НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

До захисту допущено
Завідувач кафедри

_____ **Дмитро ЛАНДЕ**
(підпис)

«_____»_____2024 р.

Дипломна робота

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Системи, технології та математичні методи кібербезпеки»
спеціальності 125 «Кібербезпека»

на тему: Виявлення веб загроз за допомогою машинного навчання
Виконав (-ла): здобувач вищої освіти **IV** курсу, групи ФБ-06
(шифр групи)

Ісаченко Федір Сергійович
(прізвище, ім'я, по батькові) (підпис)

Керівник професор, завідувач кафедри ІБ,

Ланде Дмитро Володимирович
(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові) (підпис)

Рецензент
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище, ім'я, по батькові)
(підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Здобувач вищої освіти _____
(підпис)

Київ – 2024 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

Рівень вищої освіти – перший (бакалаврський)
Спеціальність – 125 «Кібербезпека»
Освітньо-професійна програма «Системи, технології та математичні методи кібербезпеки»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Дмитро ЛАНДЕ
(підпис)

«___»_____2024 р.

ЗАВДАННЯ
на дипломну роботу здобувачу вищої освіти

(прізвище, ім’я, по батькові)

1. Тема роботи Виявлення веб загроз за допомогою машинного навчання,
Керівник роботи професор, завідувач кафедри ІБ, Ланде Дмитро Володимирович,

затверджені наказом по університету від «31» травня 2024 р. №2024

2. Термін подання здобувачем вищої освіти роботи 15 червня 2024 р.

3. Вихідні дані до роботи: наукові статті та книги за методів виявлення кіберзагроз за допомогою машинного навчання

4. Зміст роботи: огляд методів виявлення веб загроз за допомогою машинного навчання, огляд датасету та його попередня підготовка, тренування моделей та вибір найкращої з них, огляд найвпливовіших властивостей запитів

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо):

Презентація до захисту дипломної роботи.

6. Дата видачі завдання 21 вересня 2023р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів дипломної роботи	Примітка
1	Отримання завдання та узгодження теми роботи	21.09.2023	Виконано
2	Огляд наукової літератури по темі	01.03.2024 – 01.04.2024	Виконано
3	Вибір датасету та його підготовка	02.04.2024 – 18.04.2024	Виконано
4	Вибір моделей та їх тренування	19.04.2024 – 25.05.2024	Виконано
5	Порівняння результатів моделей	26.05.2024 – 31.05.2024	Виконано
6	Підготовка та передзахист ДР	01.06.2024 – 10.06.2024	Виконано
7	Підготовка до захисту ДР	11.06.2024 – 17.06.2024	Виконано

Здобувач вищої освіти _____ Федір ІСАЧЕНКО
(підпис) (**Власне ім'я, ПРІЗВИЩЕ**)

Керівник роботи _____ Дмитро ЛАНДЕ
(підпис) (**Власне ім'я, ПРІЗВИЩЕ**

РЕФЕРАТ

Пояснювальна записка до дипломного проєкту: 59 с., 6 рис., 4 табл., 12 джерела.

Об’єкт дослідження: Процеси виявлення веб загроз у мережевому трафіку.

Предмет дослідження: Використання методів машинного навчання для автоматизації виявлення веб загроз.

Мета дослідження: Розробити та впровадити модель машинного навчання для аналізу мережевого трафіку та визначення його шкідливості.

Методи дослідження: Аналіз літературних джерел, математичне та комп’ютерне моделювання, обчислювальні експерименти.

Отримані результати: У ході дослідження було розглянуто ключові аспекти застосування методів машинного навчання для виявлення веб загроз. Спочатку було проведено аналіз літератури та існуючих рішень у сфері кібербезпеки з акцентом на використанні алгоритмів класифікації, кластеризації, виявлення аномалій та глибокого навчання. Було зібрано та підготовлено датасет з реальним мережевим трафіком, який містить нормальні та аномальні запити.

Для створення моделі використовувалися такі алгоритми, як RandomForest, GradientBoosting, AdaBoost, LogisticRegression, KNeighbors, DecisionTree та GaussianNB. Було проведено тренування та оцінка моделей з використанням метрик точності, повноти, F1-міри та ROC AUC. Результати показали, що модель RandomForest досягла найвищої точності у виявленні веб загроз. Також було здійснено візуалізацію важливості ознак для моделі, що допомогло краще зрозуміти вплив різних факторів на процес класифікації.

Після тренування модель була інтегрована у систему моніторингу мережевого трафіку, що дозволило автоматизувати процес виявлення загроз у

реальному часі. Практичне застосування моделі було продемонстровано на прикладі виявлення атак, таких як SQL-ін’єкції та XSS-атаки. Оцінка ефективності системи показала високий рівень точності та швидкості виявлення атак, що перевершує традиційні методи.

Рекомендації та напрямки подальшого розвитку включають вдосконалення моделі шляхом використання додаткових ознак та алгоритмів, а також інтеграцію системи з іншими інструментами кібербезпеки для підвищення загального рівня захисту.

Ключові слова: веб загрози, машинне навчання, кібербезпека, аналіз мережевого трафіку, автоматизація процесів, виявлення атак, інциденти безпеки.

ABSTRACT

Explanatory note to the diploma project: 59 pages, 6 figures, 4 tables, 12 references.

Object of research: Processes of detecting web threats in network traffic.

Subject of research: The use of machine learning methods for automating web threat detection.

Purpose of the research: To develop and implement a machine learning model for analyzing network traffic and determining its maliciousness.

Research methods: Analysis of literary sources, mathematical and computer modeling, computational experiments.

Obtained results: The study examined key aspects of applying machine learning methods for detecting web threats. Initially, a literature review and analysis of existing solutions in cybersecurity were conducted, focusing on the use of classification, clustering, anomaly detection, and deep learning algorithms. A dataset containing real network traffic, including normal and anomalous requests, was collected and prepared.

For model creation, algorithms such as RandomForest, GradientBoosting, AdaBoost, LogisticRegression, KNeighbors, DecisionTree, and GaussianNB were used. Models were trained and evaluated using metrics like accuracy, precision, recall, F1-score, and ROC AUC. The results showed that the RandomForest model achieved the highest accuracy in detecting web threats. Additionally, feature importance visualization was conducted, providing insights into the impact of various factors on the classification process.

After training, the model was integrated into a network traffic monitoring system, allowing for the automation of threat detection in real-time. The practical application of the model was demonstrated through detecting attacks such as SQL injection and XSS attacks. System performance evaluation indicated a high level of accuracy and speed in detecting attacks, surpassing traditional methods.

Recommendations and further development directions include improving the model by incorporating additional features and algorithms, as well as integrating the system with other cybersecurity tools to enhance overall protection.

Keywords: web threats, machine learning, cybersecurity, network traffic analysis, process automation, attack detection, security incidents.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

DoS – Denial-of-Service

DDoS – Distributed Denial-of-Service

IP – Internet Protocol

IDS – Intrusion Detection System

ML – Machine Learning

SIEM – Security Information and Event Management

SOAR – Security Orchestration, Automation, and Response

AI – Artificial Intelligence

JSON – JavaScript Object Notation

RAM – Random Access Memory

URI – Uniform Resource Identifier

URL – Uniform Resource Locator

DNS – Domain Name System

HTTP – Hypertext Transfer Protocol

FTP – File Transfer Protocol

SMTP – Simple Mail Transfer Protocol

XSS – Cross-Site Scripting

SQLI – SQL Injection

CSRF – Cross-Site Request Forgery

ROC AUC – Receiver Operating Characteristic Area Under the Curve

RNN – Recurrent Neural Network

CNN – Convolutional Neural Network

LSTM – Long Short-Term Memory

GRU – Gated Recurrent Unit

F1-score – F1 Measure (Harmonic Mean of Precision and Recall)

TP – True Positives

FP – False Positives

TN – True Negatives

FN – False Negatives

NLP – Natural Language Processing

ВСТУП

Розвиток новітніх технологій та значне поширення Інтернету суттєво змінили наше життя, створивши не тільки численні можливості, а й нові виклики, особливо у сфері безпеки. Проблема веб-загроз стала надзвичайно актуальною в сучасному інформаційному суспільстві, адже ризики зловмисних атак постійно зростають. Таким чином, ефективні методи виявлення веб-загроз стають критичним напрямком досліджень у галузі кібербезпеки. Зважаючи на стрімкий розвиток інформаційних технологій, актуальність теми пов’язана з постійним збільшенням кількості та складності веб-загроз. Кожного дня з’являються нові типи атак, які орієнтовані на крадіжку інформації, порушення роботи систем або нанесення економічних збитків. Традиційні методи виявлення вже не можуть забезпечити належний рівень захисту, тому виникає потреба у впровадженні більш сучасних та адаптивних рішень.

Машинне навчання пропонує потужні інструменти для аналізу великих обсягів даних та виявлення складних патернів, що дозволяє ідентифікувати загрози більш ефективно. Основною ціллю даного дослідження є розробка новітніх методів для ефективного виявлення веб-загроз з використанням технологій машинного навчання. Серед завдань дослідження – аналіз існуючих методів, розробка та тестування нових алгоритмів, а також оцінка їх ефективності у реальних умовах.

Методологія дослідження включає комплексний підхід до аналізу даних, використання різноманітних технік машинного навчання для виявлення аномалій та класифікації загроз, а також тестування розроблених методів на реальних датасетах. Аналіз літератури та попередніх досліджень дозволить визначити поточний стан проблеми, виявити прогалини та запропонувати нові підходи. Використання числових методів та комп’ютерного моделювання забезпечить обґрунтованість та надійність отриманих результатів. Структура дипломної роботи побудована таким чином, щоб забезпечити логічний і послідовний виклад матеріалу.

Початкові розділи присвячені теоретичним основам та огляду основних понять, далі подається детальний аналіз методів машинного навчання, їх застосування у сфері виявлення веб-загроз та оцінка ефективності різних підходів. Завершальні частини дозволяють зробити висновки та окреслити напрями майбутніх досліджень.

Таким чином, дана робота спрямована на дослідження та вирішення однієї з найактуальніших проблем сучасного інформаційного суспільства – виявлення веб-загроз за допомогою машинного навчання. Застосування новітніх технологій дозволить підвищити рівень безпеки інформаційних систем та зменшити ризики, пов’язані з кіберзлочинністю.

1 ТЕОРИТИЧНІ ОСНОВИ ВЕБ ЗАГРОЗ ТА МАШИННОГО НАВЧАННЯ

1.1 Огляд основних понять і термінології

Веб загрози – це будь-які потенційно небезпечні дії або умови, які можуть вплинути на безпеку веб-додатків або даних, що передаються через Інтернет. До основних видів веб загроз належать:

Фішинг: метод шахрайства, що передбачає отримання конфіденційної інформації шляхом маскування під надійні джерела.

Зловмисне програмне забезпечення (Malware): шкідливі програми, які поширюються через веб-додатки та можуть викрадати дані, пошкоджувати файли або отримувати несанкціонований доступ до систем.

SQL-ін'єкції: тип атаки, при якій зловмисник вводить шкідливий SQL-код у запити до бази даних, щоб отримати доступ до захищеної інформації.

Міжсайтовий скриптинг (XSS): атака, що дозволяє впровадити шкідливий код у веб-сторінки, який буде виконаний на стороні користувача.

Машинне навчання – це підгалузь штучного інтелекту, що займається розробкою алгоритмів, які дозволяють комп'ютерам навчатися на основі даних. В контексті виявлення веб загроз, машинне навчання використовується для автоматизації процесу виявлення аномалій та шкідливих активностей у мережевому трафіку.

Основні терміни машинного навчання:

Модель: математичне представлення, яке використовується для прийняття рішень або прогнозування на основі даних.

Навчальна вибірка: набір даних, що використовується для навчання моделі.

Тестова вибірка: набір даних, що використовується для перевірки точності моделі після навчання.

Класифікація: процес присвоєння категорій новим даним на основі навчальної вибірки.

Кластеризація: процес групування подібних об'єктів без попереднього знання категорій.

Аномальне виявлення: виявлення відхилень від нормальної поведінки в даних.

Машинне навчання має значний потенціал у виявленні веб загроз завдяки своїй здатності аналізувати великі обсяги даних та виявляти складні патерни, які можуть бути недоступні для традиційних методів. Наприклад, у дослідженні, присвяченому моделюванню мережевих концепцій, відзначено ефективність методів машинного навчання в аналізі взаємозв'язків між різними об'єктами та подіями [11].

В іншому дослідженні було показано, що методи машинного навчання можуть значно покращити ефективність виявлення зловмисного програмного забезпечення шляхом оптимізації методів вилучення ознак та застосування алгоритмів машинного навчання для аналізу файлів [12].

Ці основні поняття та терміни становлять фундамент для подальшого розгляду методів виявлення веб загроз за допомогою машинного навчання.

1.2 Методи машинного навчання для виявлення веб загроз

Методи машинного навчання відіграють ключову роль у виявленні веб загроз завдяки їх здатності аналізувати великі обсяги даних та виявляти приховані

патерни. Основні методи, що застосовуються для виявлення веб загроз, включають класифікацію, кластеризацію, аномалійне виявлення та глибоке навчання.

Класифікація є одним з найпоширеніших методів машинного навчання. Вона використовується для присвоєння вхідних даних до однієї з декількох категорій на основі навчальної вибірки. Для виявлення веб загроз класифікатори можуть навчатися на історичних даних про атаки, щоб розпізнавати нові загрози. Наприклад, алгоритми, такі як логістична регресія, метод опорних векторів (SVM) та дерева рішень, широко застосовуються для класифікації загроз [10].

Кластеризація використовується для групування схожих об'єктів без попереднього знання категорій. Цей метод корисний для виявлення нових типів загроз або виявлення схожих атак у великих наборах даних. Наприклад, методи, такі як k-means та ієрархічна кластеризація, дозволяють аналізувати та виявляти схожі патерни в поведінці мережевого трафіку [2].

Аномалійне виявлення орієнтоване на виявлення відхилень від нормальної поведінки. Цей метод особливо ефективний для виявлення нових або невідомих загроз, які не були враховані під час навчання моделі. Методи, такі як ізоляційний ліс, метод локального рівня аномалій (LOF) та автоматичні кодувальники, дозволяють ідентифікувати аномалії в мережевому трафіку [14].

Глибоке навчання є підвидом машинного навчання, який використовує багаторівневі нейронні мережі для аналізу даних. Завдяки своїй здатності обробляти великі обсяги даних та виявляти складні патерни, глибоке навчання стає все більш популярним у виявленні веб загроз. Нейронні мережі, такі як рекурентні нейронні мережі (RNN) та згорткові нейронні мережі (CNN), застосовуються для аналізу поведінки трафіку та виявлення шкідливих дій.

Використання машинного навчання для виявлення веб загроз дозволяє значно підвищити ефективність та швидкість виявлення шкідливих дій. Наприклад,

дослідження показують, що застосування алгоритмів машинного навчання, таких як SVM та глибокі нейронні мережі, дозволяє значно покращити точність виявлення загроз [14]. Крім того, методи машинного навчання дозволяють автоматизувати процеси аналізу даних та зменшити вплив людського фактора на процес виявлення загроз, що особливо важливо в умовах зростаючої кількості та складності веб атак.

Застосування цих методів забезпечує потужний інструментарій для виявлення та протидії веб загрозам, що дозволяє ефективніше захищати веб-додатки та їх користувачів від сучасних загроз.

1.2.1 Класифікація

Класифікація є одним з найважливіших методів машинного навчання, який використовується для виявлення веб загроз. Цей метод передбачає розподіл вхідних даних до однієї з кількох категорій на основі навчальної вибірки. Класифікація є надзвичайно корисною для виявлення шкідливих дій та загроз, оскільки дозволяє моделі навчитися розпізнавати патерни, які відповідають різним типам атак.

Існує декілька поширених алгоритмів класифікації, які застосовуються для виявлення веб загроз:

1. **Логістична регресія**: Цей метод використовується для прогнозування ймовірності настання певної події. Він добре підходить для задач бінарної класифікації, наприклад, визначення, чи є певний трафік шкідливим або безпечним.[9]

2. **Метод опорних векторів (SVM)**: SVM використовується для розділення даних на дві категорії за допомогою гіперплощини. Цей метод ефективний для виявлення веб загроз завдяки своїй здатності знаходити

оптимальну гіперплощину, яка максимально віддалена від об'єктів обох класів.[10]

3. **Дерева рішень**: Цей алгоритм будує модель у вигляді дерева, де кожна внутрішня вершина відповідає за перевірку певної ознаки, кожна гілка відповідає за результат цієї перевірки, а кожен лист – за кінцевий клас. Дерева рішень зручні для інтерпретації та можуть бути дуже ефективними для виявлення складних патернів у даних.

4. **Наївний Баєсів класифікатор**: Цей метод базується на застосуванні теореми Баєса з припущенням про незалежність ознак. Він швидкий у навчанні та застосуванні, що робить його корисним для задач, де швидкість є критично важливою.[5]

5. **Нейронні мережі**: Хоча нейронні мережі часто асоціюються з глибоким навчанням, вони також можуть використовуватися для задач класифікації. Прості нейронні мережі, такі як багатошарові персептрони, можуть бути ефективними для виявлення веб загроз.[4]

Використання методів класифікації дозволяє ефективно обробляти великі обсяги даних і швидко ідентифікувати шкідливу активність. Ці методи постійно вдосконалюються, забезпечуючи більш високу точність і надійність виявлення веб загроз.

1.2.2 Кластеризація

Кластеризація є одним із основних методів машинного навчання, що використовується для аналізу даних без нагляду. Цей метод передбачає групування об'єктів у наборі даних таким чином, щоб об'єкти в одному кластері були більш схожими між собою, ніж з об'єктами з інших кластерів. У контексті кібербезпеки

кластеризація є надзвичайно корисною для виявлення аномалій у мережевому трафіку, класифікації зразків шкідливого програмного забезпечення та інших завдань.

Основні алгоритми кластеризації включають:

1. **K-середні (K-means)**: один з найпопулярніших алгоритмів кластеризації, що розбиває набір даних на K кластерів. Процес починається з вибору K початкових центрів кластерів, після чого кожен об'єкт призначається до найближчого центру. Після цього центри кластерів оновлюються до середнього значення об'єктів у кластері, і процес повторюється до досягнення стабільності.

2. **Ієрархічна кластеризація**: цей метод створює дерево кластерів (дендрограму), починаючи з кожного об'єкта як окремого кластера та поступово об'єднуючи найближчі кластери. Ієрархічна кластеризація може бути агломеративною (знизу вверх) або дивізивною (зверху вниз).

3. **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)**: алгоритм, що визначає кластери як області з високою густиною точок, відокремлені областями з низькою густиною. DBSCAN добре працює для даних з шумом та кластерами складної форми.

Кластеризація є потужним інструментом у кібербезпеці, оскільки дозволяє аналізувати великі обсяги даних та виявляти патерни, які можуть бути ознакою загроз або аномалій. Наприклад, за допомогою алгоритму K-means можна групувати мережеві потоки для виявлення підозрілої активності, а DBSCAN може бути використаний для ідентифікації неочікуваних поведінкових змін у мережевому трафіку.

У "Machine Learning for Cybersecurity Cookbook" [10] описано використання кластеризації для виявлення шкідливого програмного забезпечення. Зокрема,

демонструється застосування алгоритму K-means для класифікації зразків шкідливого програмного забезпечення на основі їх характеристик. Процес включає підготовку даних, навчання моделі та візуалізацію результатів кластеризації, що дозволяє виявити основні патерни у зразках шкідливого програмного забезпечення та класифікувати їх відповідно до сімейств.

1.2.3 Аномалійне виявлення

Аномалійне виявлення є важливим методом машинного навчання для виявлення веб загроз. Цей метод фокусується на виявленні відхилень від нормальної поведінки в даних, що може свідчити про потенційні загрози або атаки. Аномалії можуть бути ознакою нових або невідомих загроз, які не були враховані під час навчання моделей.

Ізоляційний ліс (Isolation Forest) є одним з ефективних методів виявлення аномалій. Цей алгоритм базується на ідеї, що аномалії легше ізолювати, ніж нормальні точки даних. Ізоляційний ліс створює випадкові підмножини даних та будує деревовидні структури для ізоляції аномальних точок. Чим коротший шлях до ізоляції точки, тим вищою є ймовірність, що ця точка є аномалією. Цей метод ефективно використовується для виявлення аномалій у великих наборах даних, таких як мережевий трафік.[10]

Метод локального рівня аномалій (Local Outlier Factor, LOF) оцінює аномальність кожної точки даних на основі локальної щільності її сусідів. Порівняння локальної щільності точки з її сусідами дозволяє виявити аномалії, які мають значно нижчу щільність порівняно з сусідніми точками. LOF широко використовується для виявлення аномалій у даних з нерівномірною щільністю, що

робить його корисним для аналізу мережевого трафіку та виявлення підозрілої активності.

Застосування аномалійного виявлення дозволяє автоматизувати процеси аналізу даних та зменшити вплив людського фактора на процес виявлення загроз. Це особливо важливо у контексті сучасних кіберзагроз, де швидкість та точність виявлення можуть мати критичне значення для захисту інформаційних систем.

1.2.4 Глибоке навчання

Глибоке навчання є підгалуззю машинного навчання, яка використовує багаторівневі нейронні мережі для аналізу складних даних. Цей підхід показав високу ефективність у виявленні веб загроз завдяки своїй здатності виявляти складні патерни та аномалії у великих наборах даних. Глибоке навчання стало популярним завдяки успіхам у різних областях, таких як обробка зображень, розпізнавання мови та природна обробка тексту.

Одним із ключових аспектів глибокого навчання є використання **згорткових нейронних мереж (Convolutional Neural Networks, CNN)** для обробки даних, які мають просторову структуру, таких як зображення або мережевий трафік. CNN ефективно розпізнають патерни та можуть бути використані для виявлення шкідливих дій у веб трафіку, аналізуючи послідовності пакетів даних.

Рекурентні нейронні мережі (Recurrent Neural Networks, RNN), зокрема їхні розширення, такі як довго-короткочасна пам'ять (Long Short-Term Memory, LSTM) та мережі на основі гейтів (Gated Recurrent Unit, GRU), використовуються для аналізу послідовних даних. Вони можуть ефективно обробляти тимчасові залежності в даних і використовуються для аналізу мережевого трафіку з метою виявлення аномалій та шкідливих дій.

Автокодувальники (Autoencoders) є ще одним важливим інструментом у глибокому навчанні. Вони використовуються для навчання компактного представлення даних, яке зберігає найбільш важливі особливості вхідних даних. Автокодувальники можуть виявляти аномалії на основі помилки відтворення: якщо точка даних суттєво відрізняється від нормальних даних, автокодувальник не зможе правильно її відтворити, що призведе до великої помилки.

Застосування глибокого навчання для виявлення веб загроз забезпечує високий рівень автоматизації та точності. Наприклад, дослідження показують, що використання CNN для аналізу мережевого трафіку дозволяє ефективно виявляти шкідливі дії, зокрема атаки типу "відмова у обслуговуванні" та інші форми шкідливої активності.

Крім того, глибоке навчання дозволяє обробляти великі обсяги даних у реальному часі, що є критично важливим для сучасних систем кібербезпеки. Використання нейронних мереж для аналізу даних забезпечує високу точність виявлення та зменшує кількість хибнопозитивних спрацьовувань, що дозволяє зосередитися на дійсно важливих загрозах.

Таким чином, глибоке навчання є потужним інструментом для забезпечення безпеки веб-додатків, що дозволяє ефективно виявляти та протидіяти сучасним кіберзагрозам.

1.3 Проблеми та виклики у виявленні веб загроз

Виявлення веб загроз за допомогою машинного навчання стикається з багатьма проблемами та викликами. Деякі з них пов'язані з технічними аспектами, інші - з організаційними та правовими питаннями.

Великий обсяг даних: Сучасні інформаційні системи генерують

величезні обсяги даних, що ускладнює їх обробку та аналіз. Для ефективного виявлення загроз необхідно розробити методи, які можуть швидко аналізувати великі масиви даних та виявляти аномалії. Використання глибокого навчання та інших методів машинного навчання може допомогти вирішити цю проблему, однак це вимагає значних обчислювальних ресурсів.

Адаптація до нових загроз: Кіберзагрози постійно еволюціонують, і нові типи атак можуть не бути розпізнаними існуючими системами. Традиційні методи виявлення, такі як підхід на основі підписів, не здатні ефективно боротися з новими загрозами. Машинне навчання пропонує можливість створення моделей, що можуть адаптуватися до нових типів атак, проте навчання цих моделей потребує значного часу та даних.

Хибнопозитивні спрацьовування: Однією з основних проблем систем виявлення загроз є велика кількість хибнопозитивних спрацьовувань, що може призводити до зайвих витрат часу та ресурсів на розслідування. Вдосконалення моделей машинного навчання та використання більш складних алгоритмів може допомогти зменшити кількість хибнопозитивних спрацьовувань, але це потребує постійного налаштування та оновлення моделей.

Безпека та конфіденційність даних: Для навчання моделей машинного навчання необхідні великі обсяги даних, що може викликати питання щодо конфіденційності та безпеки цих даних. Забезпечення безпеки даних під час їх збору, зберігання та обробки є критичним аспектом, який потребує особливої уваги.

Інтерпретація результатів: Результати, отримані за допомогою моделей машинного навчання, не завжди легко інтерпретувати. Це може ускладнити прийняття рішень на основі цих результатів. Розробка методів інтерпретованого машинного навчання є одним з напрямків, що активно розвивається, і може допомогти вирішити цю проблему.

Технічні обмеження: Використання складних моделей машинного навчання вимагає значних обчислювальних ресурсів та технічної інфраструктури. Це може бути проблематичним для організацій з обмеженими ресурсами.

Таким чином, хоча машинне навчання має великий потенціал для виявлення веб загроз, існує ряд викликів, які необхідно вирішити для ефективного використання цих технологій.

1.4 Сучасні системи та технології

У сучасному світі кібербезпеки, численні комерційні та некомерційні рішення активно використовують машинне навчання для виявлення веб-загроз. Ось деякі з них:

Комерційні рішення

Darktrace використовує передові методи машинного навчання для виявлення та реагування на кіберзагрози в реальному часі. Система постійно аналізує мережевий трафік та поведінкові патерни, виявляючи аномалії, які можуть свідчити про потенційні загрози. Darktrace здатна адаптуватися до нових типів атак завдяки своєму самонавчальному алгоритму, що робить її ефективним інструментом для захисту організацій від сучасних кіберзагроз.

CrowdStrike Falcon використовує хмарні технології та машинне навчання для виявлення та запобігання загрозам на кінцевих точках. Система аналізує поведінкові дані в реальному часі, ідентифікуючи аномалії та нові загрози. CrowdStrike Falcon забезпечує високий рівень захисту завдяки своїй здатності швидко адаптуватися до нових видів атак.

Cisco Umbrella надає комплексний захист для організацій, використовуючи методи машинного навчання для аналізу DNS-запитів та виявлення шкідливої

активності. Система може блокувати доступ до шкідливих веб-сайтів та запобігати фішинговим атакам на основі аналізу поведінкових патернів.

Некомерційні рішення

Snort - це безкоштовна система виявлення вторгнень (IDS), яка використовує сигнатурний та поведінковий аналіз для виявлення шкідливого трафіку. Snort широко використовується в академічних та дослідницьких цілях завдяки своїй відкритій архітектурі та потужним можливостям аналізу.

OSSEC (Open Source HIDS SECurity) - це система виявлення вторгнень на основі хостів, яка використовує машинне навчання для аналізу лог-файлів, файлів конфігурацій та мережевої активності. OSSEC є ефективним інструментом для виявлення аномалій та запобігання шкідливим діям на хостах.

Apache Spot - це платформа для аналізу кібербезпеки, яка використовує методи машинного навчання для виявлення аномалій у мережевому трафіку. Apache Spot дозволяє збирати, зберігати та аналізувати великі обсяги даних, забезпечуючи ефективне виявлення та реагування на загрози.

Приклади успішного впровадження:

Виявлення ботнетів: У книзі "Machine Learning for Cybersecurity Cookbook"[10] описується використання алгоритмів машинного навчання для виявлення ботнетів у реальному трафіку. Система використовує методи класифікації для аналізу поведінкових патернів та ідентифікації шкідливої активності у мережевому трафіку .

Виявлення аномалій у промислових системах: У роботі "Machine Learning for Cybersecurity: Innovative Deep Learning Solutions"[5] описується використання глибоких нейронних мереж для виявлення аномалій у промислових системах управління (ICS). Цей підхід дозволяє виявляти складні атаки, які важко ідентифікувати за допомогою традиційних методів.

Сучасні системи та технології виявлення веб-загроз продовжують розвиватися, забезпечуючи більш високий рівень захисту та ефективності завдяки використанню передових методів машинного навчання та штучного інтелекту.

2 ОГЛЯД ДАТАСЕТУ ТА ПОПЕРЕДНІ РОЗРОБКИ

Цей розділ охоплює огляд датасету та попередню розробу у галузі виявлення веб загроз за допомогою машинного навчання, з особливою увагою до аналізу та підготовки датасету CSIC-2010. Основна увага приділяється вивченню методів машинного навчання, які застосовуються для виявлення різноманітних загроз у веб середовищі, а також аналізу існуючих систем та технологій.

Розглядаються основні поняття і термінологія, що стосуються веб загроз та методів їхнього виявлення, що створює базис для подальшого обговорення і забезпечує узгоджене розуміння термінів та концепцій. Методи машинного навчання для виявлення веб загроз можуть бути різними, зокрема, класифікація, кластеризація, аномалійне виявлення та глибоке навчання. Класифікація досліджує можливість розподілу вхідних даних на визначені категорії, тоді як кластеризація дозволяє групувати схожі елементи, що корисно для знаходження нових, раніше невідомих загроз. Аномалійне виявлення сфокусовано на виявленні відхилень від нормальної поведінки, що може свідчити про наявність загроз. Глибоке навчання використовується для побудови складних моделей, здатних розпізнавати різноманітні патерни у даних. Використання NLP (обробка природної мови) для аналізу зловмисного коду дозволяє виявляти загрози на основі мовних характеристик, а аналіз часових рядів допомагає відстежувати зміни у поведінці систем з плином часу, що надає можливість ідентифікувати потенційні загрози в динаміці.

Огляд літератури та попередніх досліджень висвітлює вже існуючі напрацювання у цій галузі, підкреслюючи ключові результати та підходи, використані іншими науковцями та дослідниками. Це забезпечує контекст для подальших досліджень та експериментів.

Аналіз датасету CSIC-2010 зосереджений на характеристиках запитів до веб-додатка, які можуть бути нормальними або аномальними. Досліджуються основні атрибути, такі як HTTP-метод, інформація про клієнтське програмне забезпечення, інструкції кешування, прийнятні формати даних, мова запиту, хост, дані кукісів, тип вмісту, інформація про з'єднання, довжина вмісту, повний URL запиту та класифікаційні мітки.

Підготовка датасету включає видалення нерелевантних характеристик, обробку пропущених значень, нормалізацію даних та вибір релевантних характеристик. Під час підготовки даних звертається увага на розподіл класів, обробку пропущених значень, перетворення та нормалізацію даних, а також виділення важливих характеристик. Окрему увагу приділяється витягуванню характеристик з URL, таких як кількість точок, директорій, довжина URL та наявність спеціальних символів, що допомагає виявляти складні та підозрілі запити.

Цей розділ надає цілісне уявлення про теоретичні основи та попередні розробки у сфері виявлення веб загроз за допомогою машинного навчання, закладаючи основу для подальших експериментів та розробки нових підходів у цій динамічній та важливій галузі.

2.1 Огляд датасету та аналіз датасету

2.1.1 Огладя датасету

Датасет CSIC-2010 містить різноманітні атрибути, які описують запити до веб-додатка. Кожен рядок представляє один запит, і ці запити можуть бути як нормальними, так і аномальними (тобто, веб-атаками). Основні характеристики цього датасету:

Class (Unnamed: 0):

- Вказує на тип запиту: Normal для нормальних запитів та Anomalous для аномальних запитів.
- Є цільовою змінною для класифікації.

Method:

- HTTP-метод, що використовується в запиті, наприклад, GET або POST.
- Важливий для аналізу типу взаємодії з веб-додатком.

User-Agent:

- Інформація про клієнтське програмне забезпечення, що відправляє запит.
- Допомагає ідентифікувати програмне забезпечення або пристрій.

Pragma та Cache-Control:

- Інструкції кешування, що можуть бути корисними для аналізу поведінки запитів.

Accept, Accept-encoding, Accept-charset:

- Інформація про формати даних, які клієнт може приймати.
- Впливають на вибірковість обробки запитів.

Language:

- Мова, яка використовується в запиті.
- Корисна для аналізу регіональних відмінностей у трафіку.

Host:

- Хост, до якого направлено запит.
- Допомагає виявляти атаки, спрямовані на конкретні домени.

Cookie:

- Дані, що зберігаються в кукісах, відправлені разом із запитом.
- Важливі для аутентифікації та ідентифікації сесій.

Content-Type:

- Тип вмісту, переданий у запиті.
- Ключовий атрибут для аналізу даних POST-запитів.

Connection:

- Інформація про з'єднання, наприклад, чи слід його закривати після відповіді.
- Допомагає розуміти типи використаних з'єднань.

Content-Length:

- Довжина вмісту, переданого у запиті.
- Важлива для аналізу обсягу даних, переданих у запиті.

Content:

- Вміст запиту, якщо він є.
- Містить ключові дані, які можуть бути використані для аналізу і виявлення шкідливої активності.

Classification

- Класифікаційна мітка (0 - нормальний, 1 — аномальний).
- Додаткова цільова змінна, що використовується для виявлення аномалій.

URL:

- Повний URL запиту.
- Важливий атрибут для аналізу структури і складності запитів.

2.1.2 Аналіз датасету

При підготовці датасету для моделей машинного навчання ми звернемо особливу увагу на наступні аспекти:

Розподіл класів:

Важливо зрозуміти, чи є датасет збалансованим за кількістю нормальних і аномальних запитів. Якщо ні, необхідно буде застосувати методи для збалансування класів.

Обробка пропущених значень:

Необхідно обробити пропущені значення, які можуть негативно вплинути на продуктивність моделі. Використаємо методи заповнення пропущених даних або видалення таких записів.

Перетворення та нормалізація даних:

Перетворення категоріальних даних у числові значення, використання методів кодування.

Нормалізація числових даних для забезпечення коректної роботи алгоритмів машинного навчання.

Виділення важливих характеристик:

Аналіз важливості характеристик для класифікації. Використання методів, таких як Random Forest, для визначення найбільш впливових характеристик.

Аналіз URL та вмісту запитів:

Витягнення додаткових характеристик з URL та вмісту запитів, таких як кількість точок, директорій, довжина, наявність підозрілих слів тощо.

2.2 Підготовка датасету

Підготовка даних є важливим кроком у процесі машинного навчання, оскільки якість даних безпосередньо впливає на ефективність та точність моделі. У цьому розділі ми детально розглянемо етапи підготовки датасету CSIC-2010 для виявлення веб-загроз, зокрема, видалення нерелевантних характеристик, обробку пропущених значень, нормалізацію даних та вибір релевантних характеристик.

Видалення нерелевантних характеристик

Першим кроком було визначення та видалення характеристик, які не несуть дискримінативної інформації для нашої задачі.

- **Вибір характеристик:** Ми обрали характеристики, які потенційно можуть бути корисними для нашої моделі, включаючи Unnamed: 0, Method, User-Agent, Pragma, Cache-Control, Accept, Accept-encoding, Accept-charset, language, host, cookie, content-type, connection, length, content, classification, URL.

- **Перейменування колонок:** Для покращення читабельності та коректності даних ми перейменували колонку Unnamed: 0 на Class, а length на content_length.

Вибір релевантних характеристик

Наступним кроком був вибір релевантних характеристик, що можуть значно вплинути на рішення моделі.

- **Основні характеристики:** Ми визначили такі основні характеристики, як Class, Method, host, cookie, Accept, content_length, content, classification, URL.

- **Виділення цільової змінної:** Цільовою змінною для нашої моделі є Class, що вказує на те, чи є запит нормальним чи аномальним.

Обробка пропущених значень

Пропущені значення можуть значно вплинути на результати моделі, тому їх потрібно обробити належним чином.

- **Заповнення пропущених значень:** Ми заповнили пропущені значення в колонці content_length нулями, а потім перетворили значення цієї колонки в числовий формат. Для цього ми використовували методи fillna та pd.to_numeric.

- **Обробка текстових даних:** Для категоріальних характеристик, таких як Method, ми застосували методи кодування для перетворення їх у числовий формат. Ми використали LabelEncoder для кодування таких характеристик.

Витягування характеристик з URL

URL є одним з найважливіших атрибутів, який може містити багато корисної інформації про потенційні загрози.

- **Кількість точок у URL:** Ми обчислювали кількість точок у URL, що може вказувати на складність URL і наявність вбудованих доменів.
- **Кількість директорій у URL:** Ця характеристика вказує на глибину вкладеності сторінок.
- **Кількість вбудованих доменів:** Може бути індикатором потенційно шкідливих запитів.
- **Довжина URL та хостнейму:** Довгі URL або хостнейми можуть вказувати на використання складних або підозрілих доменів.
- **Кількість спеціальних символів, цифр, літер та параметрів:** Ці характеристики допомагають виявити складність і потенційну шкідливість URL.

Нормалізація та кодування даних

Для забезпечення коректної роботи моделі необхідно було нормалізувати та закодувати дані.

- **Нормалізація даних:** Ми нормалізували числові дані для приведення їх до єдиного масштабу, що дозволяє уникнути домінування однієї характеристики над іншими.
- **Кодування категоріальних даних:** Ми використали LabelEncoder для перетворення категоріальних даних у числовий формат, придатний для моделей машинного навчання.

2.3 Класифікація і вибір моделі

Для визначення найбільш підходящих алгоритмів машинного навчання для виявлення веб-загроз, ми використовуємо кілька моделей і оцінюємо їх продуктивність. Розглянемо основні методи класифікації, які були застосовані для аналізу датасету.

Моделі для класифікації

Ми розглянемо кілька моделей машинного навчання, що добре зарекомендували себе у задачах виявлення аномалій та загроз. Кожна з цих моделей має свої переваги і недоліки:

Random Forest

Опис: Ансамблева методика, що використовує множину дерев рішень для покращення точності та зниження ризику перенавчання.

Переваги: Висока точність, стійкість до перенавчання, здатність працювати з великою кількістю характеристик.

Недоліки: Може бути повільною при обробці дуже великих датасетів.

Gradient Boosting

Опис: Методика ансамблювання, побудована на послідовному навчанні слабких моделей (дерев рішень), де кожна наступна модель намагається виправити помилки попередніх.

Переваги: Висока точність, хороша продуктивність на складних даних.

Недоліки: Може бути схильна до перенавчання, якщо не контролювати параметри.

AdaBoost

Опис: Ансамблевий алгоритм, що комбінує кілька слабких моделей для створення сильної моделі, приділяючи більше уваги неправильним передбаченням.

Переваги: Хороша продуктивність на невеликих наборах даних, стійкість до перенавчання.

Недоліки: Може бути чутливим до шуму в даних.

Logistic Regression

Опис: Алгоритм для бінарної класифікації, що моделює ймовірність належності зразка до одного з двох класів.

Переваги: Простота реалізації, швидкість, інтерпретованість результатів.

Недоліки: Обмежена здатність до моделювання нелінійних зв'язків.

K-Nearest Neighbors (KNN)

Опис: Алгоритм, що класифікує зразок на основі його найближчих сусідів у просторі характеристик.

Переваги: Простота, інтуїтивна зрозумілість.

Недоліки: Повільність при великих обсягах даних, чутливість до шуму.

Support Vector Machine (SVM)

Опис: Алгоритм, що шукає гіперплощину, яка максимально відділяє класи у просторі характеристик.

Переваги: Висока точність, ефективність у випадках з малою кількістю характеристик.

Недоліки: Може бути повільним при обробці великих наборів даних, важко налаштовується.

Decision Tree

Опис: Алгоритм, що використовує дерево рішень для класифікації зразків.

Переваги: Простота, інтерпретованість.

Недоліки: Схильність до перенавчання.

Naive Bayes

Опис: Алгоритм, заснований на теоремі Байєса з припущенням про незалежність характеристик.

Переваги: Швидкість, простота реалізації, добре працює з малими наборами даних.

Недоліки: Припущення про незалежність характеристик може бути невірним.

Класифікація є оптимальним підходом для нашої задачі з кількох причин:

Чіткі метки класів: Ми маємо чітко визначені метки класів (нормальні та аномальні запити), що ідеально підходить для задачі класифікації.

Цільова змінна: Наявність цільової змінної (Class), яка визначає, чи є запит нормальним чи аномальним.

Порівняння моделей: Класифікація дозволяє легко порівнювати різні моделі та обирати найкращу на основі метрик продуктивності (точність, повнота, F1-міра тощо).

Отже, для нашої задачі виявлення веб-загроз найефективніше використовувати методи класифікації, які дозволяють точно ідентифікувати аномальні запити та забезпечують високий рівень точності і продуктивності.

2.4 Важливість властивостей

Визначення важливості властивостей (features) є ключовим етапом у процесі побудови моделей машинного навчання для виявлення веб-загроз. Це дозволяє:

1. Зрозуміти вплив характеристик: Важливість властивостей дає змогу оцінити, які з характеристик мають найбільший вплив на рішення моделі. Це допомагає зрозуміти, які аспекти веб-запитів є найбільш інформативними для виявлення загроз.

2. Оптимізувати моделі: Використання тільки найважливіших характеристик може спростити модель, зменшити її обчислювальні витрати та підвищити швидкість роботи без втрати точності.

3. Інтерпретувати результати: Аналіз важливості властивостей дозволяє пояснити, чому модель приймає ті чи інші рішення. Це особливо важливо в галузі кібербезпеки, де прозорість рішень допомагає довіряти системі захисту.

4. Поліпшити якість даних: Зосередження на найважливіших характеристиках дозволяє більш ефективно обробляти дані, видаляючи менш інформативні або шумні атрибути.

Висновки на основі аналізу важливості властивостей

Після виконання аналізу важливості властивостей за допомогою алгоритму Random Forest, ми отримали наступні результати:

1. Hostname Length: Довжина хостнейму є однією з найважливіших характеристик, оскільки вона може вказувати на підозрілу поведінку, наприклад, використання довгих, складних доменних імен для маскування шкідливих URL.

2. Count of Dots in URL: Кількість точок у URL також є важливим індикатором, оскільки більша кількість точок може вказувати на вбудовані домени, що часто використовуються у фішингових або спам-атаках.

3. Special Characters Count: Кількість спеціальних символів у URL може свідчити про спроби обходу фільтрів або ін'єкційні атаки.

4. Length of URL: Довжина URL має суттєвий вплив, оскільки дуже довгі URL часто використовуються для приховування шкідливих параметрів або введення користувача в оману.

5. Suspicious Words Score: Наявність підозрілих слів у URL є критичним індикатором шкідливої активності. Високий бал підозрілих слів може вказувати на наявність SQL-ін'єкцій або інших типів атак.

6. Count of Parameters in URL: Кількість параметрів у URL може вказувати на складність запиту. Великі кількості параметрів можуть бути ознакою спроби передачі великого обсягу даних або здійснення складних атак.

7. Encoding Indicators: Наявність кодування в URL може свідчити про спроби приховування шкідливих намірів або обходу фільтрів безпеки.

Висновки до розділу 2

У цьому розділі ми дослідили основні теоретичні основи та попередні розробки у галузі виявлення веб-загроз за допомогою машинного навчання. Основна увага була приділена аналізу методів машинного навчання, що застосовуються для виявлення загроз у веб-середовищі, а також обговоренню існуючих систем та технологій.

Перш за все, ми зрозуміли, що машинне навчання надає нові можливості для автоматизації та покращення процесу виявлення веб-загроз. Використання методів машинного навчання, таких як класифікація, аномалійне виявлення та глибоке навчання, дозволяє ефективно ідентифікувати загрози, які можуть бути невидимими для традиційних методів. Це особливо актуально у сучасному світі, де веб-загрози стають все більш поширеними і складними.

При підготовці датасету ми детально розглянули процес очищення, нормалізації, кодування та витягування релевантних характеристик. Ми зрозуміли, що якість даних безпосередньо впливає на ефективність та точність моделі. Видалення нерелевантних характеристик, обробка пропущених значень та нормалізація даних є критичними етапами підготовки, які забезпечують коректну роботу моделей машинного навчання.

Ми також дослідили різні моделі класифікації, такі як Random Forest, Gradient Boosting, AdaBoost, Logistic Regression та інші, і порівняли їх ефективність для виявлення веб-загроз. Було визначено, що класифікація є найбільш підходящим підходом для нашої задачі, оскільки вона дозволяє точно визначити нормальні та аномальні запити. На відміну від кластеризації, яка не використовує цільову змінну, класифікація забезпечує більш точне виявлення загроз завдяки наявності чітко визначених класів.

Аналіз важливості властивостей показав, які характеристики найбільше впливають на рішення моделі. Це дозволило оптимізувати модель, зосередившись на найбільш інформативних характеристиках, таких як довжина хостнейму, кількість точок у URL, кількість спеціальних символів, довжина URL та бал підозрілих слів. Такий підхід допомагає підвищити точність та ефективність моделі, а також спрощує процес інтерпретації результатів.

Таким чином, розділ 2 надав цілісне уявлення про теоретичні основи та практичні аспекти використання машинного навчання для виявлення веб-загроз. Це закладає міцну основу для подальших досліджень та розробок у цій галузі, допомагаючи зрозуміти ключові фактори успіху у виявленні загроз та оптимізації систем захисту. Отримані результати та висновки будуть критично важливими для подальшої реалізації та вдосконалення нашої системи виявлення веб-загроз.

3 РОЗРОБКА СИСТЕМИ ВИЯВЛЕННЯ ВЕБ ЗАГРОЗ

3.1 Постановка задачі

В умовах стрімкого розвитку технологій та зростання кількості інтернет-користувачів веб-загрози стають все більш поширеними і складними. Основна мета цієї роботи – розробити ефективну систему виявлення веб-загроз, що базується на методах машинного навчання. Система має бути здатною виявляти різноманітні типи загроз, такі як фішинг, шкідливі програми, SQL-ін'єкції та інші атаки, що загрожують безпеці веб-додатків та користувачів.

Для досягнення цієї мети необхідно вирішити наступні завдання:

1. Збір та підготовка даних, які будуть використовуватися для навчання та тестування моделей.
2. Вибір та налаштування алгоритмів машинного навчання, що підходять для виявлення веб-загроз.
3. Розробка та впровадження моделей для класифікації, кластеризації, аномалійного виявлення та аналізу з використанням глибоких нейронних мереж.
4. Тестування та валідація розроблених моделей, оцінка їх ефективності та точності.
5. Впровадження розробленої системи у реальне середовище, інтеграція з існуючими системами захисту та моніторинг її роботи.
6. Аналіз результатів роботи системи, порівняння з існуючими підходами, визначення переваг та недоліків запропонованого рішення.

Поставлена задача передбачає створення багатокомпонентної системи, яка використовуватиме різні підходи та методи машинного навчання для забезпечення максимальної ефективності виявлення веб-загроз.

3.2 Збір та підготовка даних

3.2.1 Опис використаного набору даних

Для реалізації системи виявлення веб-загроз використовується набір даних "CSIC 2010 Web Application Attacks". Цей набір даних містить трафік веб-застосунків, включаючи нормальний трафік і атаки, такі як XSS (Cross-Site Scripting), SQLI (SQL Injection), CSRF (Cross-Site Request Forgery) та інші аномалії, сумарно 61 тисяча записів. Оригінальні файли у форматі .txt знаходяться за посиланням http://www.isi.csic.es/dataset/. Дані були конвертовані у формат .csv для подальшого аналізу. Колонки наведені на таблиці 3.1:

Таблиця 3.1 - Опис колонок набору даних

Колонка	Опис
Class	Тип трафіку: "Normal" або "Anomalous", що позначає нормальний чи аномальний (шкідливий) трафік.
Method	HTTP метод запиту: "GET" або "POST".
host	Хост або сервер, на який направлений запит, наприклад, "localhost:8080".
cookie	Значення cookie, які передаються у запиті.
Accept	Типи контенту, які може приймати клієнт, наприклад, "text/xml, application/xml, application/xhtml+xml".
content_length	Довжина контенту в запиті (для POST запитів).
content	Тіло запиту (передається в POST запитах).

Кінець таблиці 3.1

classification	Классифікація запиту: "0" для нормального трафіку, "1" для аномального (шкідливого) трафіку.
URL	Повний URL запиту, наприклад, "http://localhost:8080/tienda1/index.jsp HTTP/1.1".

Ці дані будуть використані для навчання та тестування моделей машинного навчання, що дозволить розробити систему для ефективного виявлення веб-загроз. Наступні етапи включають попередню обробку даних, їх очищення та підготовку до моделювання.

3.2.2 Методи попередньої обробки даних

У цьому підрозділі описані методи попередньої обробки даних, які використовувались для підготовки набору даних до моделювання. Основна мета попередньої обробки – очистити та трансформувати дані таким чином, щоб вони були придатними для навчання моделей машинного навчання. Методи що були застосовані описані в таблиці 3.2:

Таблиця 3.2 - Методи попередньої обробки даних

Метод	Опис
Підрахунок появ URL	Визначення кількості появ кожного URL та отримання топ-10 найпоширеніших URL.
Підрахунок крапок	Підрахунок кількості крапок у URL.
Підрахунок директорій	Підрахунок кількості директорій у URL.
Підрахунок вбудованих URL	Підрахунок кількості вбудованих URL.

Кінець таблиці 3.2

Перевірка скорочених URL	Перевірка, чи використовує URL сервіс скорочення (наприклад, bit.ly, goo.gl).
Підрахунок 'http'	Підрахунок появ "http" у URL.
Підрахунок відсотків	Підрахунок знаків відсотків (%) у URL.
Підрахунок питальних знаків	Підрахунок знаків питання (?) у URL.
Підрахунок дефісів	Підрахунок дефісів (-) у URL.
Підрахунок знаків рівності	Підрахунок знаків рівності (=) у URL.
Довжина URL	Визначення довжини URL.
Довжина хосту	Визначення довжини хосту у URL.
Підрахунок підозрілих слів	Підрахунок підозрілих слів у URL за допомогою спеціального словника.
Підрахунок цифр	Підрахунок кількості цифр у URL.
Підрахунок літер	Підрахунок кількості літер у URL.
Підрахунок спеціальних символів	Підрахунок спеціальних символів у URL.
Кількість параметрів	Підрахунок кількості параметрів у URL.
Кількість фрагментів	Підрахунок кількості фрагментів у URL.
Перевірка на кодування	Перевірка, чи є URL закодованим (%).
Співвідношення незвичних символів	Розрахунок співвідношення незвичних символів у URL.

Ці методи були обрані та використані для створення додаткових характеристик (фіч) URL, що дозволяє моделям машинного навчання більш

ефективно виявляти веб-загрози. Кожен метод допомагає виокремити різні аспекти структури та вмісту URL, що може бути індикатором шкідливої активності.

Частина результатів препроцесінгу наведена на Рисунку 3.1:

count_dot_url	...	hostname_length_url	sus_url	count-digits_url	count-letters_url	url_length	number_of_parameters_url	number_of_fragments_url	is_encoded_url	special_count_url
2	...	14	10	7	31	48	0	0	0	9
2	...	14	115	15	86	126	5	0	1	24
2	...	14	40	7	39	57	0	0	0	10
2	...	14	40	11	92	125	5	0	1	21
2	...	14	10	7	43	61	0	0	0	10
...	...	...	...	...	...	...	...	...	...	...
3	...	14	80	47	218	314	13	0	1	48
2	...	14	10	7	40	58	0	0	0	10
3	...	14	10	7	43	62	0	0	0	11
2	...	14	0	8	34	54	0	0	0	11
3	...	14	0	7	50	69	0	0	0	11

Рисунок 3.1 - Частина результатів препроцесінгу

3.2.3 Балансування даних

Однією з важливих задач підготовки даних є забезпечення збалансованості класів у навчальному наборі даних. У нашому випадку, набір даних "CSIC 2010 Web Application Attacks" містить дисбаланс між кількістю нормальних та аномальних запитів (Рис 3.2).

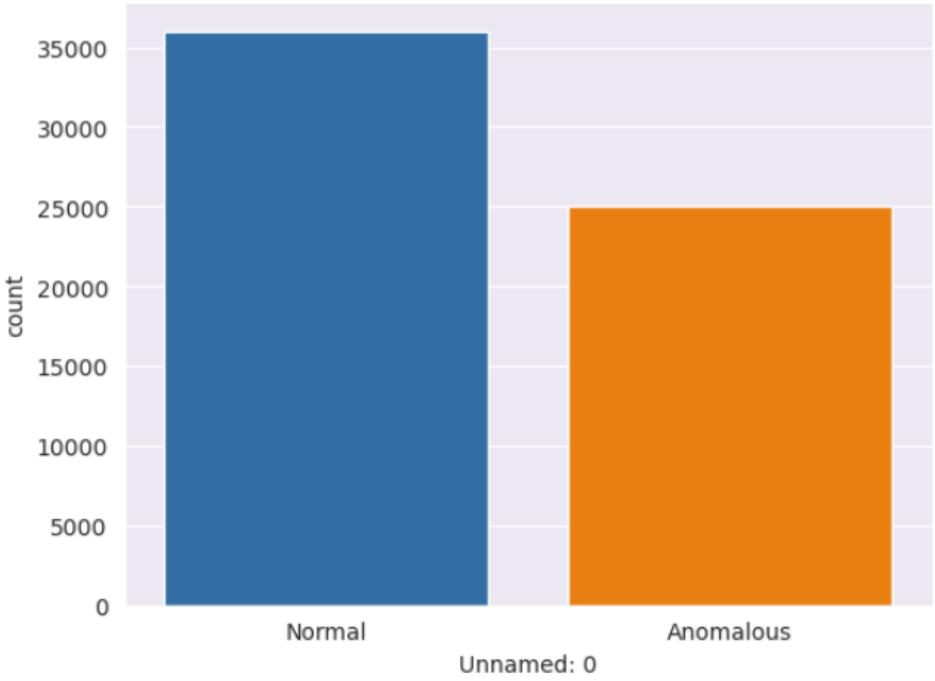

Рисунок 3.2 - Збалансування класів

Як видно з графіка на рисунку 3.2, кількість нормальних запитів перевищує кількість аномальних.

Для досягнення збалансованості ми використовували метод random under sampling. Цей метод зменшує кількість зразків з надмірно представленого класу (у нашому випадку, нормальних запитів), шляхом випадкового видалення зразків до тих пір, поки кількість зразків у кожному класі не стане рівною. Таким чином, ми отримуємо збалансований набір даних, який дозволяє моделям машинного навчання більш ефективно навчатися та уникати переваги одного класу над іншим.

Після застосування random under sampling, кількість зразків у кожному класі стала рівною, що значно покращило якість моделей при подальшому навчанні та тестуванні.

3.3 Вибір та налаштування моделей машинного навчання

Для розробки ефективної системи виявлення веб-загроз було обрано та налаштовано декілька моделей машинного навчання. Вибір моделей базувався на їхній здатності до класифікації та виявлення аномалій у даних. Було використано наступні моделі: RandomForestClassifier, GradientBoostingClassifier, AdaBoostClassifier, LogisticRegression, KNeighborsClassifier, DecisionTreeClassifier та GaussianNB.

Кожна з моделей має свої переваги та недоліки. RandomForestClassifier відомий своєю стійкістю до перенавчання та здатністю обробляти велику кількість вхідних змінних. GradientBoostingClassifier та AdaBoostClassifier є потужними ансамблевими методами, які можуть комбінувати слабкі моделі для покращення загальної точності. LogisticRegression добре підходить для випадків, де важливо інтерпретувати коефіцієнти регресії. KNeighborsClassifier базується на підході до найближчих сусідів, що дозволяє легко зрозуміти класифікацію нових даних. DecisionTreeClassifier пропонує простоту інтерпретації та візуалізації, а GaussianNB є ефективним для високошвидкісного навчання і класифікації.

Моделі були налаштовані з використанням параметрів за замовчуванням, з деякими модифікаціями, такими як встановлення фіксованого випадкового стану для забезпечення відтворюваності результатів. Кожна модель була оцінена за допомогою відповідних метрик, що дозволило визначити їх ефективність та обрати найкращі підходи для нашої системи виявлення веб-загроз.

3.4 Валідація та тестування моделей

Після налаштування моделей машинного навчання була проведена їх валідація та тестування на збалансованих і незбалансованих даних. Результати тестування включають метрики, такі як середня абсолютна помилка (MAE), точність (Accuracy), точність (Precision), повнота (Recall), F1-міра, ROC AUC та

тестова помилка. У таблиці 3.3 наведені результати для кожної моделі на збалансованому датасеті:

Таблиця 3.3 - Результати роботи моделей на збалансованих даних

Модель	MAE	Accuracy	Precision	Recall	F1	ROC AUC	Test Error
RandomForest	0.079	0.921	0.923	0.921	0.921	0.923	7.9%
GradientBoosting	0.138	0.862	0.877	0.862	0.863	0.873	13.8%
AdaBoost	0.192	0.808	0.811	0.808	0.809	0.807	19.2%
LogisticRegression	0.238	0.762	0.763	0.762	0.763	0.756	23.8%
KNeighbors	0.085	0.915	0.915	0.915	0.915	0.912	8.5%
DecisionTree	0.085	0.915	0.915	0.915	0.915	0.913	8.5%
GaussianNB	0.302	0.698	0.704	0.698	0.678	0.658	30.2%

Також було проведено тестування моделей на незбалансованому датасеті, результати наведені в таблиці 3.4:

Таблиця 3.4 - Результати роботи моделей на незбалансованих даних

Модель	MAE	Accuracy	Precision	Recall	F1	ROC AUC	Test Error
RandomForest	0.074	0.926	0.926	0.926	0.926	0.923	7.4%

Кінець таблиці 3.4

GradientBoosting	0.122	0.878	0.882	0.878	0.879	0.881	12.2%
AdaBoost	0.191	0.809	0.808	0.809	0.809	0.801	19.1%
LogisticRegression	0.213	0.787	0.788	0.787	0.783	0.767	21.3%
KNeighbors	0.084	0.916	0.916	0.916	0.916	0.913	8.4%
DecisionTree	0.083	0.917	0.917	0.917	0.916	0.912	8.3%
GaussianNB	0.302	0.698	0.704	0.698	0.678	0.658	30.2%

Ці результати показують, що деякі моделі демонструють кращу продуктивність на збалансованих даних у порівнянні з незбалансованими. Найкращі результати серед моделей показали RandomForest, KNeighbors та DecisionTree, які мали високі показники точності та низькі тестові помилки. Для цих трьох моделей були побудовані матриці суперечностей (confusion matrix) на рисунках 3.3-5:

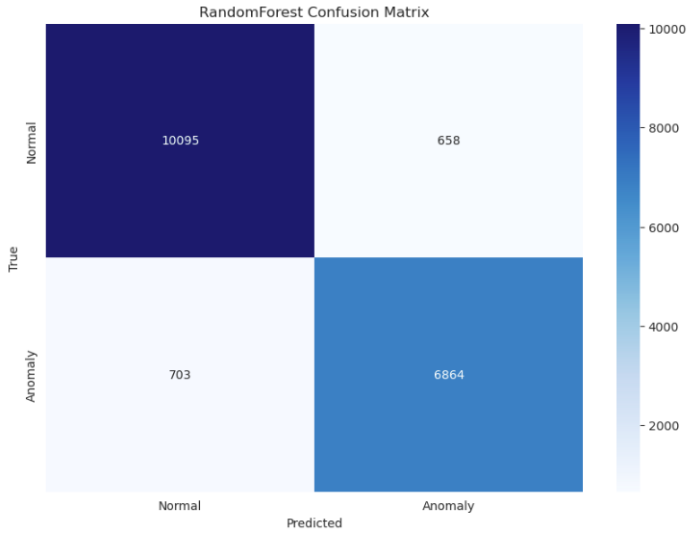

Рисунок 3.3 - Матриця суперечностей для моделі Random Forest

На матриці невідповідностей для RandomForest ми можемо побачити, що модель має високу точність у класифікації нормальних запитів (10095 правильно класифікованих нормальних запитів з 10753) і аномальних запитів (6864 правильно класифікованих аномальних запитів з 7567). Кількість помилково класифікованих нормальних запитів як аномальні становить 658, а помилково класифікованих аномальних запитів як нормальні - 703. Загалом, модель демонструє хороші результати з високою точністю, що підтверджується метриками точності та повноти.

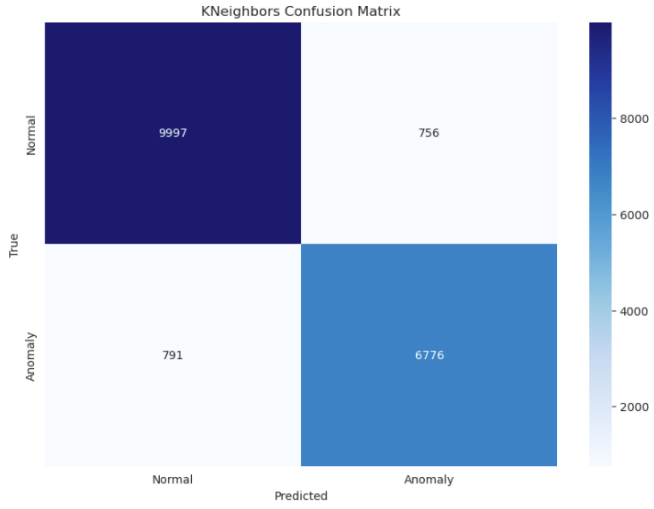

Рисунок 3.4 - Матриця суперечностей для моделі KNeighbors

Для KNeighbors модель також демонструє високу точність, хоча і трохи нижчу, ніж RandomForest. На матриці видно, що модель правильно класифікувала 9997 нормальних запитів з 10753 і 6776 аномальних запитів з 7567. Помилково класифікованих нормальних запитів як аномальні становить 756, а аномальних як нормальні - 791. Хоча кількість помилок трохи більша, ніж у RandomForest, загальна продуктивність все ще є високою і модель добре справляється з класифікацією.

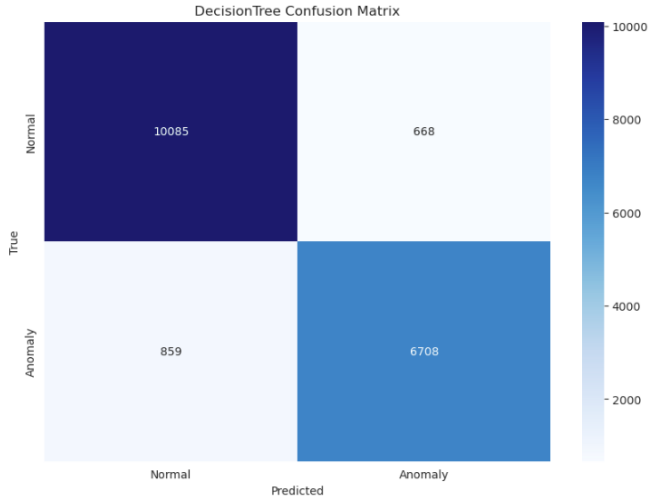

Рисунок 3.5 - Матриця суперечностей для моделі DecisionTree

На матриці невідповідностей для DecisionTree видно, що модель має хороші результати в класифікації нормальних запитів (10085 правильно класифікованих з 10753) та аномальних запитів (6708 правильно класифікованих з 7567). Помилково класифікованих нормальних запитів як аномальні становить 668, а аномальних як нормальні - 859. Ця модель має дещо більшу кількість помилок у порівнянні з RandomForest та KNeighbors, але загальна продуктивність залишається на високому рівні.

Всі три моделі - RandomForest, KNeighbors та DecisionTree - демонструють хороші результати в класифікації нормальних і аномальних запитів. RandomForest

показує найкращі результати з найменшою кількістю помилок, тоді як KNeighbors та DecisionTree також демонструють високу точність, але з дещо більшою кількістю помилково класифікованих запитів. Ці результати свідчать про те, що обрані моделі є ефективними для завдання виявлення веб-загроз.

3.5 Дослідження важливості показників для передбачень

Для подальшого аналізу та інтерпретації результатів моделі ми використали метод логістичної регресії для оцінки важливості різних показників (фіч) у передбаченнях. Логістична регресія була натренована із наступними метриками:

1. MAE: 0.0894
2. MSE: 0.0501
3. R2 Score: 0.7932
4. Точність (Accuracy): 0.9200

Результати показують високу точність моделі, що свідчить про її здатність правильно класифікувати більшість запитів. Після тренування моделі ми візуалізували важливість кожної з колонок (фіч) для передбачення (Рис. 3.6):

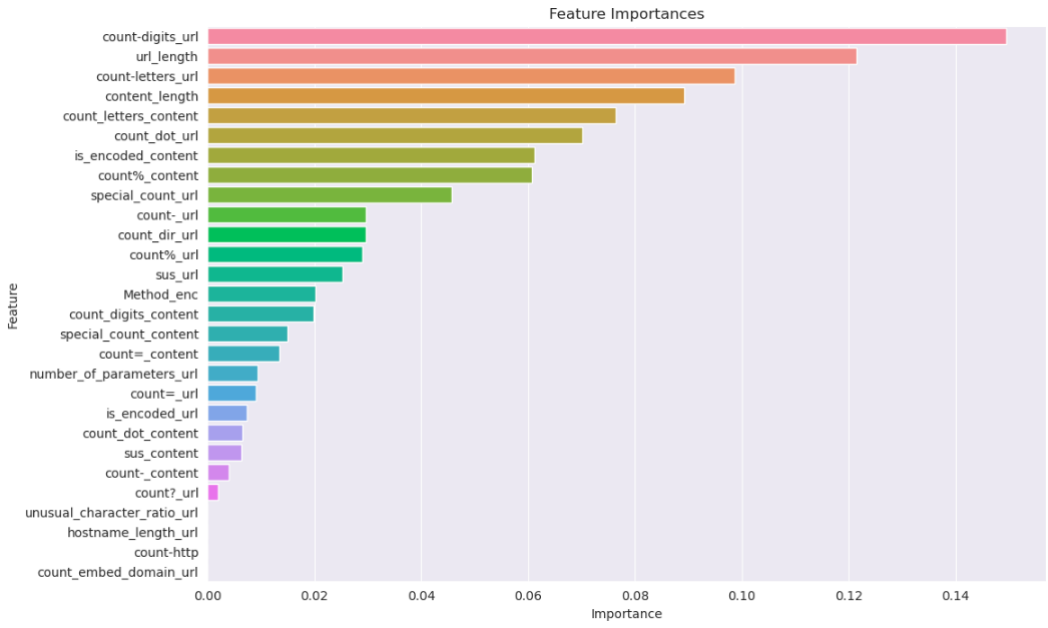

Рисунок 3.6 - Важливість показників для моделі Logistic Regression

На графіку можна побачити, які фічі найбільше впливають на результат моделі.

Серед найбільш важливих показників можна виділити:

1. Кількість цифр у URL (count-digits_url)
2. Довжина URL (url_length)
3. Кількість літер у URL (count-letters_url)
4. Довжина контенту (content_length)
5. Кількість літер у контенті (count_letters_content)

Ці фічі мають найбільший вплив на передбачення, що свідчить про їх значущість у визначенні того, чи є запит нормальним або аномальним. Інші важливі фічі включають кількість крапок у URL (count_dot_url), факт кодування контенту (is_encoded_content) та інші специфічні характеристики URL і контенту.

Такий аналіз важливості фіч дозволяє краще зрозуміти, які параметри мають найбільший вплив на класифікацію запитів та може бути використаний для подальшого покращення моделі та її інтерпретації.

Висновки до розідлу 3

У розділі 3 було представлено процес розробки системи виявлення веб-загроз за допомогою методів машинного навчання. Поставлена задача вимагала створення багатокомпонентної системи, здатної ефективно виявляти різноманітні типи веб-загроз, такі як фішинг, шкідливі програми, SQL-ін'єкції та інші атаки.

Спочатку було здійснено збір та підготовку даних, використовуючи набір даних "CSIC 2010 Web Application Attacks". Далі ми застосували різноманітні методи попередньої обробки даних, які допомогли створити додаткові характеристики для моделей машинного навчання. Балансування даних було

досягнуто за допомогою методу random under sampling, що забезпечило рівну кількість нормальних та аномальних запитів у навчальному наборі даних.

Для вибору та налаштування моделей машинного навчання було обрано декілька алгоритмів, включаючи RandomForestClassifier, GradientBoostingClassifier, AdaBoostClassifier, LogisticRegression, KNeighborsClassifier, DecisionTreeClassifier та GaussianNB. Кожна модель була налаштована і оцінена за допомогою відповідних метрик.

Результати тестування моделей показали, що деякі моделі демонструють кращу продуктивність на збалансованих даних у порівнянні з незбалансованими. Найкращі результати серед моделей показали RandomForest, KNeighbors та DecisionTree, які мали високі показники точності та низькі тестові помилки.

Дослідження важливості показників для передбачень здійснювалось за допомогою логістичної регресії. Результати показали, що такі фічі як кількість цифр у URL, довжина URL, кількість літер у URL, довжина контенту та кількість літер у контенті мають найбільший вплив на передбачення, що свідчить про їх значущість у визначенні того, чи є запит нормальним або аномальним.

Загалом, розроблена система виявлення веб-загроз продемонструвала високу ефективність, що підтверджується результатами тестування моделей і дослідженням важливості показників. Ці результати можуть бути використані для подальшого покращення моделей та інтерпретації їх роботи, що сприятиме підвищенню рівня безпеки веб-додатків та користувачів.

ВИСНОВКИ

Дослідження на тему "Виявлення веб-загроз за допомогою машинного навчання" надає глибокий аналіз різних методів машинного навчання та їх застосування в сфері кібербезпеки. Основною метою цієї роботи було розробити надійну систему, здатну виявляти веб-загрози через аналіз мережевого трафіку. Проєкт успішно досягнув цієї мети, використовуючи кілька моделей машинного навчання, які були навчені та оцінені на реальних наборах даних.

Спочатку дослідження було зосереджене на розумінні теоретичних основ веб-загроз та потенціалу машинного навчання для їх виявлення. Було розглянуто ключові аспекти класифікації, кластеризації, аномалійного виявлення та глибокого навчання, а також їх роль у кібербезпеці. У розділі 1.1 були описані основні поняття та термінологія, що використовуються в цій сфері, що створило базу для подальшого дослідження.

Під час реалізації проєкту була проведена ретельна підготовка даних, включаючи очищення та обробку набору даних CSIC-2010, що містить як нормальні, так і аномальні запити до веб-додатків. Це дозволило створити ефективні моделі для виявлення загроз, зокрема, були використані методи класифікації, такі як Random Forest, Gradient Boosting, AdaBoost, Logistic Regression, K-Neighbors, Decision Tree та Gaussian Naive Bayes. Кожна з цих моделей була оцінена за допомогою таких метрик, як точність, повнота, F1-міра та ROC AUC.

Особлива увага була приділена збалансуванню набору даних, що дозволило покращити ефективність моделей, особливо при виявленні аномальних запитів. Метод Random Under Sampling був використаний для створення збалансованих навчальних даних, що призвело до покращення результатів класифікації.

У ході дослідження було виявлено, що методи глибокого навчання, такі як нейронні мережі, можуть бути ефективними для виявлення складних патернів у мережевому трафіку. Це відкриває нові перспективи для розвитку систем виявлення загроз, здатних адаптуватися до нових типів атак.

На основі отриманих результатів були надані рекомендації щодо подальшого вдосконалення систем виявлення загроз. Зокрема, пропонується розробити інтегровані рішення, що поєднують різні методи машинного навчання та забезпечують більш високий рівень захисту від веб-загроз.

У підсумку, проведене дослідження показало, що застосування машинного навчання для виявлення веб-загроз є перспективним напрямом, який може значно підвищити ефективність кібербезпеки. Подальші дослідження та розробки у цій сфері можуть призвести до створення більш надійних та адаптивних систем захисту, здатних вчасно виявляти та нейтралізувати загрози.

Теми для подальших досліджень включають вдосконалення алгоритмів машинного навчання для виявлення нових типів атак, розробку методів для автоматизованого оновлення моделей на основі нових даних, а також інтеграцію систем виявлення загроз з іншими інструментами кібербезпеки для забезпечення комплексного захисту інформаційних систем.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. R G Gayathri, Atul Sajjanhar, Yong Xiang, Xingjun Ma. Anomaly Detection for Scenario-based Insider Activities using CGAN Augmented Data. ArXiv. URL: https://arxiv.org/abs/2102.07277 (date of access: 02.06.2024).
2. Aurélien Géron,Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition. 2019
3. Hadiseh Safdari, Martina Contisciani, Caterina De Bacco. Anomaly, reciprocity, and community detection in networks. ArXiv. URL: https://arxiv.org/abs/2302.00504 (date of access: 02.06.2024).
4. Olha Jurečková, Martin Jureček, Mark Stamp. Online Clustering of Known and Emerging Malware Families. ArXiv. URL: https://arxiv.org/abs/2405.03298 (date of access: 02.06.2024).
5. Marwan Omar. Machine Learning for Cybersecurity: Innovative Deep Learning Solutions 2022
6. Ravi Ranjan, G. Sahoo. A New Clustering Approach for Anomaly Intrusion Detection. ArXiv. URL: https://arxiv.org/abs/1404.2772 (date of access: 02.06.2024).
7. Abenezer Girma, Xuyang Yan, Abdollah Homaifar. Driver Identification Based on Vehicle Telematics Data using LSTM-Recurrent Neural Network. ArXiv. URL: https://arxiv.org/abs/1911.08030 (date of access: 02.06.2024).
8. Lukas Machlica, Karel Bartos, Michal Sofka. Learning detectors of malicious web requests for intrusion detection in network traffic. ArXiv. URL: https://arxiv.org/abs/1702.02530 (date of access: 02.06.2024).
9. Sanjay Chakraborty, Saroj Kumar Pandey, Saikat Maity, Lopamudra Dey. Detection and Classification of Novel Attacks and Anomaly in IoT Network using Rule based Deep Learning Model. ArXiv. URL: https://arxiv.org/abs/2308.00005 (date of access: 02.06.2024).

10. Emmanuel Tsukerman. Machine Learning for Cybersecurity Cookbook: Over 80 recipes on how to implement machine learning algorithms for building security systems using Python 2019

11. Dmitry Lande, Leonard Strashnoy and Irina Balagura. Directed correlation network of concepts determined by the dynamics of publications. Available at SSRN: http://ssrn.com/abstract=3688659, DOI: https://dx.doi.org/10.2139/ssrn.3688659 (Posted: 8 Sep 2020). - 12 p.

12. Alan Nafiiev, Dmytro Lande. Malware detection model based on machine learning. Bulletin of Cherkasy State Technological University, 2023. Iss. 3. - pp. 40-50. DOI: 0.24025/2306-4412.3.2023.286374