її вразливості та потенційні точки входу, для підготовки до майбутньої атаки,
- Shellcode - атака, яка передбачає виконання шкідливого коду, як правило, у формі сценаріїв оболонки, на цільовій системі.
- Worm (хробаки) - тип зловмисного програмного забезпечення, яке автоматично поширюється на інші системи, часто завдаючи шкоди в процесі.

Ці дев’ять категорій охоплюють широкий діапазон типів атак, які можна використати для використання системи, і важливо знати про них, щоб захиститися від потенційних загроз безпеці.

Було використано два набори даних: один тренувальний для навчання моделі (рис 3.1), а інший - тестові дані, на яких проводився аналіз ефективності моделі [21].

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	id	dur	proto	service	state	spkts	dpkts	sbytes	dbytes	rate	sttl	dttl	sload	dload
2	1	0.000011	udp	-	INT	2	0	496	0	90909.090	254	0	1,8E+08	0
3	2	0.000008	udp	-	INT	2	0	1762	0	125000.00	254	0	8,81E+08	0
4	3	0.000005	udp	-	INT	2	0	1068	0	200000.00	254	0	8,54E+08	0
5	4	0.000006	udp	-	INT	2	0	900	0	166666.66	254	0	6E+08	0
6	5	0.00001	udp	-	INT	2	0	2126	0	100000.00	254	0	8,5E+08	0
7	6	0.000003	udp	-	INT	2	0	784	0	333333.32	254	0	1,05E+09	0
8	7	0.000006	udp	-	INT	2	0	1960	0	166666.66	254	0	1,31E+09	0
9	8	0.000028	udp	-	INT	2	0	1384	0	35714.285	254	0	1,98E+08	0
10	9	0	arp	-	INT	1	0	46	0	0	0	0	0	0
11	10	0	arp	-	INT	1	0	46	0	0	0	0	0	0
12	11	0	arp	-	INT	1	0	46	0	0	0	0	0	0
13	12	0	arp	-	INT	1	0	46	0	0	0	0	0	0
14	13	0.000004	udp	-	INT	2	0	1454	0	250000.00	254	0	1,45E+09	0
15	14	0.000007	udp	-	INT	2	0	2062	0	142857.14	254	0	1,18E+09	0
16	15	0.000011	udp	-	INT	2	0	2040	0	90909.090	254	0	7,42E+08	0
17	16	0.000004	udp	-	INT	2	0	1052	0	250000.00	254	0	1,05E+09	0
18	17	0.000003	udp	-	INT	2	0	314	0	333333.32	254	0	4,19E+08	0
19	18	0.00001	udp	-	INT	2	0	1774	0	100000.00	254	0	7,1E+08	0
20	19	0.000002	udp	-	INT	2	0	1568	0	500000.00	254	0	3,14E+09	0
21	20	0.000004	udp	-	INT	2	0	2054	0	250000.00	254	0	2,05E+09	0
22	21	0.00001	udp	-	INT	2	0	2170	0	100000.00	254	0	8,68E+08	0
23	22	0.000009	udp	-	INT	2	0	202	0	111111.10	254	0	89777776	0
24	23	0.00001	udp	-	INT	2	0	1334	0	100000.00	254	0	5,34E+08	0
25	24	0.000005	udp	-	INT	2	0	2058	0	200000.00	254	0	1,65E+09	0

Рис. 3.1. Набір вхідних даних про кібератаки [21]

На рис 3.2 та 3.3 представлено фрагмент програмного коду для введення вхідних даних і отриманий результат.

```
# Завантаження даних з CSV файлів
1 usage
def load_data():
    training = pandas.read_csv("UNSW_NB15_training-set.csv")
    testing = pandas.read_csv("UNSW_NB15_testing-set.csv")
    print("training ", training.shape)
    print("testing ", testing.shape)
    return pandas.concat([training, testing]).drop( labels: 'id', axis=1).reset_index(drop=True)
```

Рис. 3.2. Обробка тестових і навчальних даних

```
training  (82332, 45)
testing  (175341, 45)
```

Рис. 3.3. Результат обробки даних

Як результатами маємо 175341 записів тренувальних даних і 82332 тестових даних з 45 параметрами.

Попередня обробка даних представлена на рис. 3.4.

```
#Попередня обробка даних, перетворення категоріальних змінних.
1 usage
def preprocess_data(df):
    for col in ['proto', 'service', 'state']:
        df[col] = df[col].astype('category').cat.codes
    df['attack_cat'] = df['attack_cat'].astype('category')
    print(df.columns)
    return df
```

```
Index(['dur', 'proto', 'service', 'state', 'spkts', 'dpkts', 'sbytes',
       'dbytes', 'rate', 'sttl', 'dttl', 'sload', 'dload', 'sloss', 'dloss',
       'sinpkt', 'dinpkt', 'sjit', 'djit', 'swin', 'stcpb', 'dtcpb', 'dwin',
       'tcprtt', 'synack', 'ackdat', 'smean', 'dmean', 'trans_depth',
       'response_body_len', 'ct_srv_src', 'ct_state_ttl', 'ct_dst_ltm',
       'ct_src_dport_ltm', 'ct_dst_sport_ltm', 'ct_dst_src_ltm',
       'is_ftp_login', 'ct_ftp_cmd', 'ct_flw_http_mthd', 'ct_src_ltm',
       'ct_srv_dst', 'is_sm_ips_ports', 'attack_cat', 'label'],
      dtype='object')
```

Рис. 3.4. Попередня обробка даних

Визначимо основні критерії для виявлення кібератак для забезпечення максимально точного значення recall:

1) dur: тривалість потоку (з'єднання) у секундах.
2) proto: використаний мережевий протокол (TCP, UDP, ICMP тощо).
3) service: використана мережева служба (HTTP, FTP, DNS тощо).
4) state: стан з'єднання (наприклад, встановлено, FIN_WAIT).
5) spkts: кількість пакетів, надісланих від джерела до призначення.
6) dpkts: кількість пакетів, надісланих від призначення до джерела.
7) sbytes: кількість байтів, надісланих від джерела до призначення.
8) dbytes: кількість байтів, надісланих від призначення до джерела.
9) rate: швидкість передачі даних у бітах за секунду.
10) sttl: значення часу життя (Time-to-Live) від джерела до призначення (вказує, скільки переходів може здійснити пакет, перш ніж буде відкинуто).
11) dttl: значення часу життя (Time-to-Live) від призначення до джерела.
12) sload: біти за секунду від джерела.
13) dload: біти за секунду від призначення.

14) sloss: кількість пакетів, втрачених від джерела до призначення.

15) dloss: кількість пакетів, втрачених від призначення до джерела.

16) sinpkt: час прибуття міжпакетного інтервалу джерела (час між пакетами).

17) dinpkt: час прибуття міжпакетного інтервалу призначення.

18) sjit: джиттер (варіація часу прибуття пакетів) джерела.

19) djit: джиттер призначення.

20) swin: розмір вікна TCP джерела (кількість даних, які можна надіслати до підтвердження).

21) stcpb: базовий номер послідовності TCP джерела.

22) dtcpb: базовий номер послідовності TCP призначення.

23) dwin: розмір вікна TCP призначення.

24) tcprtt: час кругового шляху TCP (час, за який пакет переходить до призначення та повертається підтвердження).

25) synack: час встановлення з'єднання TCP (час між пакетами SYN та SYN-ACK).

26) ackdat: час встановлення з'єднання TCP (час між пакетами SYN-ACK та ACK).

27) smean: середній розмір пакетів, надісланих від джерела.

28) dmean: середній розмір пакетів, надісланих від призначення.

29) trans_depth: глибина конвеєра у з'єднанні.

30) response_body_len: розмір тіла відповіді (наприклад, у HTTP).

31) ct_srv_src: кількість з'єднань з тією ж службою та адресою джерела.

32) ct_state_ttl: кількість з'єднань з тим самим станом та значенням TTL.

33) ct_dst_ltm: Кількість з'єднань з тією ж адресою призначення в останньому часовому вікні.

34) ct_src_dport_ltm: кількість з'єднань з тією ж адресою джерела та портом призначення в останньому часовому вікні.

35) ct_dst_sport_ltm: кількість з'єднань з тією ж адресою призначення та портом джерела в останньому часовому вікні.

36) ct_dst_src_ltm: кількість з'єднань з тією ж адресою джерела та призначення в останньому часовому вікні.

37) is_ftp_login: бінарне значення (1, якщо спроба входу FTP, 0 в іншому випадку).

38) ct_ftp_cmd: кількість команд FTP у сеансі.

39) ct_flw_http_mthd: кількість методів HTTP, використаних у потоці.

40) ct_src_ltm: кількість з'єднань з тією ж адресою джерела в останньому часовому вікні.

41) ct_srv_dst: кількість з'єднань з тією ж службою та адресою призначення.

42) is_sm_ips_ports: бінарне значення (1, якщо IP-адреси та порти джерела та призначення однакові, 0 в іншому випадку).

43) attack_cat: категорія атаки (якщо застосовно).

44) label: бінарна мітка (зазвичай 1 для атаки, 0 для звичайного трафіку).

Розглянемо категорії досліджуваних атак та визначимо кількість атак за різними видами (рис. 3.5):

- Нормальний (Normal): Це стосується легітимного, доброякісного мережевого трафіку, який не становить жодної загрози. Це базова лінія, з якою порівнюються інші категорії.
- Бекдор (Backdoor): Бекдор - це метод обходу звичайної автентифікації або отримання несанкціонованого доступу до системи. Зловмисники встановлюють бекдори, щоб підтримувати постійний доступ для зловмисних цілей.
- Аналіз (Analysis): Ця категорія охоплює дії, спрямовані на збір інформації про цільову систему або мережу. Це може включати сканування портів, сканування вразливостей або картографування мережі.
- Фаззери (Fuzzers): Фаззинг передбачає надсилання деформованого або неочікуваного вводу в систему, щоб спробувати викликати збої або виявити вразливості.

- Шеллкод (Shellcode): Шеллкод - це невеликий фрагмент коду, який використовується як корисне навантаження в експлойті. Зазвичай він спрямований на надання зловмиснику командної оболонки на цільовій системі.
- Розвідка (Reconnaissance): Подібно до аналізу, розвідка включає методи збору інформації про ціль перед початком атаки. Це може включати соціальну інженерію, перерахування DNS або пінг-сканування.
- Експлойти (Exploits): Експлойти - це атаки, які використовують відомі вразливості в програмному забезпеченні або системах для отримання несанкціонованого доступу або виконання шкідливого коду.
- DoS (Відмова в обслуговуванні): DoS-атаки спрямовані на перевантаження системи або мережі трафіком, що робить її недоступною для легітимних користувачів.
- Черв'яки (Worms): Черв'яки - це самовідтворюване шкідливе програмне забезпечення, яке може поширюватися від однієї системи до іншої без участі людини.
- Загальний (Generic): Це широка категорія, яка може включати атаки, які не вписуються в інші категорії, або атаки, які мають деякі характеристики кількох категорій.

```
def plot_attack_distribution(df):
    valid_attacks = df[df['label'] == 1]['attack_cat'].value_counts()
    print(valid_attacks)
    plt.figure(figsize=(15, 8))
    bars = plt.bar(valid_attacks.index, valid_attacks.values)
    plt.xlabel("Категорія атаки")
    plt.ylabel("Кількість атак")
    plt.xticks(rotation=0, ha="center", fontsize=12)

    total = len(df[df['label'] == 1])
    for bar in bars:
        height = bar.get_height()
        plt.text(bar.get_x() + bar.get_width() / 2, height, f'{height/total:.1%}', ha='center', va='botto

    plt.show()
```

Рис. 3. 5. Дані про кібератаки

Найменше атак типу Worm – всього 174, в той час як максимальна кількість атак Generic становила 58871. На основі отриманих даних побудували діаграму, представлену на рис. 3.6, яка дозволяє відмітити найпоширеніші види атак, до яких належать Generic, Exploits, Fuzzers.

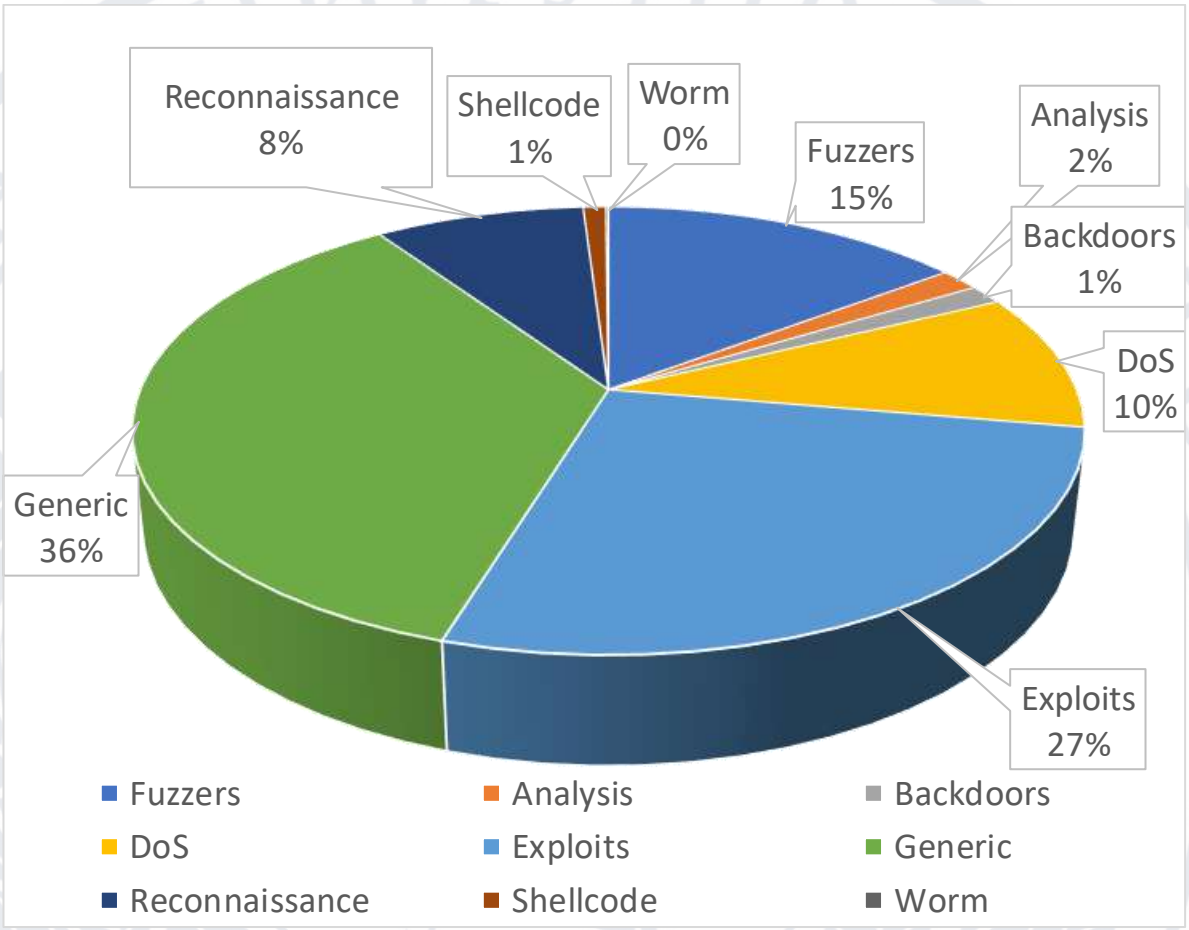

Рис. 3.6. Візуалізація категорій атак

Generic атаки - це широка категорія атак, які важко класифікувати в інші, більш специфічні типи. Вони часто використовують загальні вразливості або методи, які можуть бути застосовані до різних систем або програм, наприклад, атаки на протоколи (TCP/IP), атаки на переповнення буфера, SQL-ін'єкції.

Exploits атаки спрямовані на використання конкретних вразливостей в програмному забезпеченні або системах. Зловмисник використовує експлойт (спеціально створений код), щоб отримати несанкціонований доступ, виконати шкідливий код або порушити роботу системи, наприклад, використання вразливостей нульового дня (невідомих розробникам), використання помилок в конфігурації системи, атаки на веб-додатки.

Наступним етапом є розділення даних для навчання моделі (рис. 3.7) та побудова дерева рішень (рис. 3.8).

```
def split_data(df):
    X = df.drop(columns=['attack_cat', 'label'])
    y = df['label'].values
    return train_test_split( *arrays: X, y, test_size=0.3, random_state=11)
```

```
x_train shape (180371, 42)
x_test shape (77302, 42)
y_train shape (180371,)
y_test shape (77302,)
```

Рис. 3.7. Розділення даних для навчання моделі

```
def perform_grid_search(x_train, y_train):
    param_grid = {
        'criterion': ['gini', 'entropy'],
        'max_depth': [2, 4],
        'min_samples_split': [2, 4],
        'min_samples_leaf': [1, 2]
    }
    dt = DecisionTreeClassifier()
    grid_search = GridSearchCV(dt, param_grid, cv=5, scoring='recall')
    grid_search.fit(x_train, y_train)
    return grid_search.best_estimator_
```

Рис. 3.8. Побудова моделі дерева рішень

Для побудови моделі було використано наступні параметри:

- Criterion – визначає як алгоритм вирішує якість розбиття.
- Gini - відноситься до домішки Джині, яка вимірює ймовірність неправильної класифікації випадково вибраного елемента з набору даних, якщо він був випадково позначений відповідно до розподілу класів у підмножині
- Entropy - належить до приросту інформації, яка вимірює зменшення ентропії (невизначеності), досягнутої шляхом розбиття даних на основі певної ознаки.
- Max_Depth контролює максимальну глибину дерева. Цей параметр обмежує скільки рівнів рішень може прийняти дерево. Глибше дерево може

фіксувати більш складні зв'язки в даних, але може бути схильним до перенавчання (запам'ятовування навчальних даних занадто добре та не узагальнення нових даних). Тут код буде тестувати дерева з максимальною глибиною 2 та 4.

- Min_Samples_Split встановлює мінімальну кількість зразків, необхідних для розділення внутрішнього вузла. Це допомагає запобігти створенню деревом розбиттів з дуже малою кількістю зразків, що може призвести до перенавчання. Він буде тестувати розбиття вузлів з мінімум 2 та 4 зразками.
- Min_Samples_Leaf визначає мінімальну кількість зразків, необхідних для перебування в листовому вузлі (кінцевий вузол, який робить прогноз). Це також допомагає запобігти перенавчанню, гарантуючи, що листові вузли представляють достатню кількість зразків. Він буде тестувати наявність принаймні 1 або 2 зразків у кожному листовому вузлі.

```
Best recall score:  1.0
```

Як результат бачимо, що модель змогла досягнути recall 1.0, що означає 100% на тренувальних даних. І найкращі параметри для цього визначено наступні:

- Criterion: 'Gini' - це означає, що модель використовувала метрику домішки Джині, щоб визначити, як розділити дані в кожному вузлі дерева рішень.
- Max_Depth: 2 - це обмежує глибину нашого дерева рішень до 2 рівнів. Менша глибина може допомогти запобігти перенавчанню, коли модель занадто добре вивчає навчальні дані та не узагальнює нові дані.
- Min_Samples_Leaf: 1 - це вимагає, щоб у листовому вузлі (кінцевому вузлі, який робить прогноз) був принаймні 1 зразок.
- Min_Samples_Split: 2 - це вимагає, щоб у вузлі було принаймні 2 зразки, перш ніж його можна буде розділити далі.

Перевірка моделі на тестових даних (рис. 3.9) також показала 100% результат.

```
clf = perform_grid_search(x_train, y_train)
clf.fit(x_train, y_train)
y_pred = clf.predict(x_test)

recall = recall_score(y_test, y_pred)
print("Recall: ", recall)
```

```
Recall:  1.0
```

Рис. 3.9. Перевірка моделі на тестових даних

На наступному етапі відбувається виявлення кібератак, застосовуючи навчену модель до відфільтрованих тестових даних (рис. 3.10) та реалізація моделі у Random Forest (рис. 3.11).

```
x_test = x_test.reset_index(drop=True)

rules = "(sttl <= 61.00 & sinpkt<= 0.00) | (sttl >  61.00 )"
ind = x_test.query(rules).index

X_test_2 = x_test.loc[ind, :]
y_test_2 = y_test[ind]
print(X_test_2.shape)
print(y_test_2.shape)
print("filtered data" , (1- np.round(X_test_2.shape[0] / x_test.shape[0], decimals: 2))*100, "%")
```

```
(59425, 42)
(59425,)
filtered data 23.0 %
```

Рис. 3.10. Фільтрування тестових даних для потенційних атак

В результаті отримуємо 23%, які є потенційними атаками.

```
rf = RandomForestClassifier(random_state=11)
results['Random Forest Model'] = model_evaluation(rf, x_train, y_train, X_test_2, y_test_2)
```

```
Recall:  0.9571815389589395
Precision:  0.9647257693326001
Accuracy:  0.9351956247370635
```

Рис. 3.11. Реалізація моделі Random Forest на тестових даних

Реалізація моделі випадково лісу на тестових даних дозволила отримати наступні результати:

- Recall: 0.9571815389589395: модель правильно визначила приблизно 95,7% фактичних позитивних випадків.
- Precision: 0.9647257693326001: приблизно 96,5% випадків, передбачених як позитивні, були насправді позитивними.
- Accuracy: 0.9351956247370635: модель правильно класифікувала близько 93,5% випадків загалом.

В підсумку бачимо, що цей код оцінює модель випадкового лісу на тестовому наборі даних та повідомляє про її продуктивність за допомогою метрик повноти та точності. Він також візуалізує результати за допомогою матриці плутанини. Це забезпечує комплексну оцінку здатності моделі правильно класифікувати випадки.

3.2 Аналіз результатів тестування та оцінка ефективності.

Щоб покращити наше розуміння змінних, залучених до виявлення кібератак, нам потрібно проаналізувати мережеві дані. Кореляційні діаграми можуть бути корисними для візуалізації того, як різні змінні пов’язані одна з одною та з кібератаками. Крім того, моделі випадкового лісу можуть допомогти визначити важливість різних характеристик у прогнозуванні цільової змінної (кібератак). Можна порівняти рейтинги функцій із моделі випадкового лісу з результатами кореляційного аналізу, щоб краще зрозуміти ключові функції, на яких слід зосередитися для ефективного виявлення кібератак.

Дані, представлені на рис. 3.12, рис. 3.14 та рис. 3.16 дозволяють шляхом візуалізації матриці невідповідностей для різних моделей оцінити атаки за рівнем відкликання та помилок.

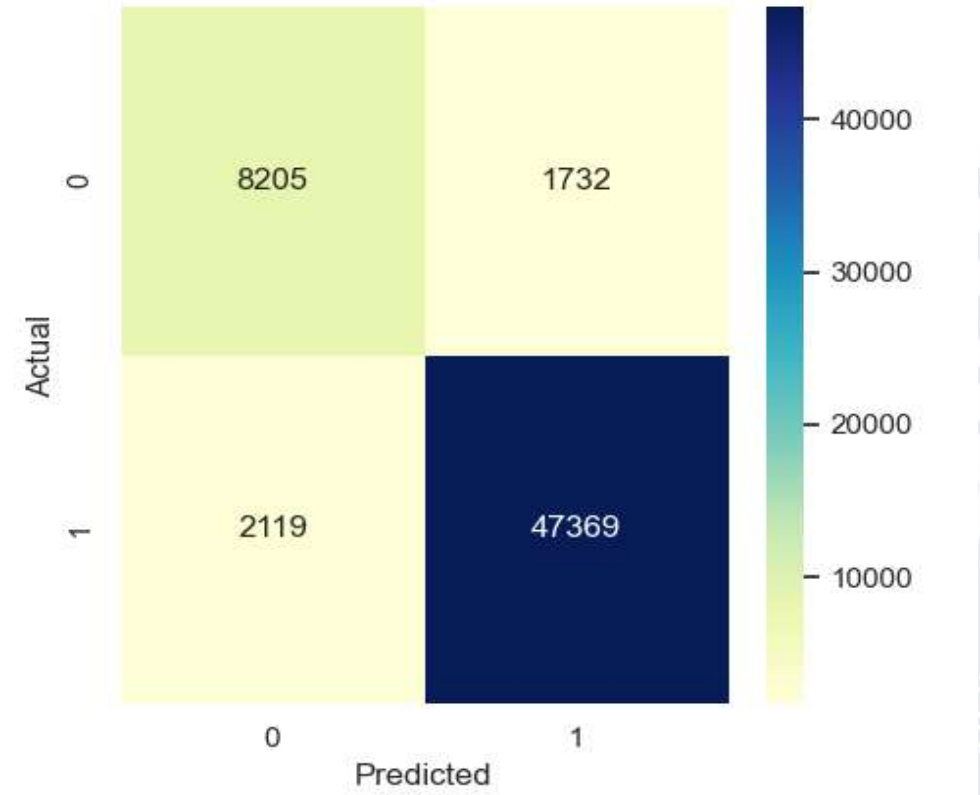

Рис. 3.12. Кількість атак та помилок відкликань для алгоритму Random Forest

Інтерпретація чисел:

- 47369 (TP): Модель правильно передбачила 47369 випадків як такі, що належать до класу 1 ("атака").
- 8205 (TN): Модель правильно передбачила 8205 випадків як такі, що належать до класу 0 ("нормальний").
- 1732 (FP): Модель неправильно передбачила 1732 випадки як клас 1 ("атака"), коли вони насправді були класом 0 ("нормальний").
- 2119 (FN): Модель неправильно передбачила 2119 випадки як клас 0 ("нормальний"), коли вони насправді були класом 1 ("атака").

Використання моделі XGBoost Claisfier (рис. 3.13).

```
xgbc = XGBClassifier()
results['XGBoost Classifier'] = model_evaluation(xgbc, x_train, y_train, X_test_2, y_test_2)
```

```
Recall:  0.9517458777885548
Precision:  0.9657774405872583
Accuracy:  0.931729070256626
```

Рис. 3.13. Реалізація моделі XGBoost Claisfier на тестових даних

- Recall: 0.9517458777885548: модель правильно визначила приблизно 95,1% фактичних позитивних випадків.
- Precision: 0.9657774405872583: приблизно 96,5% випадків, передбачених як позитивні, були насправді позитивними.
- Accuracy: 0.931729070256626: модель правильно класифікувала близько 93,1% випадків загалом.

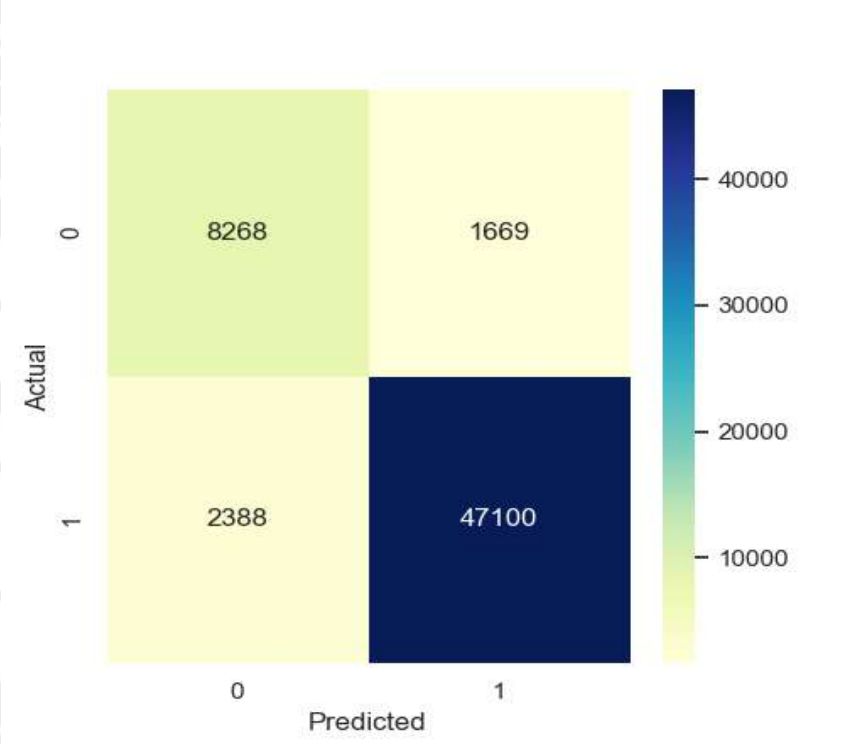

Рис. 3.14. Візуалізація матриці невідповідностей для XGBoost Claisfier

Інтерпретація чисел:

- 47100 (Справжні позитивні - True Positives): Модель правильно передбачила 47100 випадків як такі, що належать до класу 1 (ймовірно, "атака").
- 8268 (Справжні негативні - True Negatives): Модель правильно передбачила 8268 випадків як такі, що належать до класу 0 (ймовірно, "нормальний трафік").

- 1669 (Хибнопозитивні - False Positives): Модель помилково передбачила 1669 випадків як клас 1 ("атака"), коли вони насправді були класом 0 ("нормальний").
- 2388 (Хибнонегативні - False Negatives): Модель помилково передбачила 2388 випадків як клас 0 ("нормальний"), коли вони насправді були класом 1 ("атака").

Використовуєм модель для LGBMClassifier (рис. 3.15)

```
lgbc = LGBMClassifier()
results['Light GBM Classifier'] = model_evaluation(lgbc, x_train, y_train, X_test_2, y_test_2)
```

```
Recall:  0.9465122858066602
Precision:  0.964898547739211
Accuracy:  0.9267816575515355
```

Рис. 3.15. Реалізація моделі LGBMClassifier на тестових даних

- Recall: 0.9465122858066602: Модель правильно визначила приблизно 94,6% фактичних позитивних випадків.
- Precision: 0.964898547739211: Приблизно 96,5% випадків, передбачених як позитивні, були насправді позитивними.
- Accuracy: 0.9267816575515355: Модель правильно класифікувала близько 92,6% випадків загалом.

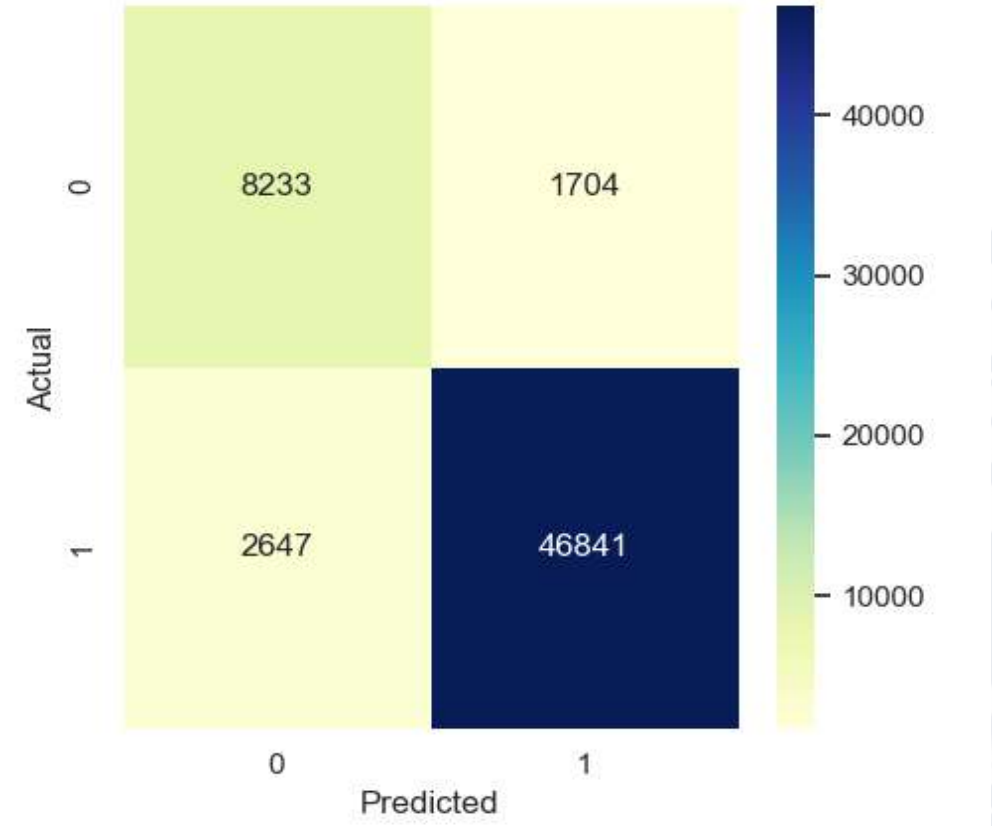

Рис. 3.16. Візуалізація матриці невідповідностей для LGBMClassifier

Інтерпретація чисел:

- 46841 (Справжні позитивні - True Positives - TP): Модель вірно класифікувала 46841 випадок як клас 1 (ймовірно, "атака").
- 8233 (Справжні негативні - True Negatives - TN): Модель вірно класифікувала 8233 випадки як клас 0 (ймовірно, "нормальний трафік").
- 1704 (Хибнопозитивні - False Positives - FP): Модель помилково класифікувала 1704 випадки як клас 1 ("атака"), хоча насправді вони належать до класу 0 ("нормальний").
- 2647 (Хибнонегативні - False Negatives - FN): Модель помилково класифікувала 2647 випадки як клас 0 ("нормальний"), хоча насправді вони належать до класу 1 ("атака").

На наступному етапі застосуємо ранговий тест суми Вілкоксона (рис. 3.17) - це непараметричний тест, який не вимагає нормального розподілу даних. Тест Вілкоксона обчислює Z-статистику та p-значення. Z-статистика є мірою того, наскільки велика варіація між трьома оцінками пригадування. P-значення є мірою ймовірності отримання такого ж екстремального результату, як спостережуваний, якщо нульова гіпотеза вірна. Нульова гіпотеза — це гіпотеза про те, що немає різниці у відкликанні трьох моделей. Якщо p-значення менше 0,05, тоді ми можемо відхилити нульову гіпотезу та зробити висновок, що існує статистично значуща різниця у відкликанні трьох моделей.

```
z_statistic, p_value = wilcoxon(comparison['Recall'])

print('Z-statistic:', z_statistic)
print('p-value:', p_value)

if p_value < 0.05:
    print('The difference in the recall of the three models is statistically significant.')
else:
    print('The difference in the recall of the three models is not statistically significant.')
```

```
Z-statistic: 0.0
p-value: 0.25
The difference in the recall of the three models is not statistically significant.
```

Рис. 3.17. Ранговий тест суми Вілкоксона

Будь-яка з трьох моделей за параметром Recall (правильне визначення фактичних позитивних випадків) може бути застосована для визначення важливих характеристик для прогнозування кібератак.

	Random Forest Model	XGBoost Classifier	Light GBM Classifier
Recall	0.957182	0.951281	0.946512

- Щоб покращити наше розуміння параметрів, задіяних у виявленні кібератак, нам потрібно проаналізувати мережеві дані. Діаграми кореляції можуть бути корисними для візуалізації того, як різні змінні пов'язані одна з одною та з кібератаками. Крім того, моделі випадкового лісу можуть допомогти визначити важливість різних ознак у прогнозуванні цільової змінної

(кібератаки). Ми можемо порівняти рейтинги ознак з моделі випадкового лісу з результатами кореляційного аналізу, щоб краще зрозуміти ключові ознаки, на яких слід зосередитись для ефективного виявлення кібератак.

```
def plot_correlation_matrix(df):
    corr_matrix = df.select_dtypes(include=[np.number]).corr()
    plt.figure(figsize=(12, 10))
    mask = np.triu(np.ones_like(corr_matrix, dtype=bool))
    sns.heatmap(corr_matrix, vmin=-1, vmax=1, cmap='BrBG', mask=mask)
    plt.show()
```

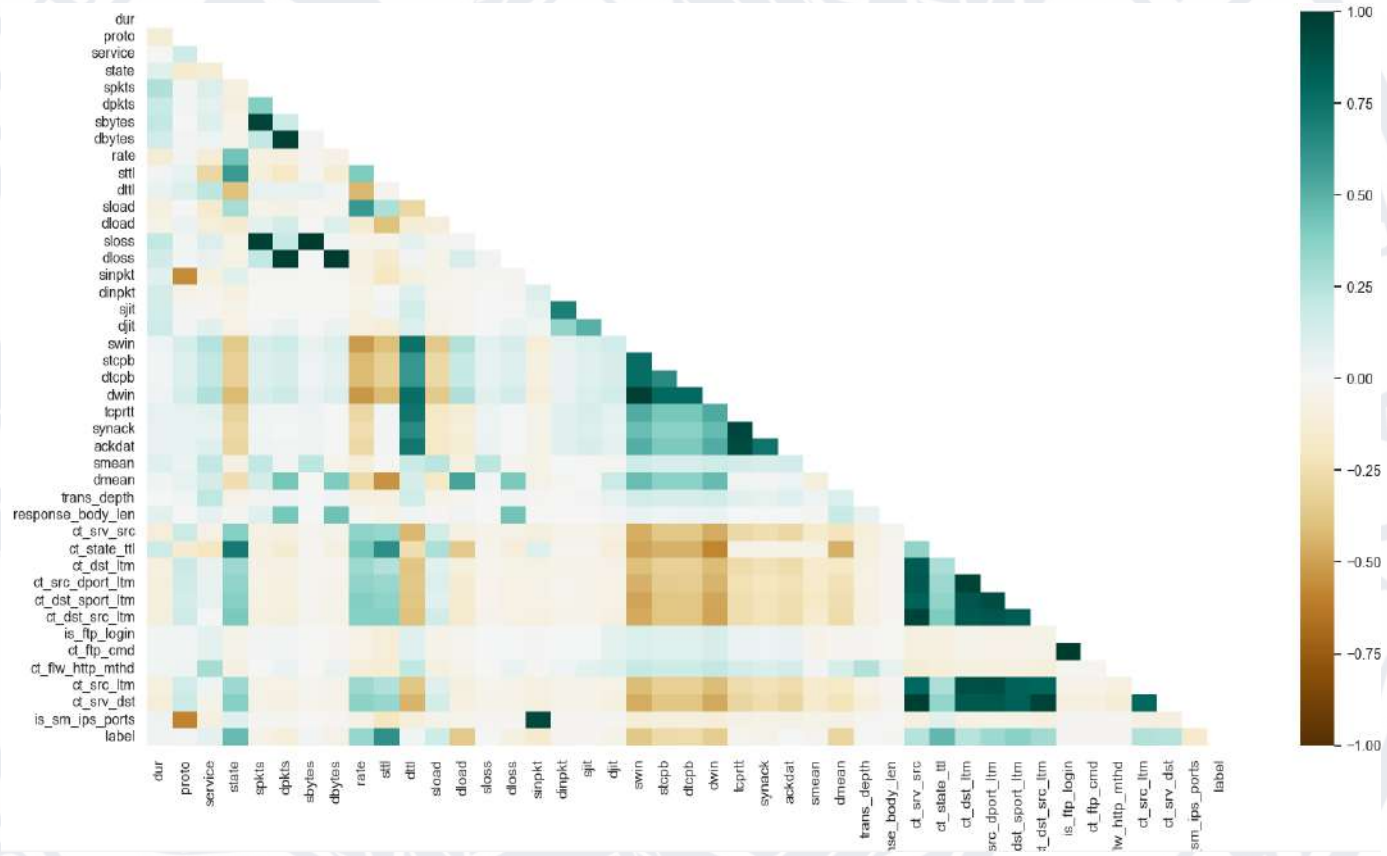

Рис. 3.18. Діаграма кореляції

```
plt.figure(figsize=(10, 10))
heatmap = sns.heatmap(corr_matrix_label, vmin=-1, vmax=1, annot=True, cmap='BrBG')
heatmap.set_title('Features Correlating with the Label', fontdict={'fontsize':18}, pad=16)
plt.show()
```

Рис. 3.29. Визначення кореляції ознак з міткою кібератака

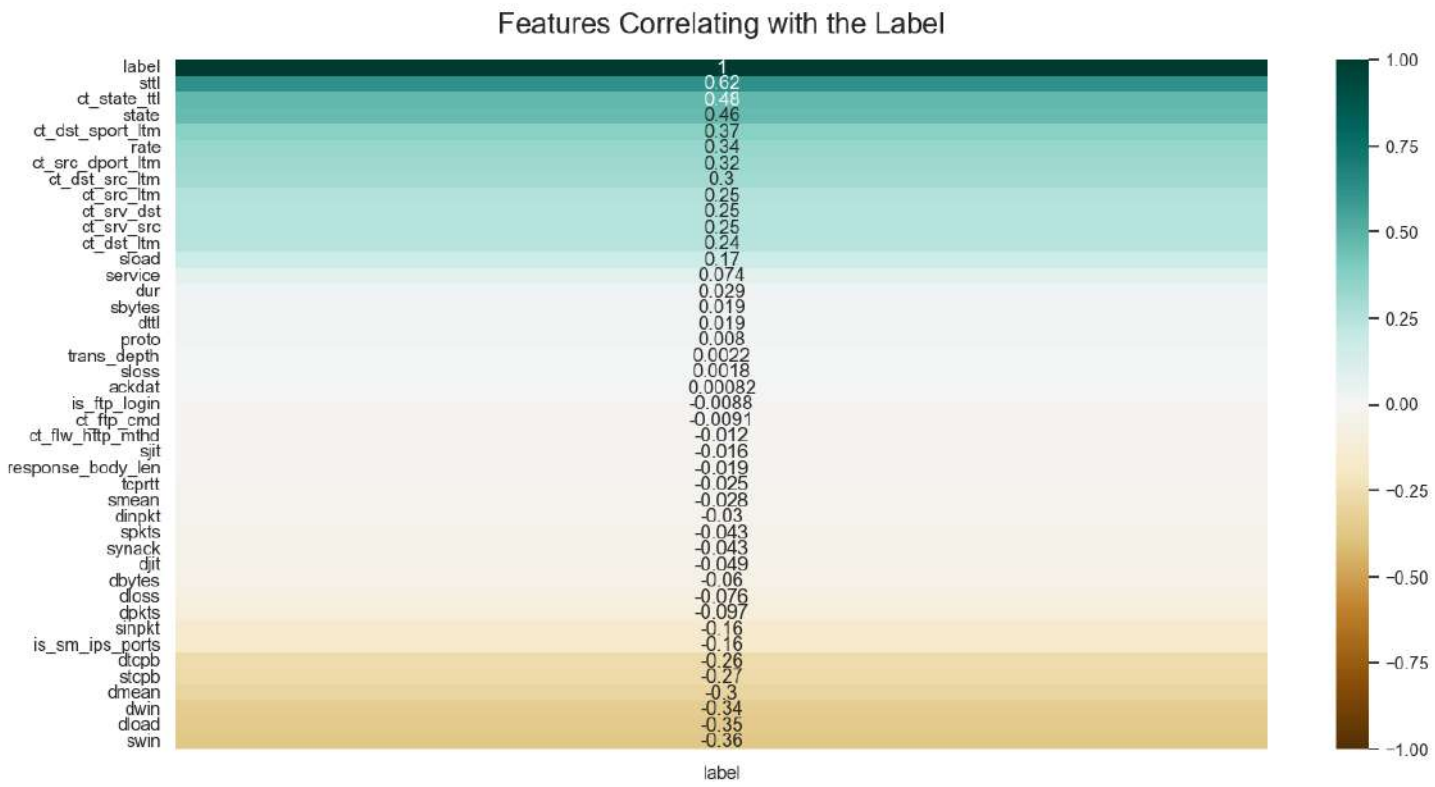

Рис. 3.20. Візуалізація коефіцієнтів кореляції між кожною числовою ознакою та змінною "label"

Графік (рис. 3.20) - це теплова карта, яка візуально представляє коефіцієнти кореляції між кожною числовою ознакою та змінною "label" (мітка).

Колір кожної комірки вказує на силу та напрямок кореляції:

- Темніший коричневий: Сильна негативна кореляція.
- Темніший синьо-зелений: Сильна позитивна кореляція.
- Світліші кольори/білий: Слабка або відсутність кореляції.

Ознаки відсортовані в порядку спадання їх кореляції з міткою, тому ознаки, що найсильніше корелюють, відображаються зверху.

Значення кореляції відображаються в кожній комірці, що дозволяє легко побачити точну силу кореляції.

Наступні змінні позитивно корелюють з кібератаками:

- sttl: Значення часу життя (Time-to-Live) від джерела до призначення.

- ct_state_ttl та state: Ці ознаки відображають різні стадії TCP-з'єднань і можуть бути пов'язані зі скануванням портів, SYN-флудом або DDoS-атаками.
- ct_dst_sport_ltm: Ця ознака вимірює кількість з'єднань з тієї ж IP-адреси джерела до того ж порту призначення за короткий проміжок часу.
- rate: Ця ознака може представляти різні типи швидкостей або частот трафіку.

Наступні змінні негативно корелюють з кібератаками:

- swin: Розмір вікна TCP джерела.
- dload: Швидкість передачі даних від призначення в бітах за секунду.

Шляхом сортування функцій випадкового лісу ми обрали лише 10 найпопулярніших функцій (рис. 3.21) з метою пошуку їх асоціацій з типом кібератаки.

	Name	Importance
0	sttl	0.127183
1	ct_state_ttl	0.098745
2	rate	0.056321
3	dload	0.049838
4	sload	0.045773
5	sbytes	0.043195
6	ct_srv_dst	0.040537
7	smean	0.039754
8	ct_dst_src_ltm	0.038934
9	tcprtt	0.033016

Рис. 3.21. Рейтинг функцій випадкового лісу

На рис. 3.22 представлено візуалізацію випадкового лісу з категорією атаки як мітками прогнозування.

```
rf = RandomForestClassifier(random_state=11, min_samples_leaf=1, min_samples_split=5, n_estimators=100)

rf.fit(x_train, y_train)

y_pred = rf.predict(x_test)

acc = accuracy_score(y_test, y_pred)
print("Accuracy: ", acc)

cross = pd.crosstab(y_test, y_pred)
plt.figure(figsize=(10, 10))
sns.heatmap(cross, annot=True, fmt='d', cmap="YlGnBu")
plt.show()
```

```
Accuracy:  0.8243253732115599
```

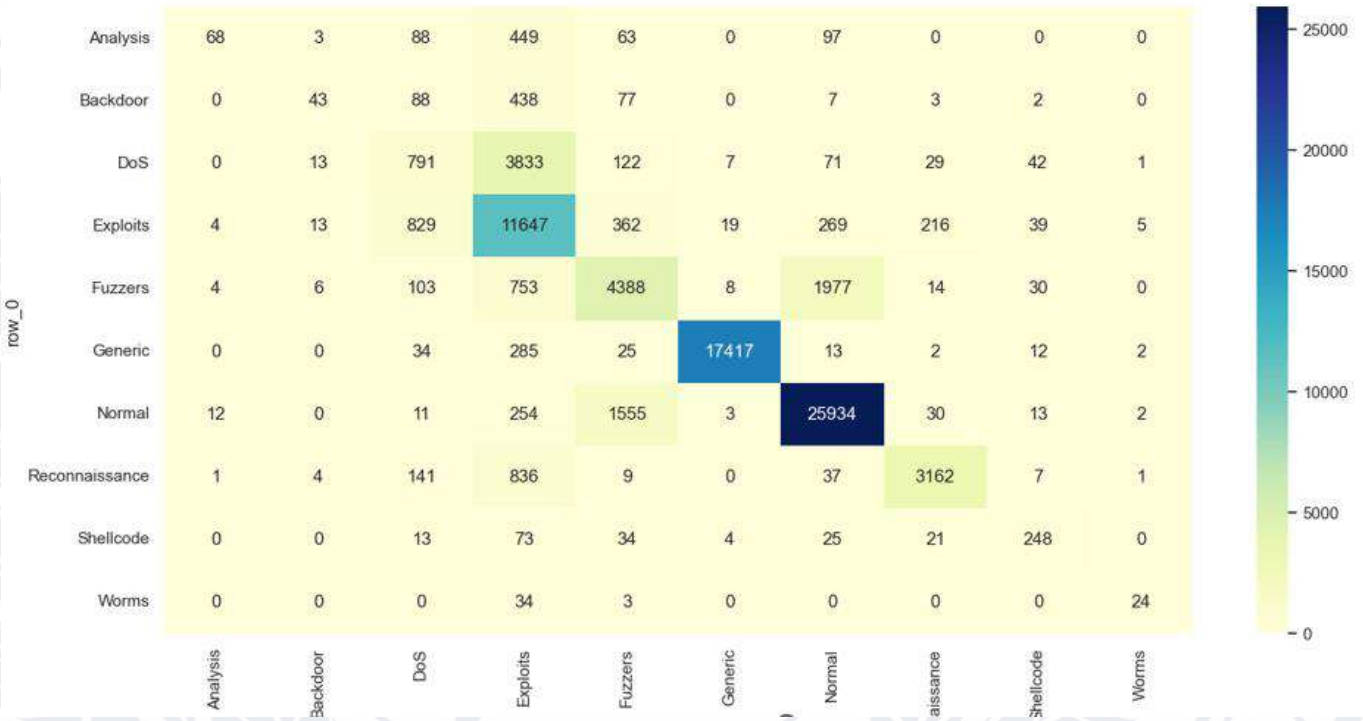

Рис. 3.22. Випадковий ліс із категорією атаки як мітками передбачення

Розглянемо детальніше матрицю невідповідностей:

Загальна структура:

- Рядки представляють справжні категорії атак (Analysis, Backdoor, DoS, Exploits, Fuzzers, Generic, Normal, Reconnaissance, Shellcode, Worms).
- Стовпці представляють категорії, передбачені моделлю.
- Комірки показують кількість випадків, які потрапили в певну комбінацію "справжня категорія - передбачена категорія".

Діагональ:

- Ідеальний сценарій - коли всі числа зосереджені на діагоналі (від верхнього лівого кута до нижнього правого). Це означало б, що модель ідеально класифікує всі атаки.
- У нашому випадку, є великі числа на діагоналі для деяких категорій, наприклад, "Normal" (25934), "Exploits" (11647), "Generic" (11417), "Fuzzers" (4088), "Reconnaissance" (3162). Це означає, що модель добре розпізнає ці типи атак.

Помилки:

- Хибнопозитивні: коли модель помилково класифікує нормальний трафік або інший тип атаки як певну категорію. Наприклад, 791 випадок "DoS" були класифіковані як "Exploits".
- Хибнонегативні: коли модель не розпізнає атаку певного типу. Наприклад, 629 випадків "Exploits" були класифіковані як "DoS".

Проблемні місця:

- Модель часто плутає "DoS" та "Exploits".
- Категорії "Backdoor", "Shellcode" та "Worms" мають дуже мало прикладів і погано розпізнаються.

Звіт про класифікацію атак, представлени на рис. 3.23 показує, як добре модель випадкового лісу розпізнає різні типи кібератак.

Давайте проаналізуємо ці метрики:

Precision (точність):

- Показує, яка частка прогнозів для певної категорії атаки була вірною.
- Наприклад, precision = 0.76 для "Analysis" означає, що 76% прогнозів, зроблених моделлю для цієї категорії, були правильними.
- Висока точність важлива, коли важливо мінімізувати хибнопозитивні результати (коли модель помилково визначає атаку).

	precision	recall	f1-score	support
Analysis	0.76	0.09	0.16	768
Backdoor	0.52	0.07	0.12	658
DoS	0.38	0.16	0.23	4909
Exploits	0.63	0.87	0.73	13403
Fuzzers	0.66	0.60	0.63	7283
Generic	1.00	0.98	0.99	17790
Normal	0.91	0.93	0.92	27814
Reconnaissance	0.91	0.75	0.82	4198
Shellcode	0.63	0.59	0.61	418
Worms	0.69	0.39	0.50	61
accuracy			0.82	77302
macro avg	0.71	0.54	0.57	77302
weighted avg	0.82	0.82	0.81	77302

Рис. 3.23. Звіт про класфікацію атак

Recall (повнота):

- Показує, яка частка справжніх атак певної категорії була правильно розпізнана моделлю.
- Наприклад, recall = 0.09 для "Analysis" означає, що модель змогла розпізнати лише 9% атак типу "Analysis".
- Висока повнота важлива, коли важливо мінімізувати хибнонегативні результати (коли модель пропускає справжні атаки).

F1-score (F1-міра):

- Це середнє гармонійне між точністю та повнотою. Вона враховує обидві метрики і дає більш збалансовану оцінку.
- Корисна, коли потрібно знайти баланс між точністю та повнотою.

Support (підтримка):

- Це кількість випадків певної категорії атаки в тестовому наборі даних.

Аналіз за категоріями:

- "Generic" та "Normal": Модель дуже добре розпізнає ці категорії, маючи високу точність, повноту та F1-міру.
- "Exploits" та "Reconnaissance": Модель також досить добре працює з цими категоріями, хоча повнота для "Reconnaissance" трохи нижча.
- "Fuzzers": Модель показує середні результати для цієї категорії.
- "Analysis", "Backdoor", "DoS", "Shellcode", "Worms": Модель має проблеми з цими категоріями, особливо з "Analysis" та "Backdoor", де повнота дуже низька. Це може бути пов'язано з недостатньою кількістю даних для навчання або зі складністю розпізнавання цих типів атак.

Загальні метрики:

- Accuracy (точність) = 0.82: Загалом, модель правильно класифікує 82% випадків.
- Macro avg та weighted avg: Ці метрики показують середню продуктивність за всіма категоріями. Weighted avg враховує кількість прикладів для кожної категорії, тому вона більше відображає продуктивність на реальних даних.

Отже, модель випадкового лісу найкраще підходить для виявлення кібератак. Модель дерева рішень була використана для визначення правил виявлення кібератак. Завдяки цьому аналізу були визначені найважливіші ознаки, які відіграють головну роль у виявленні кібератак.

3.3. Розробка рекомендацій щодо захисту інформаційного середовища "Smart City".

Взаємопов'язаний характер інфраструктури "розумного міста" означає, що вразливість в одній системі може мати каскадні наслідки для багатьох служб, посилюючи потенційний вплив кіберінцидентів. Необхідна значна кількість ресурсів для збору трафіку з усіх девайсів в мережі розумного міста. Оскільки саме постійний моніторинг та аналіз трафіку з усіх девайсів дозволить новим рішенням в сфері IDS/IPS виявляти загрози краще, враховуючи сучасний

напрямок розвитку IDS/IPS, а саме інтегрування технологій Deep learning (DL) та Machine learning (ML). Це рішення має як переваги так і недоліки, до переваг можна віднести те що в потенціалі такі системи будуть автоматично виявляти та правильно реагувати на більшість сучасних загроз, але кількість навчального матеріалу для подібних систем обмежена. Розумне місто це дуже велика та комплексна система яка потребує індивідуального підходу, враховуючи це та вище згаданий недолік це робить розумне місто вразливим перед новими атаками та на початку свого створення до моменту впровадження систем реагування на інциденти.

Провівши аналіз ризиків інформаційного середовища розумного міста, сучасних методів забезпечення захисту інформаційного середовища настільки комплексних та складних мереж, дозволяє нам зробити висновок, що існує кілька ключових технологічних векторів, які можуть значно підвищити стійкість цих цифрових екосистем до кіберзагроз.

Розробка гармонізованої системи кібербезпеки дозволить операторам "розумних міст" впроваджувати загальні рекомендації. Такі рамки повинні підтримуватися муніципалітетами, які будуть виступати в ролі координатора між зацікавленими сторонами (операторами, виробниками тощо). Гармонізація важлива для забезпечення послідовного рівня безпеки для будь-якої служби міста. Ця структура може інтегрувати відповідні стандарти безпеки (стандарти інтероперабельності, які гарантують, що різне обладнання може безпечно працювати разом; регулярні перевірки безпеки, що передбачають регулярні перевірки обладнання та способів його використання з метою швидкого знаходження та вирішення проблеми; використання спеціальних способів безпечного введення в експлуатацію пристроїв та захисту обладнання, розроблених з урахуванням вимог безпеки, що може забезпечити додатковий захист.

Гармонізована система кібербезпеки за підтримки муніципалітетів також дозволить інтегрувати існуючі підходи до управління ризиками, враховуючи при цьому особливості міста (наприклад, соціальні аспекти, архітектуру ІКТ тощо).

Організації, відповідальні за впровадження технології "розумного міста", повинні визначити, чи потребуватимуть пристрої та обладнання Інтернету речей, які забезпечать "розумну" функціональність, підтримки сторонніх або зовнішніх сервісів. Ці організації повинні визначити, як пристрої зберігають та обмінюються даними і як пристрої захищають дані в стані спокою, під час передачі та використання. Організації повинні вести реєстр ризиків, який визначає залежність від підтримки хмарних обчислень, зовнішніх компонентів та інших подібних залежностей, як власну, так і від постачальників

Організації повинні ретельно переглядати угоди про надання хмарних послуг, включаючи положення про безпеку даних та моделі розподілу відповідальності.

Розширена поверхня атаки є найкритичнішим ризиком для інформаційного середовища розумного міста. Більшість відомих атак на розумні міста були успішними саме через відсутність грамотної сегментації мережевих девайсів. Розміщення пристроїв Інтернету речей в окремій частині мережі може підвищити безпеку як самих девайсів, так і решти мережі. Розділивши мережу на менші, керовані сегменти, міста можуть ізолювати критичні системи і дані від менш чутливих, обмежуючи потенційне поширення кібератаки. Заходи контролю доступу повинні бути суворими, гарантуючи, що пристрої та користувачі матимуть доступ лише до тих сегментів мережі та даних, які необхідні для їхньої роботи. Розгортання передових рішень безпеки, які використовують штучний інтелект і машинне навчання, може допомогти виявити незвичайні патерни і потенційні інциденти безпеки до того, як вони розростуться. У поєднанні з надійною програмою управління вразливостями, яка включає регулярне сканування, оцінку та своєчасне застосування виправлень і оновлень, міста можуть значно зменшити вразливість до кіберзагроз. Підхід на основі мікросигментації дозволяє розумним містам впроваджувати управління доступом на основі політик, які динамічно коригуються в залежності від контексту запиту на доступ, значно зменшуючи поверхню атаки і обмежуючи потенційний вплив порушення.

Використання спеціального програмного забезпечення для захисту пристроїв IoT, створеного розробниками.

Також підвищити рівень безпеки може використання багатофакторної автентифікації. Шифрування інформації, яка надсилається пристроями IoT також підвищує їх рівень захищеності.

Регулярне тестування на предмет злому пристроїв IoT перед потраплянням до користувачів з метою виявлення проблеми на ранніх стадіях.

Встановлення правил на рівні регулювання урядом, яких повинні дотримуватися виробники пристроїв, щоб гарантувати, що пристрої Інтернету речей відповідають певним стандартам безпеки.

До краших практик на сьогодні можна віднести також практику управління оновленнями програмного забезпечення. Наразі – це складне завдання для IoT, але компанії, які дбають про свою репутацію, працюють над тим, щоб надсилання оновлень на пристрої було простішим і безпечнішим.

Використання надійних постачальників – використання пристроїв від компаній, які дотримуються належних практик безпеки, може допомогти уникнути проблем.

Прогнози

Збільшення кількості пристроїв – у майбутньому в будинках і на підприємствах буде ще більше пристроїв Інтернету речей. Це означає, що забезпечення їхньої безпеки стане ще більш важливим.

Розумніші атаки – люди, які хочуть недобросовісно скористатися цими пристроями, будуть робити це все краще і краще. Безпека також повинна розвиватися.

Нові правила та стандарти – оскільки все більше людей користуються цими пристроями, уряди розроблятимуть нові правила, щоб забезпечити їхню безпеку.

Висновок до розділу ІІІ

Захист Інтернету речей – непросте завдання через використання безлічі різних типів пристроїв. Використання безпечних, добре захищених мереж, до яких підключені пристрої Інтернету речей, має дуже важливе значення.

Моделі виявлення мережевих атак на IoT системи є потужним інструментом у боротьбі з кіберзагрозами та мають потенціал для подальшого вдосконалення та застосування у різних сферах.

Вибір моделі виявлення аномалій здійснено з урахуванням конкретних потреб та характеристик досліджуваної системи. Модель дерева рішень використовувалася для визначення правил виявлення кібератак. У результаті цього аналізу було визначено найважливіші функції, які відіграють важливу роль у виявленні кібератак та встановлено, що модель випадкового лісу найкраще підходить для виявлення кібератак, добре працює як із залежною функцією, так і з міткою та категорією атаки.

Модель виявлення аномалій продемонструвала надійність та ефективність виявлення відхилень у поведінці IoT-пристроїв.

ВИСНОВКИ

В роботі проаналізовано складові інформаційних та комунікаційних технологій «Smart City», зокрема, IT_{IoT} (Інтернет речей), IT_{BD} (великі дані) та IT_{OD} (відкриті дані), IT_{BL} (блокчейн), IT_{CL} (хмарні технології), IT_{AI} штучний інтелект) та IT_{ML} (машинне навчання), що дало змогу сформувати структурну схему інформаційних потоків та узагальнену модель життєвого циклу даних «Smart City».

Розглянуто види моделей та методи виявлення кібератак та обґрунтовано вибір моделі виявлення аномалій на основі методів машинного навчання. Дана модель актуальна універсальністю, адаптивністю, здатністю знижувати false positive спрацювання, виявляти складні атаки та перспективністю розвитку. Ця модель є потужним інструментом у боротьбі з кіберзагрозами та має потенціал для подальшого вдосконалення та застосування у різних сферах.

Вибір моделі виявлення аномалій здійснено з урахуванням конкретних потреб та характеристик досліджуваної системи. У відповідності з вимогами щодо надійності та ефективності виявлення відхилень у поведінці IoT-пристроїв.

Реалізація моделі вторгнень була виконана за допомогою наступних алгоритмів класифікації: Random Forest, XGBoost Claisfier та LGBMClassifier. Отримані результати були проаналізовані, керуючись якісними параметрами алгоритмів. Найкращим алгоритмом для вирішення поставлених задач виявився Random Forest Classifier, який продемонстрував наступні значення показників: Recal (повнота) - точність розпізнавання атак становить 95,7%; Precision - приблизно 96,5% випадків, передбачених як позитивні, були насправді позитивними; Accuracy - загалом близько 93,5% випадки модель класифікувала правильно. Модель періодично перенавчається на нових даних, щоб враховувати зміни в нормальній поведінці. У разі потреби архітектура моделі може бути змінена для врахування нових типів даних або загроз. Запропонована модель може бути хорошим доповненням до стандартних IDS, що може покращити процес виявлення мережевих атак на IoT системи «Smart City».

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. D. Kumar. The Conception and Discourse of Smart City. *Smart Cities and Regional Development Journal*, vol. 7, no. 1, pp. 71-84, 2023.

2. K.C. Laudon and J.P. Laudon. Management Information Systems: Managing the Digital Firm. *Pearson Prentice Hall*, vol. 16, 2020.

3. О. Палка, Л. Дмитроца. Аналіз мікросервісної архітектури розумного міста. *Вісник Хмельницького національного університету*. Серія: «Технічні науки». №6(329). С. 142-149. doi:10.31891/2307-5732-2023-329- 6-142-149. 2023.

4. A.S. Syed, D. Sierra-Sosa, A. Kumar, and A. Elmaghraby. IoT in Smart Cities: A Survey of Technologies, Practices and Challenges. *Smart Cities*, vol. 4, pp. 429-475, 2021.

5. N.F.M. Shari and A. Malip. State-of-the-art solutions of blockchain technology for data dissemination in smart cities: A comprehensive review. *Computer Communications*, vol. 189, pp. 120-147, 2022.

6. Володимир Бачинський Application Security Engineer, Sigma Software. URL: https://itcluster.lviv.ua/itid/vyklyky-bezpeky-iot-novi-rishennya-ta-krashhi-praktyky/

7. Чайка О. Російські хакери координують дії з військовими та посилюють атаки напередодні зими. Як Україна протистоїть кібератакам на енергосистему. 2023. URL: https://forbes.ua/company/rosiyski-khakeri-koordinuyut-dii-z-viyskovimi-ta-posilyuyut-ataki-naperedodni-zimi-yak-ukraina-protistoit-kiberatakam-na-energosistemu-08112023-17242

8. Тартачний О. Російсько-українська кібервійна: що та чому стає мішенню хакерів. URL: https://thepage.ua/ua/it/yak-rosijsko-ukrayinska-vijna-prohodit-u-kiberprostori

9. Злам федеральної податкової служби рф — деталі чергової кіберспецоперації ГУР. 2023. URL: https://gur.gov.ua/content/zlam-federalnoi-podatkovoi-sluzhby-rf-detali-cherhovoi-kiberspetsoperatsii-hur.html

10. Дудикевич В.Б., Микитин Г.В., Бортнік Л.Л., Стосик Т.Р. Методологія безпеки кіберфізичних систем та інтернету речей в

інтелектуалізації об’єктів інфраструктури. *Computer Systems And Networks.* Vol. 6, No. 1, 2024. С. 44-53.

11. Mehta S., Bhushan B., Kumar R. Machine learning approaches for smart city applications: Emergence, challenges and opportunities. *Recent Advances in Internet of Things and Machine Learning: RealWorld Applications*, pp. 147- 163, 2022.

12. Pratama S., Suswanta I. Artificial Intelligence in Realizing Smart City through City Operation Center. *Advances in Economics, Business and Management Research*. vol. 209, pp. 53-60, 2021.

13. Herath H.M.K.K.M.B., Mittal M. Adoption of artificial intelligence in smart cities: A comprehensive review. *International Journal of Information Management Data Insights*, vol. 2, 100076, 2022.

14. Dou X., Chen W., Zhu L., Bai Y., Li Y., Wu X. Machine Learning for Smart Cities: A Comprehensive Review of Applications and Opportunities. *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 9, pp. 999-1016, 2023.

15. Zhou X., Liu H., Pourpanah F., Zeng T., Wang X. A survey on epistemic (model) uncertainty in supervised learning: Recent advances and applications. *Neurocomputing*, vol. 489, pp. 449-465, 2022.

16. Kosarirad H., Ghasempour Nejati M., Saffari A., Khishe M. Mohammadi M. Feature selection and training multilayer perceptron neural networks using grasshopper optimization algorithm for design optimal classifier of big data sonar. *Journal of Sensors*, vol. 22, 2022.

17. Бобровнікова К., Гурман І., Попов Ю., Бойчук Я., Качур В. Виявлення кібератак в інфраструктурі інтернету речей на основі машинного навчання. *Вісник Хмельницького національного університету*, №3, 2023 (321). С. 251-257.

18. Бобровнікова, К., Капустян, М., & Денисюк, Д. Дослідження методів виявлення кібератак в інфраструктурі інтернету речей на основі машинного

навчання. *Computer Systems and Information Technologies*, (3), 2022. 110–115. URL: https://doi.org/10.31891/CSIT-2021-5-15

19. Ravi, N., & Shalinie, S. M. Learning-driven detection and mitigation of DDoS attack in IoT via SDN-cloud architecture. *IEEE Internet of Things Journal*, 2020, 7(4), pp. 3559-3570.

20. Otoum Y., Liu D., Nayak, A. DL-IDS: a deep learning-based intrusion detection framework for securing IoT. *Transactions on Emerging Telecommunications Technologies*, 2019, p. 3803.

21. Набори даних UNSW-NB15. URL: https://research.unsw.edu.au/projects/unsw-nb15-dataset

22. Shafiq M., Tian Z., Bashir,A. K., Du X., Guizani M. Corrauc: a malicious bot-iot traffic detection method in iot network using machine learning techniques. *IEEE Internet of Things Journal*, 2020, pp. 356-359

23. Aysa M. H., Ibrahim A. A., Mohammed A. H. IoT ddos attack detection using machine learning. *Studies and Innovative Technologies* (ISMSIT), IEEE, 2020, pp. 1-7.

24. Недоступ Д. М., Солом'яний М. В. Аналіз підходів забезпечення відмовостійкості архітектур Extended Cloud. *Проблеми електромагнітної сумісності перспективних безпроводових мереж зв'язку (ЕМС-2022) : мат. Мжн. наук-техн. конф.* Харків, ХНУРЕ, 2022. С. 39-40. URL: https://openarchive.nure.ua/handle/document/21185

25. Marinova-Boncheva V. Applying a Data Mining Method for Intrusion Detection. *International Conference on Computer Systems and Technologies CompSysTech*. 2007, pp. 48-59.

26. Що таке IPS/IDS і де застосовується? 2022. URL: https://www.hostzealot.com.ua/blog/about-solutions/shho-take-ipsids-i-de-zastosovujetsya

27. Zhang R., Qian D., Ba C., Wu W. and Guo X. Multi-agent based intrusion detection architecture. *In Proceedings of 2001 IEEE International Conference on Computer Networks and Mobile Computing*, pp. 494–501, oct. 2001.

28. Vigna G., Valeur F., Kemmerer. Richard A. Designing and implementing a family of intrusion detection systems. *In Proceedings of the 9th European software engineering conference held jointly with 10th ACM SIGSOFT international symposium on Foundations of software engineering*, Helsinki, Finland, 2003. Source: ACM Portal. pp. 88–97.

29. Ajit Abraham, Ravi Jain, Johnson Thomas, Sang Yang Han. D-SCIDS: Distributed softcomputing intrusion detection system. *Journal of Network and Computer Applications*, Elsevier, 2005.

30. Chou T. S., Yen K. K. and Luo J. Network Intrusion Detection Design Using Feature Selection of Soft Computing Paradigms. *International Journal of Computaional Intelligence*. Vol. 4, no. 3, 2007.

31. Intrusion Detection Systems. 2002. URL: http://www.ciscopress.com/articles/ article.asp?p=25334.

32. NSL-KDD dataset. URL: http://www. unb.ca/cic/datasets/nsl.html.

33. Литвин, А. Тенденції розвитку світового ринку інформаційних технологій. *Теоретичні і практичні аспекти економіки та інтелектуальної власності: збірник наукових праць*. ПДТУ. Вип. 2. 2020. С. 132-137.

34. Махнов Я.Г, Половенко Л.П. , Загоруйко Л.В. Сучасні загрози та кібератаки на важливі інформаційні структури. *Відновлення України у повоєнні часи: виклики, стратегічні пріоритети, ресурсне забезпечення, потенціал майбутнього розвитку: Міжнародна науковопрактична конференція, присвячена 10-річчю переміщення Донецького національного університету імені Василя Стуса до м. Вінниці (10–11 жовтня 2024 р.). Вінниця: ДонНУ імені Василя Стуса*, 2024. С. 36-38. URL: https://jrupp.donnu.edu.ua/issue/view/547

35. Метрики в задачах машинного навчання. 2017. URL: https://habrahabr.ru/ company/ods/blog/328372/.

36. McAfee Labs Threats Report. URL: https://www.mcafee.com/enterprise/en-us/lp/threats-reports/oct-2021.htm

37. OWASP. Intrusion Detection. 2002. URL: https://www.owasp.org/index.php/ Intrusion_Detection.

38. Палка О. Аналіз інтегрованої архітектури розумного міста з блокчейном та IoT. *Науковий вісник НЛТУ України*, 33(6), 2023. С. 94-99. doi:10.36930/40330612.

39. Пороло Є. Застосування концепції Data Bank в мережі хмарного IoT. *Перспективи розвитку інформаційно-телекомунікаційних технологій та систем*. м. Київ, 2020. С. 368.

40. Gifty Jeya P., Ravichandran M., Ravichandran C. S. Efficient Classifier for R2L and U2R Attacks. 2012. URL: https://pdfs.semanticscholar.org/bbf5/bfe8c4fc238405df91f54849eabb2aadf1cb.pdf.

41. Verma A., Ranga, V. Machine learning based intrusion detection systems for IoT applications. *Wireless Personal Communications*, 2020, 111(4), pp. 2287-2310.

42. Tanzila Saba, Amjad Rehman Khan, Tariq Sadad , Seng-phil Hong. Securing the IoT System of Smart City against Cyber Threats Using Deep Learning. *Discrete dynamics in nature and society*. 2022. ULR: https://downloads.hindawi.com/journals/ddns/2022/1241122.pdf

43. Alam, T. Cloud-based IoT applications and their roles in smart cities. *Smart Cities* 2021, 4, 1196–1219.

44. Kumar S. Rathore R.S., Mahmud M.; Kaiwartya O., Lloret, J. BEST—Blockchain-Enabled Secure and Trusted Public Emergency Services for Smart Cities Environment. *Sensors* 2022, 22, p. 5733.

45. Bagui S., Wang X., Bagui S. Machine Learning Based Intrusion Detection for IoT Botnet. *International Journal of Machine Learning and Computing*, 11(6), 2021.

46. Wu C. K. Internet of Things Security. Architectures and Security Measures. *Advances in Computer Science and Technology*. *Springer, Singapore*, 2021. p.245 DOI: 10.1007/978-981-16-1372-2

47. Yaacoub Jean-Paul A., Salman Ola, Noura Hassan N., Kaaniche Nesrine, Chehab Ali, Malli Mohamad. Cyber-physical systems security: Limitations, issues and

future trends. *Microprocessors and Microsystems*, Volume 77, 2020, DOI: 10.1016/j.micpro.2020.103201

48. Koomen E., van Bemmel M.S., van Huijstee J., André B.P.J., Ferdinand P.A., van Rijn F.J.A. An integrated global model of local urban development and population change. *Computers, Environment and Urban Systems*, vol. 100, 2023.

49. Rafiq I., Mahmood A., Razzaq S., Jafri S.H.M., Aziz I. IoT applications and challenges in smart cities and services. *The Journal of Engineering*, vol. 2023, no. 4, pp. 1-25, 2023.

50. Fiore M., Mongiello M. Blockchain for smart cities improvement: an architecture proposal. *Software and Modeling* (ICCSM 2023), pp. 1-5, 2023.

ДОДАТОК А