Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
Факультет інформаційних технологій
Кафедра безпеки інформації та телекомунікацій

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеню бакалавра

студента *Шалагінова Іллі Олеговича*

академічної групи *125-20-1*

спеціальності *125 Кібербезпека*

спеціалізації[1]

за освітньо-професійною програмою *Кібербезпека*

на тему *Застосування технологій штучного інтелекту для виявлення та нейтралізації вебатак*

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	доц. Ткач М.О.			
розділів:				
спеціальний	доц. Ткач М.О.			
економічний	к.е.н., доц. Пілова Д.П.			
Рецензент				
Нормоконтролер	ст. викл. Мєшков В.І.			

Дніпро
2024

ЗАТВЕРДЖЕНО:
завідувач кафедри
безпеки інформації та телекомунікацій
_____ д.т.н., проф. Корнієнко В.І.

«_____»_____20___року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня бакалавра

студенту ***Шалагінову Іллі Олеговичу*** **академічної групи** ***125-20-1***
(прізвище ім'я по-батькові) (шифр)

спеціальності *125 Кібербезпека*
(код і назва спеціальності)

на тему ***Застосування технологій штучного інтелекту для виявлення та нейтралізації вебатак***

затверджену наказом ректора НТУ «Дніпровська політехніка» від 23.05.2024 № 469-с

Розділ	Зміст	Термін виконання
Розділ 1	Аналіз вебатак, існуючих засобів протидії вебатакам, технологій штучного інтелекту для виявлення та нейтралізації вебатак. Постановка задачі.	15.03.2024
Розділ 2	Розробка підходів для виявлення та нейтралізації вебатак за допомогою технологій штучного інтелекту.	10.05.2024
Розділ 3	Розрахунок економічної доцільності застосування технологій штучного інтелекту для виявлення та нейтралізації вебатак.	11.06.2024

Завдання видано _____ Максим ТКАЧ
(підпис керівника) (ім'я, прізвище)

Дата видачі: 15.01.2024р.

Дата подання до екзаменаційної комісії: 28.06.2024р.

Прийнято до виконання _____ Ілля ШАЛАГІНОВ
(підпис студента) (ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 69 с., 15 рис., 4 додатка 24 джерела.

Об’єкт розробки: технології штучного інтелекту для виявлення та нейтралізації вебатак.

Предмет розробки: засоби застосування технологій штучного інтелекту для виявлення та нейтралізації вебатак.

Мета роботи: підвищення рівня захисту інформаційної системи за рахунок впровадження технологій штучного інтелекту для виявлення та нейтралізації вебатак.

У першому розділі розглянуто класифікацію, методи здійснення вебатак, їхню природу, механізми дії та вплив на безпеку інформаційних систем. Аналізуючи атаки, було визначено їх типи та механізми, а також способи впливу на системи та потенційні небезпеки.

У другому розділі досліджується використання штучного інтелекту для виявлення та нейтралізації вебатак. Показано, як ШІ допомагає системам адаптуватися до нових загроз та виявляти відомі атаки через самонавчання.

В економічному аспекті дослідження здійснено оцінку ефективності розробки та впровадження технологій штучного інтелекту для боротьби з вебатаками. Здійснено аналіз капітальних та експлуатаційних витрат, а також оцінку загального збитку від можливих атак на інформаційну та технологічну систему. Крім того, проведено розрахунок загального ефекту від застосування рекомендацій, спрямованих на підвищення безпеки та стійкості інформаційної інфраструктури організації.

ШТУЧНИЙ ІНТЕЛЕКТ, ВЕБАТАКИ, КІБЕРБЕЗПЕКА, МАШИННЕ НАВЧАННЯ, НЕЙРОННІ МЕРЕЖІ, ЗАХИСТ ІНФОРМАЦІЇ, ЗАПОБІГАННЯ АТАКАМ.

ABSTRACT

Explanatory note: 69 pp., 15 pic., 4 app, 24 sources.

Object of Development: artificial intelligence technologies for detecting and neutralizing web attacks.

Subject of Development: means of applying artificial intelligence technology for detecting and neutralizing web attacks.

Objective of the Work: enhancing the security level of the information system by implementing artificial intelligence technologies for detecting and neutralizing web attacks.

The first chapter examines the classification, methods of execution of DDoS and web attacks, their nature, mechanisms of action, and impact on the security of information systems. Analyzing the attacks, their types, mechanisms, as well as methods of influencing systems and potential dangers were determined.

The second chapter explores the use of artificial intelligence for detecting and neutralizing web attacks. It demonstrates how AI helps systems adapt to new threats and identify known attacks through self-learning.

In the economic aspect of the study, an assessment of the effectiveness of developing and implementing artificial intelligence technologies to combat DDoS and web attacks is carried out. An analysis of capital and operational costs, as well as an evaluation of the overall damage from potential attacks on the information and technological system, are conducted. Additionally, a calculation of the overall effect of implementing recommendations aimed at enhancing the security and resilience of the organization's information infrastructure is performed.

ARTIFICIAL INTELLIGENCE, WEB ATTACKS, CYBERSECURITY, MACHINE LEARNING, NEURAL NETWORKS, INFORMATION SECURITY, ATTACK PREVENTION.

СПИСОК УМОВНИХ СКОРОЧЕНЬ

AI – Artificial Intelligence;

AWS – Amazon Web Services;

BERT – Bidirectional Encoder Representations from Transformers;

CSRF – Cross-Site Request Forgery;

DBSCAN – Density-Based Spatial Clustering of Applications with Noise;

DDoS – Distributed Denial of Service;

DNS – Domain Name System;

GPT – Generative Pre-trained Transformer;

HTML – Hypertext Markup Language;

HTTP/HTTPS – Hypertext Transfer Protocol/Hypertext Transfer Protocol Secure;

IBM – International Business Machines;

IDPS – Intrusion Detection and Prevention System;

IDS/IPS – Intrusion Detection System/Intrusion Prevention System;

IP – Internet Protocol;

ML – Machine Learning;

NLG – Natural Language Generation;

NLP – Natural Language Processing;

OWASP – Open Web Application Security Project;

RF – Random Forest;

RL – Reinforcement Learning;

SL – Supervised Learning;

SQL – Structured Query Language;

SQLI – Structured Query Language Injection;

SSL – Secure Sockets Layer;

SVM – Support Vector Machine;

TLS – Transport Layer Security;

UL – Unsupervised Learning;

URL – Uniform Resource Locator;

WAF – Web Application Firewall;

XSS – Cross-Site Scripting;

ШІ – штучний інтелект.

ЗМІСТ

с.

ВСТУП

Використання мережі «Інтернет», та, зокрема, вебдодатків, останнім часом зростає з великою швидкістю. Пандемія коронавірусу лише прискорила цей процес, мотивуючи бізнеси переносити свої послуги на власні вебресурси, надаючи більш зручний спосіб надання послуг. Крім того, різноманітні гаджети давно увійшли у повсякдення, полегшуючи доступ до інформації. Число смартфонів у світі в 2024 році становило приблизно 7.2 млрд., що на 400 млн. більше ніж торік [9].

Однак зі зростанням популярності, вони все частіше стають ціллю для атак кіберзлочинців, що вигадують нові підходи та засоби, адаптація до яких потребує час та гроші.

Традиційні рішення добре працюють для протидії відомим атакам, але розвиток технологій робить системи на основі цих рішень неефективними проти нових типів атак і зловмисних програм. Наприклад, системи IDS/IPS можуть не виявляти нові види атак, оскільки вони покладаються на правила, які можуть бути застарілими Таким чином, є потреба в тому, щоб запровадити рішення, яке могло б динамічно та швидко адаптуватися під змінювані умови.

Тому, використання сучасних технологій штучного інтелекту, наприклад машинне навчання, або глибоке навчання, може стати надійним рішенням для потенційних кібератак. Ці рішення можуть допомогти у виявленні зловмисного програмного забезпечення, запобіганні вторгнень, ідентифікації спаму, класифікації DNS-атак, і розрізненні розширених загроз. Машинне навчання може використовуватися для створення моделей, що автоматично розпізнають аномалії та підозрілу активність, тоді як глибокі нейронні мережі здатні виявляти складні патерни, які не помітні для традиційних методів аналізу.

Об'єктом розробки є технології штучного інтелекту, які можуть бути використані для виявлення та нейтралізації вебатак.

Предметом розробки є засоби застосування технологій штучного інтелекту для ефективного виявлення та нейтралізації вебатак.

Метою роботи є підвищення рівня захисту інформаційної системи за рахунок впровадження технологій штучного інтелекту для виявлення та нейтралізації вебатак.

Завдання роботи включають:

1. Проаналізувати характеристики вебатак, а також традиційні методи їх виявлення та протидії;
2. Розглянути принципи роботи штучного інтелекту;
3. Проаналізувати сучасні методи на основі ШІ для виявлення вебатак;
4. Дослідити засоби нейтралізації вебатак на основі ШІ;

Практичне значення роботи полягає у розробці нових засобів, що дозволять ефективно захищати інформаційні системи від високоскладних кіберзагроз. Результати дослідження можуть бути використані в розробці програмного забезпечення та в сфері інформаційної безпеки для забезпечення стабільної роботи та захисту конфіденційної інформації.

РОЗДІЛ 1. СТАН ПИТАННЯ. ПОСТАНОВКА ЗАДАЧІ

1.1 Класифікація вебатак

1. SQLI – це тип атаки, що розрахований не уразливості в безпеці баз даних, коли в поле вводу вставляються шкідливі SQL-інструкції для виконання, що часто призводить до несанкціонованого доступу і витоку даних [5].

Цей тип атаки не є новою загрозою, та незважаючи на це, вона стабільно займає одне з перших місць серед загроз безпеці вебдодатків.

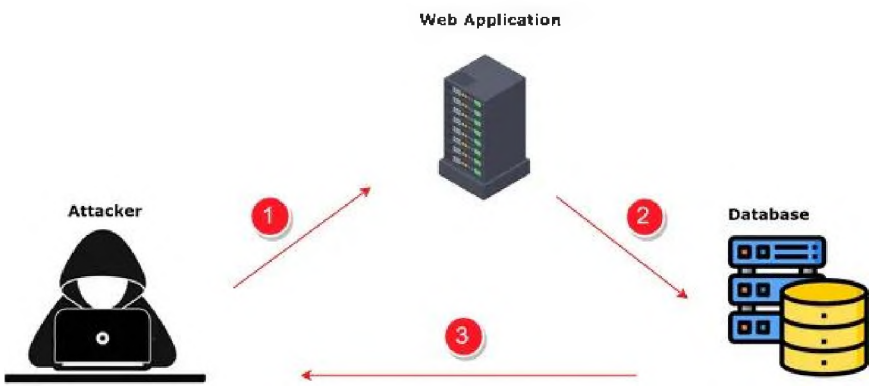

Рисунок 1.1 – Схема роботи SQLI [5]

Атакуючий надсилає серверу завантаження даних і спостерігає за відповіддю та поведінкою сервера, щоб дізнатися більше про його структуру. Цей метод називається сліпою SQL-ін'єкцією, оскільки дані не передаються з бази даних вебсайту до атакуючого, тому він не може бачити інформацію про атаку безпосередньо.

За даними OWASP, SQLI залишається однією з 10 основних вразливостей вебдодатків, часто призводячи до катастрофічних порушень, якщо вчасно не вжити превентивних заходів [8].

На рис. 1.2 зображено основні типи загроз для додатків, де видно зміни в їх пріоритетності з 2017 року. SQLI, яка раніше займала перше місце серед загроз, поступово втратила свої позиції, опустившись на третє місце. Це свідчить

про вдосконалення захисту від SQL-ін'єкцій, але також про зростаючий рівень інших атак.

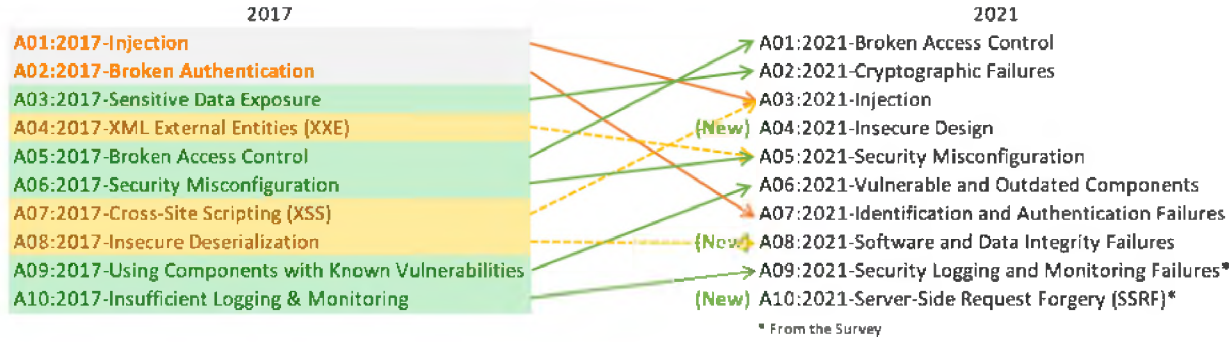

Рисунок 1.2 – Схема розповсюдженості вебатак [8]

Приклади найвідоміших атак типу SQLI:

– вразливість Cisco: у 2018 році було виявлено уразливість типу SQLI у Cisco Prime License Manager, що дозволяло зловмисникам отримати доступ до системи, де було розгорнуто цей менеджер ліцензій [15].

– вразливість Fortnite: у популярній онлайн-грі Fortnite, яка має понад 350 мільйонів користувачів, у 2019 році було знайдено уразливість типу SQLI injection. Ця уразливість дозволяла зловмисникам отримати доступ до облікових записів користувачів [15].

Щоб показати, як працює даний тип атаки можна взяти за приклад систему аутентифікації, яка використовує таблицю бази даних з іменами користувачів та паролями. Запит користувача надає змінні дані, які вставляються у SQL-запит:

– SELECT id FROM users WHERE username="" + user + "" AND password="" + pass + "".

Вразливість полягає в тому, що в SQL-запиті використовується склеювання для поєднання даних. Зловмисник може надати такий рядок для паролю:

– password' OR 5=5.

Кінцевий SQL-запит буде виглядати наступним чином:

– SELECT id FROM users WHERE username='user' AND password='pass' OR 5=5'.

Оскільки умова 5=5 завжди оцінюється як істинна, оператор WHERE буде істинним, незалежно від наданого імені користувача або пароля.

2. Cross-site Scripting – це атака з ін'єкцією коду на стороні користувача. Зловмисник прагне виконати шкідливі скрипти у веббраузері жертви. Атака відбувається, коли жертва відвідує вебсторінку, які виконують цей шкідливий код. Тоді вебсайт стає засобом доставки шкідливого скрипту до браузера користувача. Найчастіше вразливі до XSS атаки форуми, дошки оголошень і вебсторінки, що дозволяють коментарі [16].

Для здійснення атаки вразливий вебсайт повинен мати можливість введення даних користувача на своїх сторінках. Зловмисник може вставити шкідливий код, який буде використовуватися у вебсторінці та оброблятися браузером жертви як вихідний код. Існують також підходи XSS-атак, де зловмисник змушує користувача відвідати спеціальний URL за допомогою соціальної інженерії.

Крім того, зловмисники часто використовують XSS для викрадення cookie, що дозволяє їм видавати себе за жертву. На рис. 1.3 показано процес захвату cookie-файлів користувача. Загалом, відбувається наступне:

– зловмисник вставляє шкідливий код у базу даних вебсайту через уразливу форму;

– жертва запитує вебсторінку у вебсервера;

– вебсервер відправляє сторінку з шкідливим кодом у складі HTML тіла;

– веббраузер жертви виконує шкідливий скрипт, який відправляє cookie жертви на сервер зловмисника;

– зловмисник отримує cookie жертви після приходу HTTP-запиту на сервер;

– використовуючи викрадені cookie, зловмисник може видавати себе за жертву.

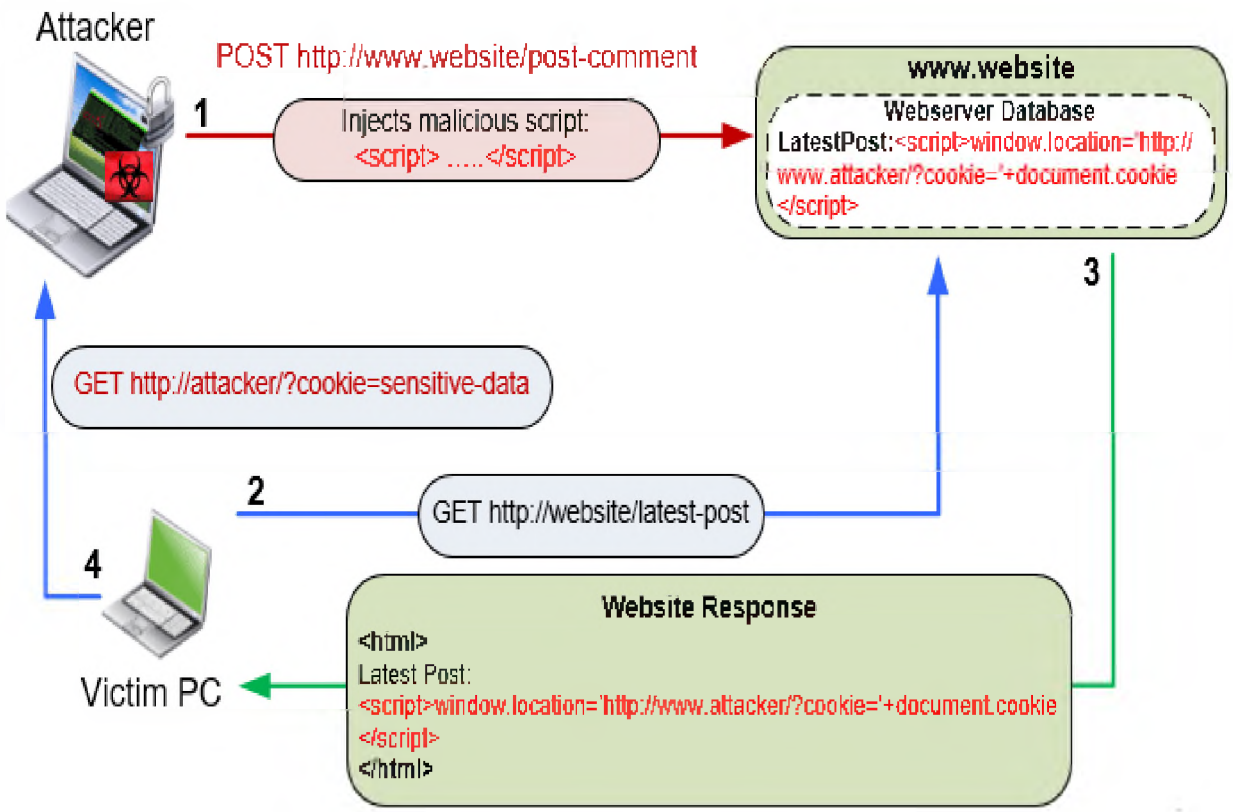

Рисунок 1.3 – Процес виконання XSS-атаки [16]

3. CSRF-атака відбувається, коли зловмисник змушує браузер жертви виконати небажані дії на вебсайті, до якого користувач уже автентифікований. Це досягається шляхом обману, щоб змусити відвідати спеціально створене шкідливе посилання або сторінку, що містить шкідливий вебскрипт. Потім браузер автоматично виконує запит від імені автентифікованого користувача. На рис. 1.4 показано, як виконується така атака.

CSRF-атака експлуатує довіру, яку вебсайт має до автентифікованого користувача. Зловмисник використовує цю довіру, щоб змусити браузер жертви виконати шкідливі дії на вебсайті. Оскільки запит надходить з браузера користувача, сервер вважає його легітимним і виконує його, не підозрюючи про шахрайство.

Ключовий аспект CSRF-атаки полягає в тому, що шкідливий запит виконується без відома і згоди користувача. Зловмисник створює шкідливе посилання або вебсторінку, яка містить шкідливий вебскрипт або форму. Коли жертва взаємодіє з цим контентом (наприклад, натискає на посилання або

відвідує вебсторінку), браузер автоматично відправляє запит до цільового вебсайту від імені автентифікованого користувача.

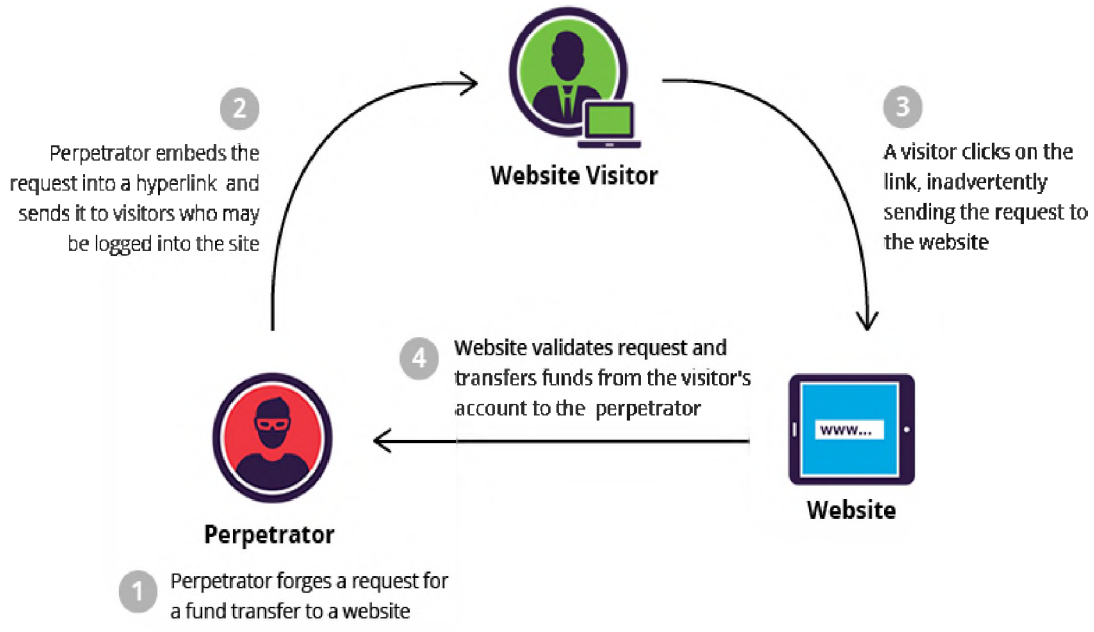

Рисунок 1.4 – Принцип дії Cross-Site Request Forgery [17]

Перш ніж здійснити атаку, зловмиснику зазвичай потрібно дослідити додаток, щоб створити підроблений запит, який виглядає як справжній.

Наприклад, типовий GET-запит для переказу 300 гривень може виглядати так:

– GET http://onlinebank.com/sendmoney?account=UserX&sum=$300 HTTP/1.1.

Хакер має можливість змінити цей сценарій, щоб це призвело до переказу 300 гривень на його власний рахунок. Тепер шкідливий запит буде виглядати так:

– GET http://onlinebank.com/sendmoney?account=AttackerУ&sum=$300 HTTP/1.1.

Зловмисник, користуючись мовою розмітки HTML, потенційно буде спроможній здійснити вставку запиту у звичайне на вигляд гіперпосилання.

```
<a href="http://onlinebank.com/sendmoney?account=AttackerY&sum=$300">Детальніше тут!</a>
```

Рисунок 1.5 – Приклад небезпечного гіперпосилання

Далі хакер може розіслати гіперпосилання електронною поштою великій кількості клієнтів банку. Ті, хто натисне на посилання, перебуваючи у своєму обліковому записі, ненавмисно ініціюють переказ 300 гривень.

Треба зазначити, що якщо на сайті банку використовується лише POST-запити, неможливо створити шкідливі запити за допомогою тегу <a>. Однак атаку можна реалізувати за допомогою тега <form> з автоматичним виконанням вбудованого JavaScript.

На рис. 1.6 показано, як може виглядати така форма.

```
<body onLoad='document.forms[0].submit()'>
  <form action='http://onlinebank.com/sendmoney' method='POST'>
    <input type='hidden' name='account' value='AttackerY' />
    <input type='hidden' name='sum' value='$300' />
    <input type='submit' value='Побачити більше!' />
  </form>
</body>
```

Рисунок 1.6 – Форма для відправки даних

4. Malware-Based атака ставить за ціль розповсюдження шкідливого ПЗ за допомогою вебзастосунків.

Одним з підходів даного типу атаки є Drive-by download, оскільки він може виконувати шкідливий код на комп'ютері користувача без його участі.

Атака Drive-by download відбувається, коли зловмисник вставляє шкідливий скрипт у вебсторінки, які відвідує користувач. Якщо атака пройде успішно, вбудований код змусить комп'ютер жертви завантажити та примусово встановити шкідливе ПЗ, надаючи атакуючому повний контроль над

комп'ютером жертви. Це дозволяє зловмиснику записувати натискання клавіш, викрадати облікові дані та конфіденційну інформацію [18].

На сьогоднішній день зловмисники, що здійснюють атаки Drive-by download, націлюються на плагіни браузера. Розробники часто нехтують безпекою плагінів порівняно з безпекою самого браузера, тому вони містять більше вразливостей.

Як можна побачити з рис. 1.7, у 2023 році було зафіксовано 6 мільярдів атак зловмисного програмного забезпечення по всьому світу.

Найпоширенішими типами шкідливих атак є хробаки, віруси, програми-вимагачі, трояни та бекдори [2]. З двох основних векторів атак - електронної пошти та вебсайтів – останні частіше використовувалися для фішингових атак.

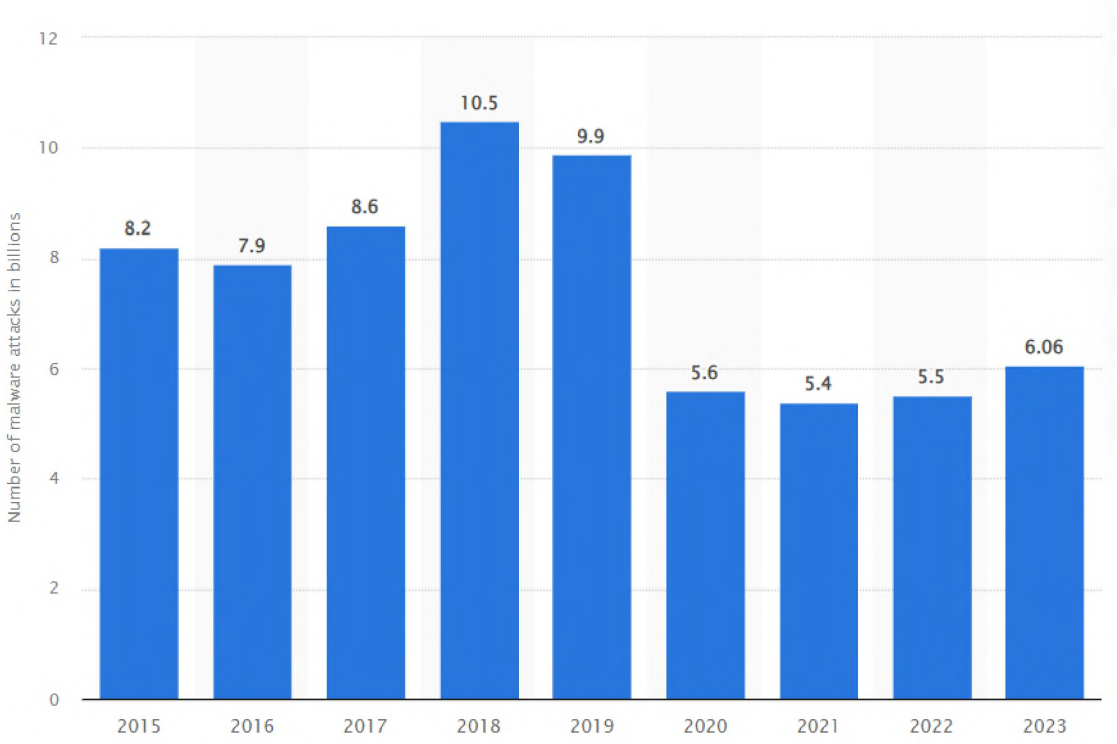

Рисунок 1.7 – Статистика атак з використанням шкідливого ПЗ [2]

1.2 Традиційні засоби захисту вебдодатків та їх обмеження

Відповідно до зростаючої загрози вебатак було розроблено різноманітні засоби захисту вебдодатків.

1. Web Application Firewall використовується для захисту вебсерверів та вебдодатків, виявляючи та запобігаючи атакам шляхом ідентифікації нормальної роботи запитів HTTP та HTTPS. Якщо пакет даних є безпечним, він пропускається через брандмауер, а в разі підозри на загрозу – блокується. Існують різні типи WAF, такі як хостові, мережеві та хмарні, залежно від платформи, на якій знаходиться вебдодаток.

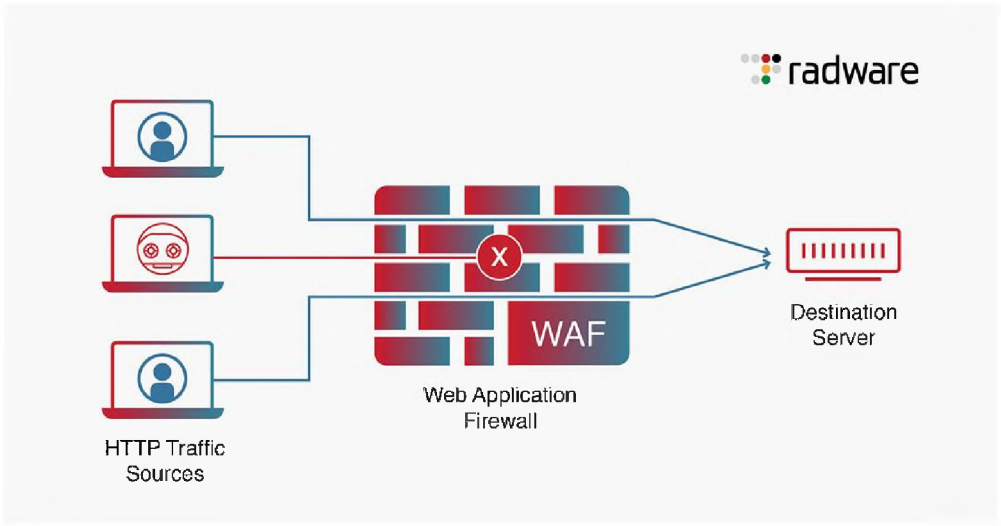

Рисунок 1.8 – Схема захисту WAF [6]

WAF може ефективно захищати від багатьох типів вебатак, проте цей підхід має низку обмежень:

– відсутність зручного інтерфейсу: у багатьох відкритих джерелах WAF не вистачає чітких і зручних для користувача інтерфейсів. Це ускладнює роботу з програмою і заважає знаходити вразливості;

– хибні спрацювання: WAF використовує заздалегідь визначені шаблони для визначення безпечності даних. Це призводить до хибних результатів, якщо є відхилення від шаблонів;

– вартість: зазвичай ціна впровадження WAF є високою, і невеликі бізнеси часто не можуть собі це дозволити;

– продуктивність: постійний аналіз мережі може призводити до низької швидкодії вебдодатків.

2. Системи виявлення та запобігання вторгненням здебільшого спрямовані на ідентифікацію можливих інцидентів. Наприклад, IDPS може виявити, коли зловмисник успішно взламав систему, використовуючи вразливість у ній. IDPS зберігає інформацію про активність і повідомляє про інцидент адміністраторам безпеки, щоб вони вжили заходів безпеки [19].

Також IDPS використовують для отримання розуміння загроз, які вони виявляють, їхньої частоти та особливостей.

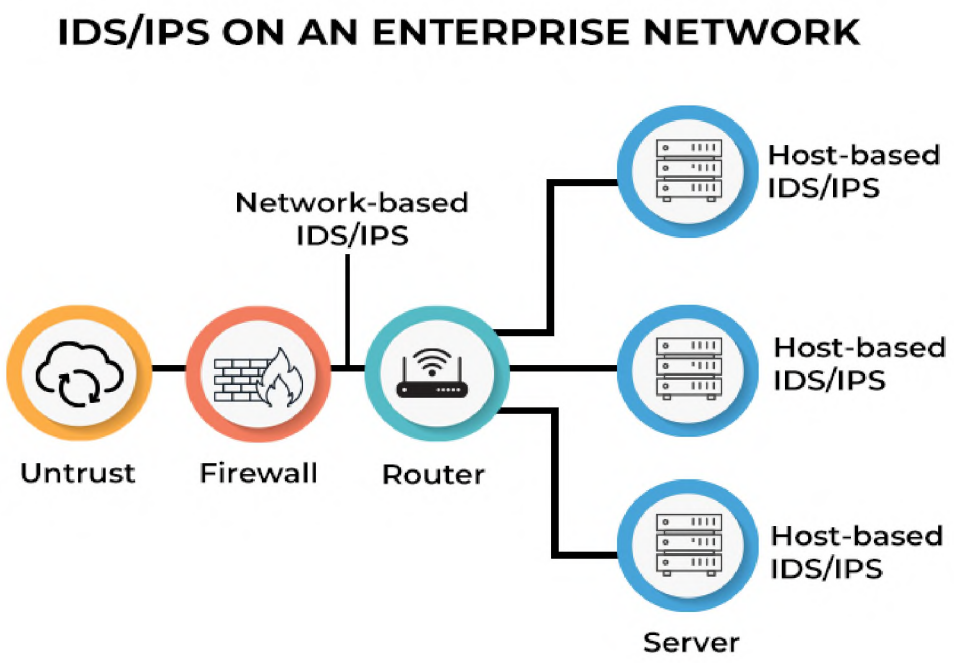

Рисунок 1.9 – Підхід IDS/IPS захисту [19]

3. Використання протоколів HTTPS та SSL/TLS забезпечить захист даних, що передаються через Інтернет. Це запобігає перегляду або зміні даних, що обмінюються між двома користувачами, зазвичай веббраузером та сервером.

Коли користувач відвідує вебсайт через HTTPS, браузер надсилає запит на з'єднання до вебсервера. У відповідь вебсервер надсилає свій SSL/TLS сертифікат, який містить публічний ключ і інші дані, необхідні для безпечного з'єднання. Браузер перевіряє цей сертифікат, щоб переконатися в його дійсності та надійності. Після успішної перевірки браузер і сервер встановлюють

зашифроване з'єднання, використовуючи публічний і приватний ключі для шифрування даних.

Це означає, що будь-які дані, передані через це з'єднання, залишаються конфіденційними та захищеними від перехоплення або зміни третіми сторонами. Наприклад, коли користувач заходить на свій акаунт в інтернет-банкінгу, його логін і пароль шифруються перед відправкою на сервер. Сервер отримує зашифровані дані, розшифровує їх і проводить необхідні дії для автентифікації користувача.

Використання HTTPS та SSL/TLS забезпечує конфіденційність, цілісність та аутентифікацію даних. Шифрування гарантує, що лише відправник і одержувач можуть переглядати зміст переданих даних, а також що дані не можуть бути змінені під час передачі без виявлення. Сертифікати SSL/TLS також підтверджують особу вебсервера, що запобігає атакам "людина посередині" (Man-in-the-Middle).

Крім того, починаючи з липня 2018 року, увесь HTTP-трафік позначається в URL як «небезпечний». Це сповіщення з’являється для всіх вебсайтів без дійсного сертифіката SSL.

4. Політика безпеки визначає основні ресурси, які необхідно захищати, можливі загрози для цих ресурсів, а також правила та контроль для їх захисту та захисту бізнесу в цілому.

Одним з найважливіших ресурсів, які потрібно захищати, є конфіденційні дані, включаючи особисту інформацію клієнтів, фінансові дані, дані про співробітників, комерційні таємниці та інтелектуальну власність. Інформаційні системи, такі як сервери, бази даних і мережеві пристрої, також є важливими ресурсами, що потребують захисту. Окрім цього, до таких ресурсів належать комунікаційні системи, як електронна пошта та системи обміну повідомленнями, а також фізичні ресурси – офісне обладнання та серверні кімнати.

Компанії часто стикаються з різноманітними загрозами для своїх даних. Багато кібератак спрямовані на співробітників організації, використовуючи їхню неуважність або обманюючи їх через фішингові атаки. Наприклад, зловмисники

можуть надсилати фальшиві електронні листи, що виглядають як повідомлення від надійних джерел, змушуючи співробітників відкривати шкідливі вкладення або переходити за небезпечними посиланнями.

1.3 Штучний інтелект. Принципи роботи

Штучний інтелект – це галузь інформатики, яка займається створенням систем, здатних виконувати завдання, які зазвичай вимагають людського інтелекту. До таких завдань відносяться навчання, розпізнавання мови, візуальне сприйняття, прийняття рішень та переклад мов [7].

Ідеї про створення мислячих машин існували ще з античних часів, де в міфах згадувалися автомати. Проте реальний розвиток штучного інтелекту почався у 20 столітті. У 1950 році Алан Тюрінг у своїй роботі "Computing Machinery and Intelligence" представив тест Тюрінга для визначення розумності машини.

У 1956 році на конференції в Дартмутському коледжі термін "штучний інтелект" був використаний вперше, що вважається народженням ШІ як наукової дисципліни. Подальші дослідження ШІ пройшли через періоди оптимізму та розчарувань, відомі як "зими ШІ".

У 1990-х роках інтерес до ШІ відновився завдяки прогресу в обчислювальній техніці та алгоритмах. Поява інтернету та збільшення обсягів даних сприяли буму досліджень у сфері машинного навчання та глибокого навчання. У 1997 році IBM Deep Blue переміг чемпіона світу з шахів, що стало значним досягненням для ШІ.

У 2010-х роках розвиток глибоких нейронних мереж зробив ШІ ще потужнішим. У 2016 році система AlphaGo від компанії DeepMind перемогла чемпіона світу з гри в го Лі Седоля.

Існують різні напрямки штучного інтелекту.

1. Машинне навчання є видом штучного інтелекту, що зосереджується на створенні комп'ютерних систем, які навчаються на основі даних. Ці методи

дозволяють програмному забезпеченню з часом покращувати свою продуктивність [13].

Алгоритми машинного навчання навчаються знаходити зв'язки та закономірності у даних. Використовуючи існуючі дані як вхідні, вони роблять прогнози, класифікують інформацію, і навіть допомагають генерувати новий контент, як це роблять сучасні програми (Dall-E, Bard, ChatGPT).

По суті, метод, за допомогою якого працюють і навчаються моделі ML, базується на людському досвіді. Хоча комп'ютери не мають вродженої здатності до розуміння та навчання через досвід, алгоритми, які живлять моделі ML, функціонують таким чином, щоб максимально наблизити цей досвід. Завдяки параметрам і налаштуванням алгоритмів моделі ML можуть відтворювати досвідчене навчання. Це дозволяє проводити глибокий аналіз та робити прогнози, які інакше були б неможливі.

Алгоритм, який використовують моделі ML для навчання, створений на основі навчальних даних, з яких модель навчається. Це дозволяє накопичувати досвід у наборі даних, що дає змогу моделям ML експоненціально підвищувати свої здібності до навчання, вивчення і прогнозування, що приносить користь користувачам.

Існує три підкатегорії машинного навчання:

– SL моделі тренуються на спеціально позначених наборах даних, що дозволяє їм з часом ставати більш точними. Наприклад, алгоритм навчається на зображеннях дерев та інших об'єктів, позначених людьми, і потім самостійно розпізнає зображення дерев. Цей тип машинного навчання є найпоширенішим типом сьогодні.

– У UL моделі програма шукає закономірності в непозначених даних. Наприклад, така програма може аналізувати дані про дії користувачів на сайті та виявляти різні типи поведінки.

– RL машинне навчання навчає машини через проби та помилки, встановлюючи систему винагород. Наприклад, такі моделі можуть навчатися

грати в ігри, отримуючи винагороди за правильні дії, що допомагає їм з часом вчитися, які дії слід виконувати.

Один із основних аспектів машинного навчання полягає у використанні алгоритмів для аналізу та інтерпретації даних. Наприклад, коли мова йде про медичну діагностику, моделі машинного навчання можуть аналізувати медичні зображення, такі як рентгенівські знімки або МРТ, і виявляти аномалії з точністю, яка часто перевершує можливості людського ока. Це дозволяє лікарям швидше діагностувати захворювання і почати лікування на ранніх стадіях, що може значно підвищити шанси на успішне одужання пацієнтів.

У фінансовому секторі машинне навчання використовується для прогнозування ринкових тенденцій та виявлення аномалій, які можуть свідчити про шахрайські дії. Алгоритми можуть аналізувати мільйони транзакцій у реальному часі, виявляючи підозрілі патерни та попереджаючи про можливі ризики. Це допомагає банкам та фінансовим установам захищати свої активи і забезпечувати безпеку клієнтів.

Сьогодні потреба та потенціал машинного навчання є більшими, ніж у будь-який інший момент історії. Обсяг та складність даних, що генеруються, є надто великими для обробки людиною. Тому машинне навчання наразі використовується у великій кількості галузей.

2. Обробка природної мови є дуже важливою частиною ШІ, яка дозволяє машинам розуміти, інтерпретувати та генерувати людську мову.

Нейролінгвістичне програмування було розроблено в Каліфорнійському університеті в Санта-Круз у 1970-х роках. Його засновниками були Річард Бендлер, студент математики та інформатики, і Джон Гріндер, професор лінгвістики [20].

NLP заповнює пробіли у спілкуванні між людиною та машиною, змінюючи те, як комп'ютери розуміють, аналізують і генерують людську мову. Головна мета NLP – надати машинам можливість інтерпретувати контекст та значення, які є у людській мові та тексті. NLP полегшує роботу для перекладу мови, спілкування

з чат-ботами до складних завдань, використовуючи обчислювальні алгоритми та штучний інтелект.

Першим кроком є токенізація, яка використовується для поділу рядка слів на корисні одиниці, які називаються токенами. Токенізація речень розбиває текст на окремі речення, а токенізація слів розбиває речення на слова. Зазвичай словесні токени розділяються пробілами, а речення – крапками. Проте, можна виконувати токенізацію для складніших структур мови, таких як словосполучення. Наприклад, речення «Сьогодні я піду в магазин» перетворюється на токени: «сьогодні», «я», «піду», «в», «магазин».

Розпізнавання сутностей є одним із найпопулярніших завдань NLP і включає перелік сутностей із тексту. Сутностями можуть бути імена, моделі, бренди, адреси проживання та інше.

Генерація природної мови створює текст на основі даних, корисний для створення звітів чи новин. Розпізнавання та синтез мовлення перетворюють усну мову в текст і навпаки, що використовується в голосових асистентах, як-от Siri чи Alexa.

Однією з основних складностей для обробки природної мови є те, що людська мова не є однозначною, оскільки навіть самим людям іноді нелегко зрозуміти, що мав на увазі співбесідник. Одним з прикладів викликів є іронія – машині тяжко інтерпретувати речення, де значення тесту протилежне сказаному.

3. Комп'ютерний зір – це технологія, яка дозволяє машинам автоматично розпізнавати та інтерпретувати зображення та відео з високою точністю та ефективністю. Він використовує штучний інтелект і машинне навчання для обробки візуальних даних, таких як розпізнавання об'єктів, розпізнавання осіб, класифікація, рекомендації, моніторинг та виявлення [14].

Раніше обробка візуальної інформації вимагала участі людини, що займало багато часу та було схильно до помилок. Сьогодні завдяки новітнім технологіям в галузі AI ці процеси автоматизовані, що робить додатки комп'ютерного зору доступними для багатьох людей. Ця технологія тепер використовується для верифікації особи, модерації контенту, аналізу відео, тощо.

Основні етапи роботи комп’ютерного зору:

– збирання зображень: система комп'ютерного зору використовує камеру або сенсор для збирання фото, відео або інших візуальних даних. Отримані зображення та відео передаються на комп'ютер для подальшої обробки та аналізу.

– обробка зображень: щоб точно відобразити необхідні дані, зібрані зображення мають бути підготовлені. Даний етап включає зниження шуму, масштабування, обрізку та інші техніки, що допомагають покращити якість зображень і забезпечити їх придатність для подальшого аналізу.

– розуміння зображень: цей етап включає виконання роботи комп'ютерного зору за допомогою моделей глибокого навчання або традиційних методів обробки зображень. Цей процес спрямований на інтерпретацію та аналіз візуальних даних для отримання корисної інформації

На рис. 1.10. показано різницю між людською системою зору та комп’ютерною.

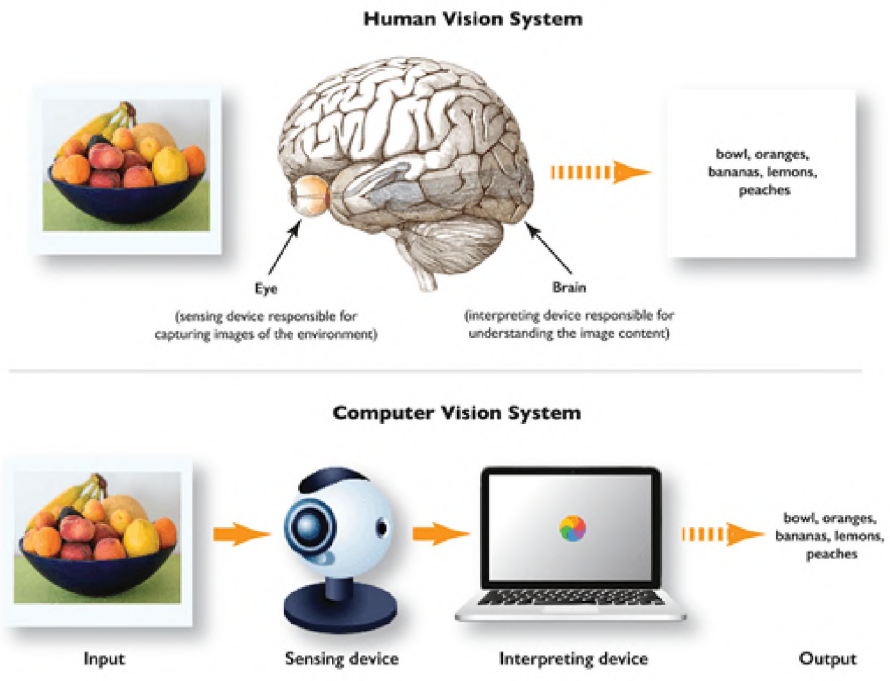

Рисунок 1.10 – Різниця між людською та комп’ютерної системами зору [14]

1.4 Постановка задачі

Традиційні підходи до виявлення та запобігання вебатак, засновані на сигнатурах, правилах і заздалегідь визначених шаблонах, часто виявляються недостатньо ефективними проти нових, витончених атак. Крім того, вони потребують постійного оновлення та адаптації до змінних умов, що ускладнює процес захисту.

У цьому контексті технології штучного інтелекту, особливо методи машинного навчання та глибокого навчання, представляють перспективний підхід до підвищення рівня безпеки вебдодатків. Ці технології дозволяють системам безпеки самостійно навчатися на даних, виявляти аномалії та складні патерни.

Тому, враховуючи актуальність проблеми, а також провівши класифікацію найбільш поширених видів атак, а також методи захисту від них, була поставлена задача розробити підходи стосовно ефективного впровадження технологій штучного інтелекту для виявлення та нейтралізації веб атак.

Для цього потрібно вирішити наступні задачі:

1. Провести аналіз існуючих методів і підходів на основі штучного інтелекту для виявлення та нейтралізації вебатак.
2. Розглянути принципи роботи різних технологій машинного навчання та глибокого навчання, застосовуваних у цій галузі, таких як виявлення аномалій, класифікація атак та інше.
3. Визначити основні підходи застосування ШІ-технологій для захисту вебдодатків від атак.
4. Зробити економічне обґрунтування основних підходів застосування технологій штучного інтелекту для виявлення та нейтралізації вебатак.

1.5 Висновок

У першому розділі було розглянуто основні види вебатак, які становлять загрозу для вебдодатків. Традиційні методи захисту, такі як брандмауери, системи виявлення та шифрування, мають обмеження у виявленні нових типів атак і

потребують постійного оновлення. З наведеної інформації можна зробити висновок, що існуючі технології по протидії атакам, попри переваги, мають певні обмеження, що ставлять під загрозу безпеку вебдодатків. Застосування технологій штучного інтелекту, зокрема машинного навчання, дозволяє покращити безпеку, забезпечуючи більш швидку адаптацію до нових загроз і виявлення аномалій.

РОЗДІЛ 2. СПЕЦІАЛЬНИЙ РОЗДІЛ

2.1 Класифікація вебтрафіку

Аналіз вебтрафіку, який включає передачу даних між клієнтами та серверами, є одним із важливих напрямків застосування ШІ для захисту вебдодатків. Технології машинного навчання дозволяють виявляти аномальні та підозрілі патерни у цих потоках, що можуть свідчити про спроби атак.

Задача класифікації вебтрафіку полягає у розподілі мережевих даних за категоріями, такими як "нормальний", "атака", а також визначення конкретних типів атак. Для вирішення цієї задачі застосовуються алгоритми SL.

Моделі можуть навчатися на наборах даних, що містять мічені приклади нормального трафіку та різних типів атак. Після навчання модель здатна аналізувати вхідний трафік і класифікувати його як нормальний або атакуючий, а також визначати конкретний тип атаки. Це дозволяє системам безпеки швидко реагувати на потенційні загрози і вживати необхідних заходів для їх нейтралізації.

Також важливо, що ефективність класифікації вебтрафіку залежить від якості та кількості даних, що використовуються для навчання моделей. Чим більше даних і чим вони якісніші, тим точнішими будуть прогнози моделі.

1. Алгоритм Random Forest, розроблений Лео Брейманом і Адель Катлером, базується на ідеї використання ансамблю дерев рішень для досягнення більш точних і стабільних результатів [10].

RF використовує bootstrap-агрегування як ансамблевий метод і дерево рішень як окрему модель. Це означає, що деякі зразки даних можуть бути вибрані кілька разів, тоді як інші можуть не бути вибраними зовсім. Даний метод навчання працює шляхом побудови декількох дерев рішень, причому кінцеве рішення приймається на основі більшості дерев і обирається RF.

Як показано на рис. 2.1, bootstrap-агрегування працює наступним чином:

– створюються декілька підмножин з початкового набору даних з рівною кількістю масивів;

– створюється базова модель на кожній з цих підмножин;

– навчання кожної моделі відбувається паралельно на кожному навчальному наборі незалежно одна від одної;

– побудова остаточних прогнозів шляхом об'єднання прогнозів усіх моделей.

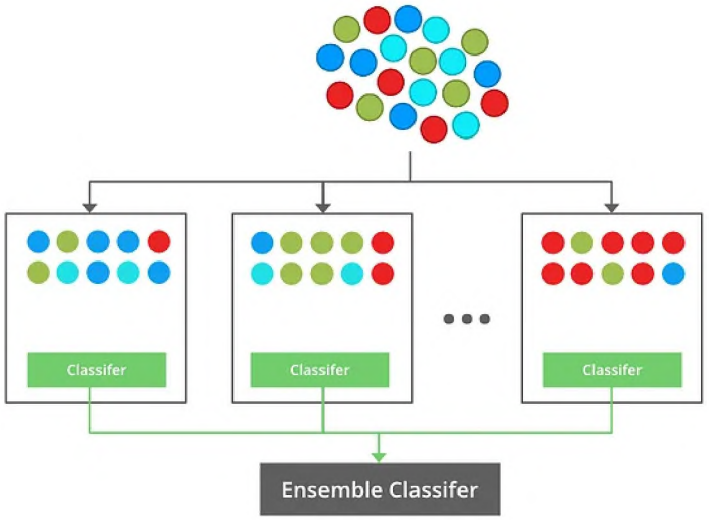

Рисунок 2.1 – Принцип роботи bootstrap-агрегування [1]

Однією з головних переваг використання алгоритму RF є зниження ризику перенавчання та зменшення часу тренування. Також цей алгоритм забезпечує високий рівень точності. RF ефективно працює з великими наборами даних і створює дуже точні прогнози.

При оцінюванні базового класифікатора в машинному навчанні використовуються різні метрики і підходи. Мета цієї оцінки полягає у визначенні того, наскільки добре модель виконує свої завдання з класифікації на тестових даних, що не використовувалися під час її навчання.

Спочатку потрібно виконати імпорт необхідних модулів.

Для роботи буде використано scikit-learn – безкоштовну бібліотеку машинного навчання з відкритим кодом для мови програмування Python.

Модуль DecisionTreeClassifier використовується для створення моделей дерев рішень, які є основним елементом у алгоритмі RF. Дерева рішень – це алгоритм, який використовує модель дерева для прийняття рішень. Дерево рішень розділяє дані на підгрупи на основі певних ознак і створює правила, що визначають, до якої підгрупи належить кожен зразок.

Модуль accuracy_score потрібен для оцінки точності моделі машинного навчання. Точність визначається як частка правильних прогнозів серед загальної кількості прогнозів.

Модуль train_test_split розділяє набори даних на навчальну та тестову вибірки. Цей крок дозволяє оцінити, наскільки добре модель буде працювати на невідомих даних. Ця функція ділить вихідний набір даних на дві частини: тренувальну вибірку, яка використовується для навчання моделі, і тестову вибірку, яка використовується для оцінки її продуктивності.

Далі потрібно завантажити дані та зберегти їх у змінні X (вхідні ознаки) та y (цільова змінна). Параметр as_frame встановлено в значення True, щоб назви ознак не були втрачені при завантаженні даних. Також варто підготувати набір даних, що буде використаний в роботі. Модуль sklearn.datasets надає доступ до різних наборів даних, що використовуються для навчання і тестування моделей машинного навчання. Ці набори даних включають як невеликі набори, так і більш великі реальні набори даних.

Щоб створити модель, яка добре узагальнює нові дані, важливо розділити дані на тренувальний та тестовий набори, для запобігання оцінці моделі на тих самих даних, які використовуються для її навчання.

З підготовленими даними можна створити базовий класифікатор і навчити його на тренувальному наборі. Функція fit у DecisionTreeClassifier використовується для навчання моделі на основі наданих даних. Вона приймає два основні параметри: X_train (вхідні дані) та y_train (цільові значення). Під час виконання цієї функції модель аналізує надані дані та будує дерево рішень, яке може бути використане для прогнозування на нових даних.

Після виконаних кроків, тепер є можливість передбачити тип атаки на тестовому наборі та оцінити продуктивність моделі.

Функція predict використовується для здійснення прогнозів на основі навченої моделі. Після того як модель була навчена на навчальних даних за допомогою функції fit, функція predict дозволяє використовувати цю модель для передбачення результатів на нових, невідомих даних.

Лістинг коду:

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
data = datasets.load_attacks(as_frame=True)
X = data.data
y = data.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25,
random_state=22)
dtree = DecisionTreeClassifier(random_state = 22)
dtree.fit(X_train,y_train)
y_pred = dtree.predict(X_test)
accuracy_score(y_true=y_train, y_pred=dtree.predict(X_train))
accuracy_score(y_true=y_test, y_pred=y_pred)
```

Результат виконання наданого коду наведено на рис. 2.2.

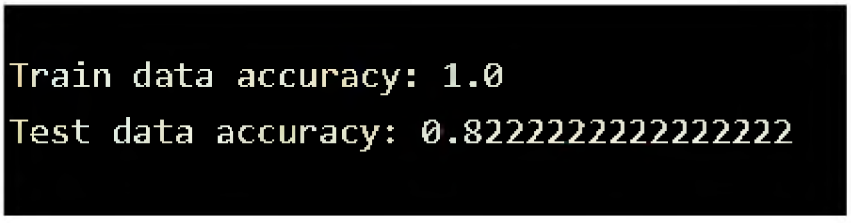

Рисунок 2.2 – Результат виконання коду базового класифікатора Decision Tree

Таким чином, початкова модель класифікації показала досить високу точність, досягнувши 82% на тестовому наборі даних з поточними налаштуваннями.

2. Логістична регресія – це алгоритм машинного навчання, що використовується для класифікаційних завдань, де метою є передбачити ймовірність належності зразка до певного класу [4].

Логістична регресія використовується для бінарної класифікації, де застосовується сигмоїдальна функція, яка приймає на вхід незалежні змінні та генерує значення ймовірності в діапазоні від 0 до 1.

Основи логістичної регресії:

– математична модель: логістична регресія використовує сигмоїдальну функцію, щоб перетворити лінійну комбінацію вхідних ознак на ймовірність;

– ймовірнісна інтерпретація: на відміну від лінійної регресії, логістична регресія передбачає значення в межах від 0 до 1, що інтерпретується як ймовірність приналежності до певного класу. Якщо ймовірність більше 0.5, то зразок належить до класу 1, інакше – до класу 0;

– функція втрат: для навчання моделі логістичної регресії використовується логістична функція втрат, яка оцінює відмінність між передбаченими ймовірностями та фактичними мітками класів.

Логістична регресія може використовуватися для вирішення різних задач, починаючи від прогнозування приросту користувачів до виявлення шахрайської діяльності. Наприклад, бізнес може використовувати логістичну регресію для аналізу даних клієнтів і прогнозування, чи ймовірно, що вони придбають певний продукт.

Для оцінки роботи логістичної регресії буде використано сучасну мову розробки Python.

Для цього буде взято numpy – бібліотеку для роботи з масивами і виконання чисельних обчислень, та sklearn.linear_model – модуль з бібліотеки scikit-learn, який містить реалізацію логістичної регресії.

Цей код виконує класифікацію значень за їх розмірами за допомогою логістичної регресії. Модель навчається на наданих даних, що містять числа та інформацію про їх статус. Після навчання модель використовується для прогнозування того, чи буде нове значення 4,13 з класом 1 чи 0.

Лістинг коду:

```
import numpy
from sklearn import linear_model
X = numpy.array([3.64, 2.08, 3.13, 1.18, 1.12, 4.92, 4.12, 4.73, 3.69, 5.19]).reshape(-1,1)
y = numpy.array([0, 0, 0, 0, 0, 1, 1, 1, 1, 1])
logr = linear_model.LogisticRegression()
logr.fit(X,y)
predicted = logr.predict(numpy.array([4.13]).reshape(-1,1))
```

В результаті виконання коду було отримано результат 1, що вказує на належність значення 4,13 до класу 1.

3. Методи опорних векторів - це набір методів машинного навчання з учителем, які використовуються для класифікації та регресії. Це поширені завдання в машинному навчанні.

SVM прагне знайти найкращу гіперплощину в багатовимірному просторі ознак, що розділяє точки даних різних класів з максимальним відступом. Гіперплощина є роздільною межею, яка розділяє точки даних на два класи. Метою є знайти гіперплощину, яка максимізує відступ, що є відстанню між гіперплощиною та найближчими точками даних кожного класу.

У SVM гіперплощина представлена вектором, перпендикулярним до неї, який називається нормальним вектором або вектором ваг. Цей вектор визначається під час навчання SVM. Процес навчання включає знаходження оптимальних ваг, які визначають гіперплощину, розв'язуючи оптимізаційну задачу. Після знаходження гіперплощини, її можна використовувати для класифікації нових точок даних.

Під час класифікації нової точки даних за допомогою SVM, алгоритм обчислює скалярний добуток між вектором ваг та вектором ознак нової точки даних. Вектор ознак представляє атрибути або характеристики точки даних. Скалярний добуток фактично вимірює схожість або проекцію нової точки даних на вектор ваг.

Якщо скалярний добуток є додатним, це означає, що нова точка даних знаходиться на тій же стороні гіперплощини, що й позитивний клас. Відповідно, якщо скалярний добуток є від'ємним, це означає, що нова точка даних знаходиться на тій же стороні гіперплощини, що й негативний клас. Величина скалярного добутку також вказує на близькість точки даних до гіперплощини.

Після знаходження гіперплощини у SVM, класифікація нової точки даних включає обчислення скалярного добутку між вектором ваг (перпендикулярним до гіперплощини) та вектором ознак нової точки даних. Знак і величина цього скалярного добутку визначають мітку класу і близькість нової точки даних до гіперплощини.

Однією з областей використання SVM є текстова класифікація та аналіз настрою. Цей метод широко застосовується для класифікації текстів, наприклад, для визначення спаму в електронній пошті або аналізу відгуків у соціальних мережах. Використання SVM у текстовій класифікації дозволяє точно визначати контекст і настрій текстів.

Ще однією важливою областю застосування SVM є розпізнавання образів, таких як рукописні цифри або класифікація зображень.

У медичній сфері SVM також широко застосується, наприклад, для діагностики захворювань на основі медичних зображень або аналізу генетичних даних.

Далі буде надано використання SVM для класифікації за допомогою бібліотеки scikit-learn.

Лістинг коду:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV
```

```
from sklearn.model_selection import train_test_split
df = pd.read_csv("data_4.csv")
cols = df.columns
cols = cols.map(lambda x: x.replace(' ', '_') )
df.columns = cols
query = df.query('Dst_Port == 80 or Dst_Port == 443')
df=query
inf=df.isin([np.inf, -np.inf])
df=df.replace([np.inf, -np.inf], np.nan).dropna(axis=1)
df = df.drop(columns=['Timestamp', 'Flow_ID', 'Src_IP', 'Dst_IP'])
df.Label[df.Label=='Benign'] = 0
df.Label[df.Label =='DDoS attacks-LOIC-HTTP'] = 1
ddos_count = df[df['Label'] == 1].shape[0]
benign_count = df[df['Label'] == 0].shape[0]
df_benign = df[df['Label'] == 0].sample(n=ddos_count, random_state=42)
df_reduced = pd.concat([df[df['Label'] == 1], df_benign])
df_reduced = df_reduced.sample(frac=1, random_state=42)
df = df_reduced
bening_df = df[df['Label']==0]
malignant_df = df[df['Label']==1]
axes = bening_df.plot(kind='scatter', x='Flow_Duration', y = 'Tot_Fwd_Pkts',
color='blue', label='Benign')
malignant_df.plot(kind='scatter', x='Flow_Duration', y = 'Tot_Fwd_Pkts',
color='red', label='maligmant', ax=axes)
df = df.sample(frac=1, random_state=42).reset_index(drop=True)
num_data = 2000
train_df = df.iloc[:num_data].copy()
train_df = train_df.astype("float64")
target = np.asanyarray(train_df.pop('Label'))
raw = np.asanyarray(train_df)
```

```
# replace infinite values with a large finite
raw[~np.isfinite(raw)] = np.finfo(raw.dtype).max
# replace NaN values with zero
raw = np.nan_to_num(raw)
# create train_df nparray varibel
raw = np.asanyarray(train_df)
test_size = 0.3
random_state = 42
X_train, X_test, y_train, y_test = train_test_split(raw, target,
test_size=test_size, random_state=random_state)
param_grid = {
   'kernel': ['rbf'],
   'C': [0.1, 1, 10, 100],
   'gamma': ['scale', 'auto', 0.1, 1, 10]
}
svm_model = SVC()
grid_search = GridSearchCV(svm_model, param_grid, cv=50, n_jobs=-1,
verbose=1)
grid_search.fit(X_train, y_train)
best_svm_model = grid_search.best_estimator_
y_pred = best_svm_model.predict(X_test)
class_report = classification_report(y_test, y_pred)
class_report_dict = {
   '0.0': {'precision': 0.98, 'recall': 0.93, 'f1-score': 0.96, 'support': 324},
   '1.0': {'precision': 0.92, 'recall': 0.98, 'f1-score': 0.95, 'support': 276}
}
palette = sns.color_palette('pastel')
class_names = sorted(list(class_report_dict.keys()))
metrics = ['precision', 'recall', 'f1-score']
fig, axs = plt.subplots(1, len(metrics), figsize=(15, 5))
```

```
for i, metric in enumerate(metrics):
    scores = [class_report_dict[class_name][metric] for class_name in
class_names]
    ax = sns.barplot(x=class_names, y=scores, ax=axs[i], palette=palette)

    ax.set(title=metric.capitalize() + ' by class', ylabel=metric.capitalize(),
ylim=(0, 1), facecolor='white')
fig.suptitle('Classification Report', fontsize=16, fontweight='bold')
plt.show()
```

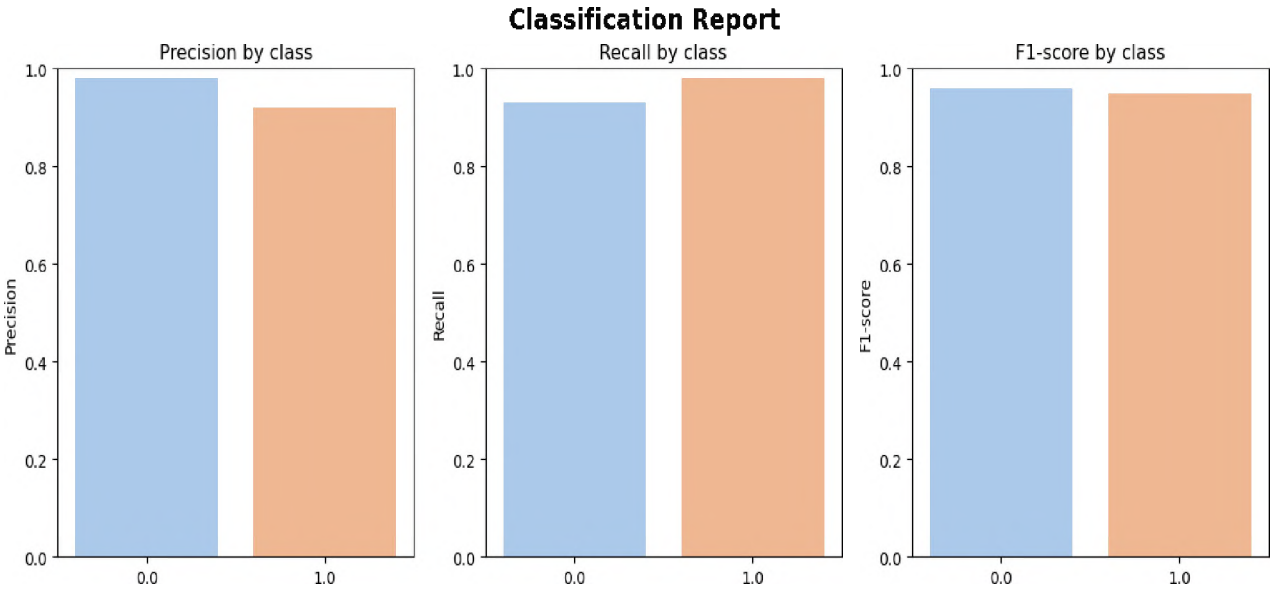

Рисунок 2.3 – Результат виконання коду з класифікації у вигляді графіку [3]

Ці алгоритми навчаються на розмічених наборах даних, що містять приклади нормального вебтрафіку та різних видів атак. У процесі навчання вони будують моделі, здатні розпізнавати характерні ознаки та патерни, що відрізняють атаки від легітимного трафіку.

2.2 Виявлення аномалій

Виявлення аномалій за допомогою штучного інтелекту знаходить відхилення в даних, які відрізняються від норми, використовуючи алгоритми машинного навчання. AI використовує складні моделі, що навчаються на даних

з часом, адаптуючись до нових шаблонів. Тут можуть використовуватися як алгоритми навчання з учителем, та і методи навчання без учителя: кластеризація, аналіз головних компонент, автокодери та моделювання багатовимірних розподілів

Ці методи будують моделі "нормальної" поведінки вебтрафіку та потім виявляють відхилення від цих моделей, що можуть бути ознаками атак.

Системи виявлення атак на основі ШІ використовують комбінацію різних методів ML для забезпечення захисту системи. Вони можуть аналізувати великі обсяги трафіку в режимі реального часу, виявляючи аномалії, які вказують на потенційні атаки.

Виявлення аномалій в режимі реального часу є аспектом системи виявлення аномалій, керованої ШІ. Воно забезпечує безперервний моніторинг потоків даних, вчасно сповіщаючи користувачів про будь-які виявлені аномалії. Цей підхід дозволяє своєчасно виявляти та усувати потенційні проблеми, мінімізуючи простої та максимізуючи продуктивність. Аналізуючи великий обсяг даних з різних джерел, система може швидко виявляти аномальні шаблони або поведінку. Завдяки потужності ШІ та використанню алгоритмів машинного навчання, вона постійно підвищує точність виявлення потенційних аномалій.

Автоматичне налаштування та навчання моделей відіграють важливу роль у методах виявлення аномалій, керованих ШІ. Автоматично налаштовуючи та навчаючи моделі, ПЗ може адаптуватися до змін у шаблонах даних, підвищуючи точність виявлення аномалій. Це заощаджує час і зусилля в порівнянні з ручними налаштуваннями і дозволяє виробникам швидко ідентифікувати та вирішувати потенційні проблеми у своїх виробничих процесах. Завдяки можливості обробляти великі обсяги даних та оновлювати моделі в режимі реального часу, автоматичне налаштування та навчання моделей використовують потужність ШІ, підвищуючи можливості виявлення аномалій та покращуючи управління ризиками.

Основна перевага використання ML для виявлення аномалій полягає в підвищеній точності ідентифікації відхилень. Навчаючись на даних, системи ML

можуть адаптуватися та вдосконалювати свої критерії для визначення нормальної поведінки та аномалій. Ця адаптивність дозволяє більш тонко розуміти дані, що веде до точнішого виявлення аномалій. Здатність алгоритмів ML враховувати широкий спектр факторів та їх взаємозв'язки означає, що аномалії можуть бути ідентифіковані з вищим ступенем впевненості, знижуючи ймовірність пропуску критичних нерегулярностей або помилкового сигналізування про хибні тривоги.

Сучасні системи виявлення атак часто використовують гібридні підходи, що поєднують наглянуте та ненаглянуте навчання. Вони складаються з декількох компонентів:

– збір даних: мережеві пакети містять інформацію про всі вхідні та вихідні дані, що проходять через мережу, дозволяючи виявляти підозрілі патерни у трафіку. Журнали подій системи реєструють різні дії та події, що відбуваються у системі, включаючи входи в систему, помилки, зміни конфігурацій та інші важливі події. Файли журналів додатків також надають цінну інформацію про активність додатків та їхні взаємодії з користувачами. Зібрані дані попередньо обробляються для видалення шуму, нормалізації та структуризації, що робить їх придатними для подальшого аналізу.

– аналіз даних: після збору даних вони передаються на етап аналізу, де використовуються алгоритми машинного навчання для виявлення аномалій. Наглянуте навчання передбачає використання мічених даних, де кожен зразок вже класифікований як нормальний або аномальний. Це дозволяє системі вчитися на основі відомих прикладів загроз і нормальної поведінки. Алгоритми ненаглянутого навчання, навпаки, працюють з немічеными даними, знаходячи патерни та аномалії без попереднього знання про їхню природу. Використовуючи обидва підходи, система може виявляти як відомі, так і нові, раніше невідомі загрози. Для аналізу даних можуть використовуватися різні техніки, включаючи кластеризацію, класифікацію, виявлення аномалій та глибинне навчання.

– реагування на загрози: після виявлення загроз система переходить до етапу реагування. Це може бути автоматичне або ручне реагування на виявлені загрози. Автоматичне реагування включає блокування підозрілого трафіку, ізоляцію скомпрометованих систем або облікових записів, а також застосування відповідних патчів та оновлень для усунення вразливостей. Ручне реагування передбачає повідомлення адміністраторів або аналітиків безпеки, які можуть додатково дослідити інцидент, підтвердити загрозу і вжити відповідних заходів. Це включає детальний аналіз журналів подій, проведення розслідувань та вжиття заходів для запобігання повторення таких інцидентів у майбутньому.

Однією з ключових переваг ML у виявленні аномалій є здатність ефективно обробляти та аналізувати неструктуровані дані. Традиційні методи часто покладаються на попередньо визначені правила або вимагають, щоб набори даних були акуратно організовані та мали мітки, що не завжди є можливим, особливо з урахуванням величезних обсягів даних, що генеруються сьогодні. Алгоритми ML, однак, можуть навчатися безпосередньо на даних, визначаючи шаблони та норми без необхідності в явних мітках. Ця здатність особливо цінна в сценаріях, де дані складні, а міткування є непрактичним або неможливим.

Ще однією значною перевагою вдосконаленого ML виявлення аномалій є підвищена чутливість системи у розрізненні справжніх аномалій від звичайного шуму. У будь-якому наборі даних завжди буде певний ступінь варіабельності або шуму, що є нормальним та очікуваним. Важливо відрізняти цей шум від справжніх аномалій, щоб уникнути хибних спрацьовувань і забезпечити, щоб були позначені лише значущі відхилення для подальшого розслідування. Алгоритми ML відмінно справляються з цим завданням, використовуючи передові аналітичні техніки для оцінки ступеня відхилення та визначення, чи є це відхиленням від норми на основі контексту та внутрішніх характеристик даних.

2.3 Виявлення аномалій за допомогою DBSCAN кластеризації

DBSCAN є алгоритмом кластеризації на основі густини, який групує точки даних на основі їх близькості та густини. Алгоритм має два основних параметри: епсілон та мінімальна кількість точок. Епсілон визначає радіус навколо точки даних, тоді як мінімальна кількість точок є значенням, необхідним для утворення густої області [11].

Алгоритм DBSCAN класифікує кожну точку даних у три категорії: основна точка, прикордонна точка і викид. Для кожної точки даних, якщо у неї є принаймні мінімальна кількість точок у межах відстані Епсілон, вона є основною точкою. Якщо в межах відстані Епсілон є менше точок, аніж мінімальна кількість точок, але точка знаходиться в сусідстві з основною точкою, то це прикордонна точка. В іншому випадку, точка даних є викидом. Для визначення розташування точок у просторі DBSCAN переважно використовує евклідову відстань, хоча можуть використовуватися й інші методи.

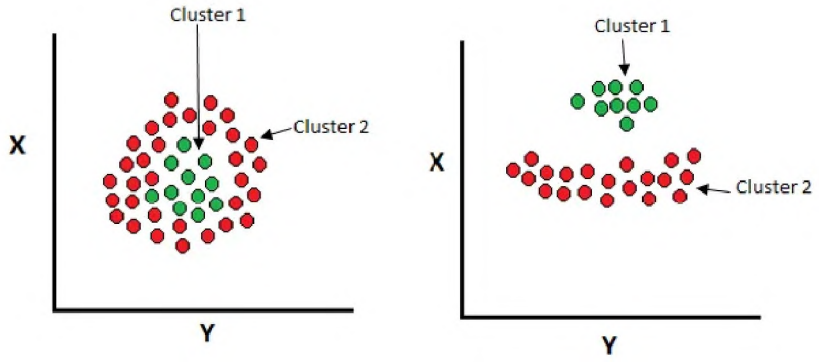

Рисунок 2.4 – Принцип роботи DBSCAN кластеризації [11]

DBSCAN дуже ефективний у виявленні аномалій у складних та зашумлених наборах даних. Прикладами застосування даного алгоритму можуть бути:

– кластерування даних: DBSCAN широко використовується для кластерування даних, наприклад, для виявлення регіонів з високим рівнем злочинності або поширення хвороб;

– виявлення аномалій: DBSCAN корисний для виявлення аномалій, оскільки ізолює шумові точки, які розташовані далеко від густонаселених областей. Ці ізольовані точки вважаються потенційними аномаліями або відхиленнями в даних;

– сегментація клієнтів: у маркетингу DBSCAN допомагає в сегментації клієнтів, визначаючи групи клієнтів з подібними покупательськими поведінками. Це дозволяє компаніям більш ефективно цілити на конкретні сегменти клієнтів, оптимізуючи маркетингові стратегії та підвищуючи задоволеність клієнтів.

Алгоритм DBSCAN для виявлення аномалій у Python можна написати за допомогою бібліотеки scikit-learn. Синтетичний набір даних для демонстрації було створено в процесі реалізації.

Лістинг коду:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs
from sklearn.cluster import DBSCAN
from sklearn.preprocessing import StandardScaler
data, _ = make_blobs(n_samples=300, centers=3, random_state=42)
outliers = np.random.uniform(low=-10, high=10, size=(20, 2))
data = np.vstack([data, outliers])
scaler = StandardScaler()
data = scaler.fit_transform(data)
epsilon = 0.3
min_pts = 10
dbscan = DBSCAN(eps=epsilon, min_samples=min_pts)
dbscan.fit(data)
labels = dbscan.labels_
core_samples_mask = np.zeros_like(labels, dtype=bool)
core_samples_mask[dbscan.core_sample_indices_] = True
```

```
unique_labels = set(labels)
colors = [plt.cm.Spectral(each) for each in np.linspace(0, 1, len(unique_labels))]
for k, col in zip(unique_labels, colors):
    if k == -1:
        col = [0, 0, 0, 1]  # чорний для викидів
    class_member_mask = (labels == k)
    xy = data[class_member_mask & core_samples_mask]
    plt.scatter(xy[:, 0], xy[:, 1], c=[col], s=50, linewidths=0.5, edgecolors='k')
    xy = data[class_member_mask & ~core_samples_mask]
    plt.scatter(xy[:, 0], xy[:, 1], c=[col], s=50, linewidths=0.5, edgecolors='k',
alpha=0.5)
```

У цьому коді спочатку генерується синтетичний набір даних за допомогою функції make_blobs, потім додаються деякі викиди за допомогою numpy.random.uniform. Після цього набір даних стандартизується за допомогою StandardScaler. Алгоритм DBSCAN застосовується до набору даних з параметрами epsilon та min_samples, встановленими на 0,3 та 10 відповідно.

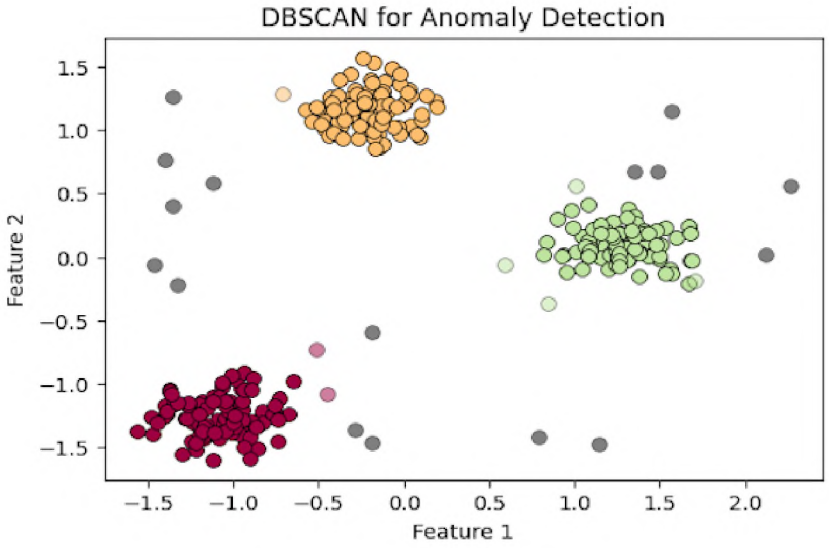

Рисунок 2.5 – Результат виконання коду DBSCAN кластерізації [11]

2.4 Нейтралізація вебатак з використанням ШІ

Штучний інтелект не тільки здатен виявляти вебатаки, але й активно нейтралізувати їх. Існують різні методи та підходи, за допомогою яких ШІ може запобігати і знешкоджувати атаки на вебдодатки.

1. ШІ здатен в реальному часі аналізувати мережевий трафік і автоматично блокувати підозрілі запити. Наприклад, якщо система виявляє раптове збільшення кількості запитів з одного або кількох IP-адрес, або ж інші нетипові патерни, це може свідчити про потенційну DDoS-атаку. Після виявлення підозрілого трафіку система ШІ ідентифікує IP-адреси, з яких надходять ці запити. Це можуть бути IP-адреси ботнетів, що використовуються для здійснення атаки. Система автоматично блокує ці IP-адреси, запобігаючи подальшому надходженню шкідливих запитів. Крім того, у системи є можливість постійно оновлювати список заблокованих IP-адрес, аналізуючи нові дані та адаптуючись до змін у патернах атак. Нею також можуть бути розблоковані IP-адреси, якщо визначиться, що вони більше не становлять загрози.

2. Системи управління загрозами на базі ШІ можуть автоматично реагувати на інциденти безпеки. Це може включати ізоляцію скомпрометованих облікових записів, автоматичну зміну паролів або обмеження доступу до певних ресурсів. Якщо виявлено спробу SQL-ін'єкції, система може тимчасово заблокувати обліковий запис користувача, що ініціював підозрілий запит, і оповістити адміністратора.

3. Чат-боти на базі ШІ можуть бути використані для виявлення та нейтралізації атак соціальної інженерії. Ці типи атак, які спрямовані на маніпуляцію людьми з метою отримання конфіденційної інформації, можуть бути дуже підступними та ефективними. Однак, чат-боти можуть діяти як перша лінія оборони, автоматично відповідати на підозрілі повідомлення або запити, відволікати зловмисників і збирати корисну інформацію для подальшого аналізу. Відповіді можуть бути розроблені таким чином, щоб здаватися звичайними, але фактично відволікати зловмисника та збирати більше інформації про його наміри. Наприклад, чат-бот може задавати додаткові запитання або надавати нейтральні

відповіді, які не містять конфіденційної інформації. Крім того, вони можуть бути інтегровані з іншими системами безпеки для більш комплексного підходу до захисту організації. Наприклад, вони можуть працювати разом з системами виявлення вторгнень та системами управління інформацією та подіями безпеки, передаючи зібрану інформацію для подальшого аналізу та реакції.

4. Системи на базі ШІ можуть сканувати програмне забезпечення на предмет вразливостей, використовуючи методи машинного навчання для аналізу коду і виявлення потенційних слабких місць. Вони здатні аналізувати великі обсяги даних у реальному часі, знаходячи патерни та аномалії, що можуть свідчити про наявність вразливостей. Наприклад, ШІ може ідентифікувати специфічні фрагменти коду, які містять небезпечні функції або неправильні конфігурації, що можуть бути використані зловмисниками. Після виявлення вразливостей системи на базі ШІ можуть автоматично завантажувати та встановлювати необхідні оновлення та патчі. Це дозволяє оперативно усувати вразливості, зменшуючи час, протягом якого система залишається вразливою до атак. Автоматизація цього процесу значно знижує ризики, пов'язані з людським фактором, такими як затримки в застосуванні оновлень або помилки під час їх встановлення.

5. Реактивні системи захисту на базі ШІ при виявленні можуть автоматично вживати низку контрзаходів для нейтралізації атаки. Наприклад, при виявленні спроби XSS-атаки система може автоматично вносити зміни до коду вебсторінок, щоб запобігти подальшим атакам. Це може включати додавання відповідних фільтрів, очищення вхідних даних або внесення змін до конфігурацій вебсерверів. Наприклад, при виявленні системою спроби XSS-атаки на вебсайт, система може аналізувати шкідливий код, визначати його джерело та характер. Потім вона може автоматично застосувати необхідні заходи, такі як очищення вхідних даних, введення нових правил для фільтрації небезпечних скриптів та внесення змін до конфігурацій безпеки вебсервера, щоб запобігти повторним спробам атаки.

WAF на базі штучного інтелекту здатні ефективно блокувати вебатаки, використовуючи алгоритми машинного навчання для виявлення та нейтралізації загроз.

Вони починають з постійного збору даних про мережевий трафік і активність вебдодатків, аналізуючи HTTP/HTTPS-запити, що надходять до сервера, щоб виявити будь-які підозрілі патерни або аномалії. Завдяки інтеграції з іншими системами безпеки та базами даних загроз, WAF можуть порівнювати вхідні запити з відомими атаками та загрозами.

Найпопулярніші WAF-атаки та їхнє порівняння:

AWS WAF є потужним інструментом для захисту вебдодатків, що працює на платформі Amazon. Він використовує алгоритми машинного навчання для виявлення та блокування шкідливого трафіку в реальному часі. Основні особливості AWS WAF включають захист від широкого спектру атак, включаючи SQL-ін'єкції, XSS та DDoS-атаки, а також можливість створення власних правил та налаштувань для захисту конкретних вебдодатків. Інтеграція з іншими сервісами AWS, такими як Amazon CloudFront, забезпечує багаторівневий захист. До переваг AWS WAF належать висока продуктивність та масштабованість, легкість інтеграції з існуючими сервісами AWS та підтримка автоматичного оновлення правил безпеки. Однак, вартість цього рішення може бути високою для невеликих організацій, і воно залежить від інфраструктури AWS [12].

Azure Web Application Firewall є компонентом Microsoft Azure. Це рішення інтегрується з Azure Front Door, Application Gateway та Azure CDN, забезпечуючи захист від OWASP Top 10 атак та автоматичне налаштування та оновлення правил безпеки на основі аналізу загроз. Azure WAF має глибоку інтеграцію з іншими сервісами Azure, підтримує гнучкі налаштування безпеки та автоматичне масштабування відповідно до навантаження. Проте, можливі складнощі з налаштуванням для нових користувачів, а вартість може зростати при збільшенні обсягів трафіку [22].

Cloudflare WAF є частиною комплексного рішення Cloudflare для захисту вебдодатків. Він забезпечує захист від вебатак та підтримує автоматичне оновлення правил безпеки на основі глобального аналізу. До переваг Cloudflare WAF належать простота використання та налаштування, висока швидкість та продуктивність, а також глобальна мережа доставки контенту для підвищення продуктивності вебдодатків. Деякі функції доступні лише у платних версіях, а можливості для підлаштування обмежені [21].

F5 Advanced WAF відомий своїми можливостями для захисту від ботів та атаками на основі поведінкового аналізу. Цей засіб забезпечує захист від вже зазначених атак, а також має можливість виявляти та блокувати ботів на основі поведінкового аналізу, а також інтеграцію з іншими продуктами F5 для забезпечення комплексного захисту. Його перевагами є потужні можливості для захисту від ботів, підтримка гнучких налаштувань та кастомізації, а також висока ефективність у виявленні складних загроз, проте у нього висока вартість ліцензії та обслуговування [23].

Таким чином, кожне з розглянутих рішень WAF на базі штучного інтелекту має свої особливі характеристики та переваги. AWS WAF, інтегрований з іншими сервісами AWS, забезпечує високу продуктивність і масштабованість, проте може бути дорогим для невеликих організацій. Azure WAF пропонує тісну інтеграцію з екосистемою Azure і підтримує автоматичне масштабування відповідно до навантаження, але налаштування можуть бути складними для нових користувачів, і вартість може зростати з обсягом трафіку. Загалом, використання WAF з інтегрованим ШІ є потужним рішенням, що могло би знизити потенційні ризики користувачів та організацій.

2.5 Висновок

У даному розділі було розглянуто основні підходи до застосування технологій штучного інтелекту для виявлення вебатак. Зокрема, були проаналізовані методи класифікації вебтрафіку за допомогою алгоритмів RF, SVM та логістичної регресії. Також було розглянуто питання виявлення аномалій

у вебтрафіку за допомогою алгоритму DBSCAN, що дозволяє ізолювати шумові точки як потенційні аномалії.

Крім того, було досліджено засоби нейтралізації вебатак на базі Web Application Firewall, які забезпечують захист вебсерверів та вебдодатків шляхом виявлення та блокування шкідливого трафіку. Були надані порівняльні характеристики різних типів WAF, включаючи їхні переваги та недоліки, що дозволяє визначити найбільш ефективні рішення для конкретних сценаріїв.

РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ

3.1 Постановка задачі

Метою розділу є обґрунтування економічної доцільності розробки засобів застосування технологій штучного інтелекту для ефективного виявлення та нейтралізації вебатак. Для виконання цієї мети необхідно виконати наступні розрахунки:

– капітальні витрати на придбання і налагодження складових системи інформаційної безпеки або витрат, що пов'язані з виготовленням апаратури, приладів, програмного забезпечення;

– річні експлуатаційні витрати на утримання і обслуговування об'єкта проєктування;

– річний економічний ефект від впровадження об'єкта проєктування.

– показники економічної ефективності запропонованого в дипломному проєкті проєктного рішення.

3.2 Розрахунок капітальних (фіксованих) витрат

Капітальні інвестиції – це кошти, вкладені в придбання, модернізацію або утримання довгострокових активів компанії, які використовуються для забезпечення її діяльності. Ці інвестиції спрямовані на підвищення виробничих потужностей, зниження витрат або поліпшення якості продукції чи послуг [24].

Компоненти капітальних витрат:

– основні фонди: придбання або модернізація виробничого обладнання, будівель, споруд, транспортних засобів та іншої нерухомості. Це можуть бути нові верстати, комп'ютерна техніка, транспортні засоби, офісне обладнання тощо.

– нематеріальні активи: інвестиції в нематеріальні активи, такі як патенти, авторські права, торгові марки, ліцензії на програмне забезпечення та інші інтелектуальні ресурси, що мають тривалу вартість і забезпечують конкурентні переваги.

– інфраструктурні інвестиції: кошти, вкладені в розвиток інфраструктури, включаючи комунікаційні мережі, системи енергопостачання, транспортні системи та інші інфраструктурні проєкти, що сприяють поліпшенню умов діяльності компанії.

– інвестиції в людський капітал: витрати на навчання, підвищення кваліфікації та розвиток персоналу. Це можуть бути освітні програми, семінари, тренінги та сертифікаційні курси, що підвищують професійний рівень працівників.

– інновації та дослідження: інвестиції в дослідження і розробки нових продуктів, технологій та рішень, що сприяють інноваційному розвитку компанії. Це включає фінансування наукових досліджень, розробку прототипів, тестування нових продуктів та технологій.

Капітальні інвестиції відіграють важливу роль у розвитку та конкурентоспроможності компанії. Вони дозволяють збільшити виробничі потужності, поліпшити якість продукції, знизити витрати, підвищити ефективність діяльності та забезпечити довгострокове зростання компанії. Крім того, капітальні інвестиції сприяють створенню нових робочих місць та розвитку економіки в цілому.

Трудомісткість реалізації – це показник, що відображає обсяг праці, необхідний для виконання певної задачі або проєкту. Вона відображає, скільки зусиль потрібно витратити на реалізацію певного обсягу робіт.

Трудомісткість розробки системи захисту інформації на підприємстві визначається тривалістю кожної робочої операції:

$$T = t_{тз} + t_{в} + t_{а} + t_{вз} + t_{озб} + t_{овр} + t_{д}, \quad (3.1)$$

де $t_{тз}$ – тривалість складання технічного завдання на розробку підходу до застосування технологій штучного інтелекту для ефективного виявлення та нейтралізації вебатак, $t_{тз} = 27$;

$t_в$ – тривалість вивчення технічного завдання, літературних джерел за темою тощо, $t_в = 34$;

$t_а$ – тривалість процесу аналізу ризиків, $t_а = 43$;

$t_{вз}$ – тривалість визначення вимог до заходів, методів та засобів захисту, $t_{вз} = 31$;

$t_{озб}$ – тривалість вибору основних рішень з забезпечення безпеки застосування технологій штучного інтелекту для ефективного виявлення та нейтралізації вебатак, $t_{озб} = 28$;

$t_{овр}$ – тривалість виконання відновлювальних робіт і забезпечення неперервного функціонування організації, $t_{овр} = 13$;

$t_д$ – тривалість документального оформлення методів, $t_д = 10$.

Отже,

$$t = 27 + 34 + 43 + 31 + 28 + 13 + 10 = 186 \text{ годин.}$$

Розрахунок витрат на створення політики безпеки інформації.

Витрати на розробку політики безпеки інформації $К_{рп}$ складаються з витрат на заробітну плату спеціаліста з інформаційної безпеки $З_{зп}$ і вартості витрат машинного часу, що необхідний для розробки політики безпеки інформації $З_{мч}$:

$$К_{рп} = З_п + З_{мч}, \quad (3.2)$$

Заробітна плата виконавця враховує основну і додаткову заробітну плату, а також відрахування на соціальне потреби (пенсійне страхування, страхування на випадок безробіття, соціальне страхування тощо) і визначається за формулою:

$$З_{зп} = t \cdot З_{іб}, \text{ грн.}, \quad (3.3)$$

де t – загальна тривалість розробки підходу щодо застосування технологій штучного інтелекту для ефективного виявлення та нейтралізації вебатак, годин;

$З_{іб}$ – средньогодинна заробітна плата спеціаліста з інформаційної безпеки з нарахуванням, грн./годину.

Средньогодинна заробітна плата спеціаліста з інформаційної безпеки становить 200 грн./годину.

Заробітна плата виконавця враховує основну і додаткову заробітну плату, а також відрахування на соціальне потреби (пенсійне страхування, страхування на випадок безробіття, соціальне страхування тощо) і визначається за формулою 3.3:

$$З_{зп} = 186 \cdot 200 = 37200 \text{ грн.} \quad (3.4)$$

Вартість машинного часу для розробки політики безпеки інформації на ПК визначається за формулою:

$$З_{мч} = t \cdot С_{мч}, \quad (3.5)$$

де t – трудомісткість розробки підходу щодо застосування технологій штучного інтелекту для ефективного виявлення та нейтралізації вебатак, годин;

$С_{мч}$ – вартість 1 години машинного часу ПК, грн./година.

Вартість 1 години машинного часу ПК визначається за формулою:

$$С_{мч} = Р \cdot t_{нал} \cdot С_е + \frac{Ф_{зал} \cdot Н_а}{F_р} + \frac{К_{лпз} \cdot Н_{апз}}{F_р}, \text{ грн.}, \quad (3.6)$$

де Р – встановлена потужність ПК, кВт;

$t_{нал}$ – кількість задіяних робочих станцій при створенні системи захисту;

$С_е$ – тариф на електричну енергію, грн./кВт·година;

$Ф_{зал}$ – залишкова вартість ПК на поточний рік, грн.;

$Н_а$ – річна норма амортизації на ПК, частки одиниці;

$Н_{апз}$– річна норма амортизації на ліцензійне програмне забезпечення, частки одиниці;

$К_{лпз}$ – вартість ліцензійного програмного забезпечення, грн.;

$F_р$ – річний фонд робочого часу (за 40-годинного робочого тижня $F_р$ =1920).

$$С_{мч} = 0,7 \cdot 2 \cdot 4,32 + \frac{12500 \cdot 0.4}{1920} + \frac{6800 \cdot 0,2}{1920} = 9,36 \text{ грн.} \quad (3.7)$$

$$З_{мч} = 186 \cdot 9,36 = 1740,83 \text{ грн.}$$

$$К_{рп} = 37200 + 1740,83 = 38940,83 \text{ грн.}$$

Капітальні (фіксовані) витрати на проєктування та впровадження проєктного варіанта системи інформаційної безпеки складають:

$$К= К_{рп} + К_{аз} + К_{зпз} + К_{навч}, \quad (3.8)$$

де $К_{рп}$– вартість розробки проєкту інформаційної безпеки та залучення для цього зовнішніх консультантів, тис. грн;

$К_{зпз}$ – вартість закупівель ліцензійного основного й додаткового програмного забезпечення (ПЗ), тис. грн.;

$К_{аз}$ – вартість закупівлі апаратного забезпечення та допоміжних матеріалів, тис. грн;

$К_{навч}$ – витрати на навчання технічних фахівців і обслуговуючого персоналу, тис. грн;

$К_{зпз}$ включає в себе такі ПЗ, як антивірусні програми, системи IDS/IPS, фаєрволи, загальна сума яких складає 5000 грн.

$К_{аз}$ позначає вартість апаратного забезпечення, необхідного для реалізації проєкту. В нього входять: сервери для тренування моделі ШІ, мережеве обладнання, кабелі.

Витрати на навчання технічних фахівців та технічного обслуговуючого персоналу $К_{навч}$ складають 5000 грн та включають в себе: курси та тренінги з кібербезпеки, навчальні матеріали.

$$К = 38940,83 + 29900 + 6000 + 5000 = 79840,83 \text{ грн.}$$

3.3 Розрахунок поточних (експлуатаційних витрат)

Експлуатаційні витрати – це поточні витрати на експлуатацію та обслуговування об'єкта проєктування за визначений період (наприклад, рік), що

виражені у грошовій формі.

Річні поточні (експлуатаційні) витрати на функціонування системи інформаційної безпеки складають:

$$С = С_в + С_к + С_{ак}, \text{ грн.}, \quad (3.9)$$

де $С_в$ – вартість Upgrade-відновлення й модернізації системи ($С_в = 0$);

$С_к$ – витрати на керування системою в цілому;

$С_{ак}$ – витрати, викликані активністю користувачів системи інформаційної безпеки ($С_{ак} = 0$).

Витрати на керування системою інформаційної безпеки ($С_к$) складають:

$$С_к=С_н + С_а + С_з + С_{єв} + С_{ел} + С_о + С_{тос}, \text{ грн.}, \quad (3.10)$$

де $С_н$ – витрати на навчання адміністративного персоналу й кінцевих користувачів, 5000 грн.;

$С_а$ – річний фонд амортизаційних відрахувань.

Річний фонд заробітної плати інженерно-технічного персоналу, що обслуговує систему інформаційної безпеки ($С_з$), складає:

$$С_з = З_{осн} + З_{дод}, \text{ грн.}, \quad (3.11)$$

де $З_{осн}$ – основна заробітна плата, грн./рік;

$З_{дод}$ – додаткова заробітна плата, грн./рік.

Основна заробітна плата визначається, виходячи з місячного посадового окладу, а додаткова заробітна плата – в розмірі 8-10% від основної заробітної плати.

Основна заробітна плата спеціаліста з інформаційної безпеки становить 18000 грн. Додаткова заробітна плата – 8% від основної заробітної плати

виходячи з місячного посадового окладу, а додаткова заробітна плата – в розмірі 8-10% від основної заробітної плати.

Виконання робіт вимагає залучення спеціаліста з інформаційної безпеки на 0.2 ставки.

$$С_з = (18000 \cdot 12 + 18000 \cdot 12 \cdot 0,08) \cdot 0,2 = 46656 \text{ грн}$$

В 2024 ставка ЄСВ складає 22% від фонду заробітної плати.

$$С_{єв} = 46656 \cdot 0,22 = 10264 \text{ грн.}$$

Вартість електроенергії, що споживається апаратурою системою інформаційної безпеки протягом року ($С_{ел}$), визначається за формулою:

$$С_{ел}=Р \cdot F_р \cdot Ц_е, \text{ грн.}, \quad (3.12)$$

де Р – встановлена потужність апаратури інформаційної безпеки, 0.7 кВт;

$F_р$ – річний фонд робочого часу системи інформаційної безпеки ($F_р$ = 1920 год.);

$Ц_е$ – тариф на електроенергію, 4,32 грн/кВт·годин.

$$С_{ел}=0,7 \cdot 1920 \cdot 4,32 = 5806,08 \text{ грн.}$$

Витрати на технічне й організаційне адміністрування та сервіс системи інформаційної безпеки ($С_{тос}$) визначаються за даними організації або у відсотках від вартості капітальних витрат – 1%.

$$С_{тос}= 79840,83 \cdot 0,01=798,40 \text{ грн}$$

Витрати на залучення сторонніх організацій для виконання деяких видів обслуговування, навчання та сертифікацію обслуговуючого персоналу ($С_о$) визначаються за даними організації – $С_о$ = 0 грн.

Річний фонд амортизаційних відрахувань вираховується за формулою:

$$С_а = С_{а1} + С_{а2}, \text{ грн.}, \quad (3.13)$$

де $С_{а1}$ – річний фонд відрахувань ПЗ (програмного забезпечення);

$С_{а2}$ – річний фонд відрахувань АЗ (апаратного забезпечення).

$$C_{a1/2} = \frac{Ф_п}{T}, \text{ грн.}, \quad (3.14)$$

де $Ф_п$ – первісна вартість придбаного ПЗ/АЗ;

Т – мінімальний срок корисного використання (2 роки для ПЗ, 5 – для АЗ).

$Ф_п$ для ПЗ дорівнює 5000 грн. та включає в себе наступні компоненти: антивірусні програми, системи IDS/IPS, фаєрволи.

$Ф_п$ для АЗ складає 29900 грн. та містить в собі: сервери для тренування моделі ШІ, мережеве обладнання, кабелі.

$$C_{a1} = \frac{5000}{2} = 2500 \text{ грн.}$$

$$C_{a2} = \frac{29900}{5} = 5980 \text{ грн.}$$

$$C_a = 2500 + 5980 = 8480 \text{ грн.}$$

$$C_к = 5000 + 8480 + 46656 + 10264 + 5806,08 + 798,40 = 77004,48 \text{ грн.}$$

Витрати, викликані активністю користувачів системи інформаційної безпеки ($C_{ак}$) можна орієнтовно визначити, користуючись величною капітальних витрат та середньостатистичними даними про активність користувачів. За середньостатистичними даними активність користувачів складає 15%. Таким чином:

$$C_{ак} = 79840,83 \cdot 0,15 = 11976,12 \text{ грн.}$$

Таким чином, річні поточні (експлуатаційні) витрати на функціонування системи інформаційної безпеки складають:

$$C = 0 + 77004,48 + 11976,12 = 88980,60 \text{ грн.}$$

3.4 Оцінка можливого збитку від атаки на вузол або сегмент мережі

Для розрахунку вартості такого збитку можна застосувати наступну спрощену модель оцінки.

Необхідні вихідні дані для розрахунку:

$t_п$ – час простою вузла або сегмента корпоративної мережі внаслідок атаки, 2 години;

$t_в$ – час відновлення після атаки персоналом, що обслуговує корпоративну мережу, 4 год.;

$t_{ви}$ – час повторного введення загубленої інформації співробітниками атакованого вузла або сегмента корпоративної мережі, 3 год.;

$З_о$ – заробітна плата обслуговуючого персоналу (адміністраторів та ін.), 17000 грн./міс.;

$З_с$ – заробітна плата співробітників атакованого вузла або сегмента корпоративної мережі, 22000 грн./міс.;

$Ч_о$ – чисельність обслуговуючого персоналу (адміністраторів та ін.), осіб.;

$Ч_с$ – чисельність співробітників атакованого вузла або сегмента корпоративної мережі, 5 осіб.;

О – обсяг продажів атакованого вузла або сегмента корпоративної мережі, 200 тис. грн. у рік;

$П_{зч}$ – вартість заміни встаткування або запасних частин, 7385 грн.;

I – число атакованих вузлів або сегментів корпоративної мережі, 1;

N – середнє число атак на рік, 53.

Упущена вигода від простою атакованого вузла або сегмента корпоративної мережі становить:

$$U = П_п + П_в + V, \text{ грн.}, \quad (3.15)$$

де $П_п$ – оплачувані втрати робочого часу та простої співробітників атакованого вузла або сегмента корпоративної мережі, грн;

$П_в$ – вартість відновлення працездатності вузла або сегмента корпоративної мережі (переустановлення системи, зміна конфігурації та ін.), грн;

V – втрати від зниження обсягу продажів за час простою атакованого вузла або сегмента корпоративної мережі, грн.

Втрати від зниження продуктивності співробітників атакованого вузла або сегмента корпоративної мережі являють собою втрати їхньої заробітної плати (оплата непродуктивної праці) за час простою внаслідок атаки:

$$П_{п} = \frac{\sum З_{с}}{F} \cdot t_{п}, \text{ грн.}, \quad (3.16)$$

де F – місячний фонд робочого часу (при 40-а годинному робочому тижні становить 176 ч).

$$П_{п} = \frac{22000 \cdot 5}{176} \cdot 2 = 1250 \text{ грн.}$$

Витрати на відновлення працездатності вузла або сегмента корпоративної мережі включають кілька складових:

$$П_{в} = П_{ви} + П_{пв} + П_{зч} \quad (3.17)$$

де $П_{ви}$ – витрати на повторне уведення інформації, грн;

$П_{пв}$ – витрати на відновлення вузла або сегмента корпоративної мережі, грн;

$П_{зч}$ – вартість заміни устаткування або запасних частин, грн.

Витрати на повторне введення інформації $П_{ви}$ розраховуються виходячи з розміру заробітної плати співробітників атакованого вузла або сегмента корпоративної мережі $З_{с}$, які зайняті повторним введенням втраченої інформації, з урахуванням необхідного для цього часу $t_{ви}$.

$$П_{ви} = \frac{\sum З_{с}}{F} \cdot t_{ви} = \frac{22000 \cdot 5}{176} \cdot 3 = 1875 \text{ грн.} \quad (3.18)$$

Витрати на відновлення вузла або сегмента корпоративної мережі $П_{пв}$ визначаються часом відновлення після атаки $t_{в}$ і розміром середньогодинної заробітної плати обслуговуючого персоналу (адміністраторів):

$$П_{пв} = \frac{\sum З_о}{F} \cdot t_в = \frac{17000 \cdot 5}{176} \cdot 4 = 1931{,}82 \text{ грн.} \quad (3.19)$$

Тоді, витрати на відновлення працездатності вузла або сегмента корпоративної мережі складуть:

$$П_в = 1875 + 1931{,}82 + 7385 = 11191{,}82 \text{ грн.}$$

Втрати від зниження очікуваного обсягу продажів за час простою атакованого вузла або сегмента корпоративної мережі визначаються виходячи із середньогодинного обсягу продажів і сумарного часу простою атакованого вузла або сегмента корпоративної мережі:

$$V = \frac{O}{F_г} \cdot (t_п + t_в + t_{ви}), \text{ грн.,} \quad (3.20)$$

де $F_г$ – річний фонд часу роботи організації (52 робочих тижні, 5-ти денний робочий тиждень, 8-ми годинний робочий день) становить близько 2080 год.

$$V = \frac{200000}{2080} \cdot (2 + 4 + 3) = 865{,}38 \text{ грн.}$$

Таким чином, упущена вигода від простою атакованого вузла або сегмента корпоративної мережі складе:

$$U=1250 + 11191{,}82 + 865{,}38 = 13307{,}20 \text{ грн.}$$

Тоді, загальний збиток від атаки на вузол або сегмент корпоративної мережі організації складе:

$$В = \Sigma i + \Sigma n \cdot U = \Sigma 1 + \Sigma 53 \cdot 13307{,}20 = 705281{,}60 \text{ грн.} \quad (3.21)$$

Загальний ефект від впровадження системи інформаційної безпеки визначається з урахуванням ризиків порушення інформаційної безпеки і

становить:

$$E = B \cdot R - C, \quad (3.22)$$

де B – загальний збиток від атаки на вузол або сегмент корпоративної мережі, тис. грн;

R – очікувана імовірність атаки на вузол або сегмент корпоративної мережі, частки одиниці (30%);

C – щорічні витрати на експлуатацію системи інформаційної безпеки, тис. грн.

Таким чином, загальний ефект від впровадження системи інформаційної безпеки складе:

$$E = 705281{,}60 \cdot 0{,}3 - 88980{,}60 = 122603{,}88 \text{ грн.}$$

3.5 Визначення та аналіз показників економічної ефективності інформаційної безпеки

Коефіцієнт повернення інвестицій ROSI показує, скільки гривень додаткового прибутку приносить одна гривня капітальних інвестицій на впровадження системи інформаційної безпеки.

Щодо інформаційної безпеки говорять не про прибуток, а про запобігання можливих втрат від атаки на сегмент або вузол корпоративної мережі, а отже:

$$ROSI = \frac{E}{K}, \quad (3.23)$$

де E – загальний ефект від впровадження системи інформаційної безпеки, тис. грн;

K – капітальні інвестиції за варіантами, що забезпечили цей ефект, тис. грн.

$$ROSI = \frac{122603{,}88}{79840{,}83} = 1{,}5, \text{ частки одиниці}$$

Проєкт системи інформаційної безпеки визнається доцільним, якщо

розрахункове значення коефіцієнта повернення інвестицій перевищує величину банківської кредитної ставки з урахуванням інфляції:

$$ROSI > \frac{(N_{кр} - N_{інф})}{100}, \quad (3.24)$$

де $N_{кр}$ – банківська кредитна ставка, (15%);

$N_{інф}$ – річний рівень інфляції, (8,6%).

Розрахункове значення коефіцієнта повернення інвестицій:

$$1.01 > \frac{15 - 8,6}{100} = 1,5 > 0,06$$

Термін окупності капітальних інвестицій T_o показує, за скільки років капітальні інвестиції окупляться за рахунок загального ефекту від впровадження системи інформаційної безпеки:

$$T_o = \frac{К}{Е} = \frac{1}{ROSI} \quad (3.25)$$

Підставимо значення:

$$T_o = \frac{К}{Е} = \frac{1}{1,5} = 0,67 \text{ років (237 днів)}$$

3.6 Висновок

Отже, виходячи з зроблених розрахунків, можна зробити висновок, що розробка підходу до застосування технологій ШІ для виявлення та нейтралізації вебатак є економічно доцільною.

Капітальні витрати становлять 79840,83 грн., а загальний економічний ефект від впровадження складає 122603,88 грн. Згідно до отриманих значень показників економічної ефективності, запропонований підхід дозволить принести 1,5 прибутку на 1 гривню капітальних витрат. Таким чином, можна

сказати, що результат від впровадження захисту буде максимально ефективним та окупиться через 237 днів.

ВИСНОВОК

У першому розділі роботи було детально розглянуто основні типи вебатак. Також було досліджено поняття штучного інтелекту, та основні принципи його роботи. Наведена інформація демонструє, що існуючі засоби протидії вебатакам мають певні недоліки, серед яких ключовий – нездатність швидко адаптуватися під зміни, оскільки традиційні підходи засновані на роботу по певним сценаріям. Цими сценаріями являються найбільш розповсюджені види атак, але з кожним разом їх стає все більше, та протидіяти їм є нелегкою задачею навіть для найбільш просунутих технологій. Тому, засоби захисту на основі ШІ могли б вирішити цю проблему, пропонуючи нові підходи та методи.

У другому розділі роботи були представлені підходи до виявлення та нейтралізації вебатак, а також оцінка їх ефективності. Зокрема було розглянуто засоби виявлення мережевих аномалій за допомогою алгоритмів класифікації. По результатам можна зробити висновок, що вони показують доволі високу точність. Наприклад, алгоритм Decision Tree показав 82% точності. Також, точність можна підвищити за допомогою більш якісних даних навчання моделі. Для нейтралізації вебатак існують різні рішення WAF з вбудованим ШІ, де використання кожного з них залежить від бюджету, також цілі захисту. Загалом, застосування технологій штучного інтелекту для захисту вебдодатків показало, що вони мають можливість швидко адаптуватися під нові загрози, не вимагаючи активної участі людини.

В економічному розділі було встановлено економічну доцільність застосування технологій штучного інтелекту для виявлення та нейтралізації вебатак. Результати проробленої роботи показали, що впровадження даних заходів дає економічний ефект у розмірі 122603,88 грн, а капітальні витрати у розмірі 79840,83 грн. При цьому , а термін окупності є невеликим – 237 днів, при коефіцієнті повернення ROSI = 1,5 грн.

ПЕРЕЛІК ПОСИЛАНЬ

1. Bootstrap aggregation. URL: https://medium.com/@hemaanusha tangellamudi/bootstrapped-aggregation-bagging-481f4812e3ea (дата звернення 18.06.2024);

2. Статистичні дані атак на основі шкідливого ПЗ. URL: https://www.statista.com/statistics/873097/malware-attacks-per-year-worldwide/ (дата звернення 28.05.2024);

3. Виявлення мережевих вторгнень на основі аномалій з використанням SVM. URL: https://github.com/DuseTrive/Anomaly-Based-NID-using-svm (дата звернення 24.05.2024);

4. Розуміння логістичної регресії. URL: https://www.spiceworks.com/ tech/artificial-intelligence/articles/what-is-logistic-regression/ (дата звернення 04.06.2024);

5. Розуміння атак SQL-ін'єкцій і способів їх запобігання. URL: https://medium.com/@craftingcode/understanding-sql-injection-attacks-and-how-to-prevent-them-8750ff1384fd (дата звернення 21.05.2024);

6. Що таке WAF. URL: https://www.radware.com/cyberpedia/application-security/what-is-waf/ (дата звернення 15.06.2024);

7. Alan Turing. Computing Machinery and Intelligence. Oxford, 1950. С. – 433-460.

8. Топ 10 найбільших ризиків для безпеки вебдодатків. URL: https://owasp.org/www-project-top-ten/ (дата звернення 24.05.2024);

9. Статистичні дані по кількість мобільних користувачів у всьому світі. URL: https://explodingtopics.com/blog/smartphone-stats (дата звернення 20.05.2024);

10. Random Forest. URL: http://surl.li/usqjj. (дата звернення 27.05.2024);

11. DBSCAN кластеризація. URL: https://towardsdatascience.com/dbscan-make-density-based-clusters-by-hand-2689dc335120 (дата звернення 17.06.2024);

12. AWS WAF. URL: https://aws.amazon.com/waf/ (дата звернення

26.05.2024);

13. Огляд поняття машинне навчання. URL: https://www.ibm.com/topics/machine-learning (дата звернення 11.06.2024);

14. Розгляд комп'ютерного зору. URL: https://www.v7labs.com/blog/what-is-computer-vision (дата звернення 15.06.2024);

15. SQL-ін'єкції. URL: https://systemweakness.com/sql-injection-ec671dfac4f8 (дата звернення 27.05.2024);

16. Cross Site Scripting (XSS). URL: https://owasp.org/www-community/attacks/xss/ (дата звернення 02.06.2024);

17. Cross site request forgery (CSRF). URL: https://www.imperva.com/learn/application-security/csrf-cross-site-request-forgery/ (дата звернення 03.06.2024);

18. Атаки на основі шкідливого ПЗ. URL: https://www.kiteworks.com/risk-compliance-glossary/malware-based-attacks/. (дата звернення 08.06.2024);

19. IDS/IPS системи. URL: https://www.spiceworks.com/it-security/network-security/articles/ids-vs-ips/. (дата звернення 30.05.2024);

20. Що таке NLP? URL: https://www.ibm.com/topics/natural-language-processing. (дата звернення 18.06.2024);

21. Cloudfare WAF. URL: https://www.cloudflare.com/lp/ppc/waf-x/. (дата звернення 08.09.2024);

22. Azure WAF. URL: https://www.radware.com/lp/web-application-and-api-protection/. (дата звернення 13.06.2024);

23. F5 Advanced Framework. URL: https://www.nomios.com/partners/f5-networks/security/advanced-waf/. (дата звернення 10.06.2024);

24. Методичні вказівки до виконання економічної частини дипломного проєкту зі спеціальності 125 Кібербезпека /: Упорядн.: Д.П. Пілова – Дніпро: Національний технічний університет «Дніпровська політехніка», 2019. – 16 с.

ДОДАТОК А. Відомість матеріалів кваліфікаційної роботи

№	Формат	Найменування	Кількість листів	Примітка
1	А4	Реферат	1	
2	А4	Список умовних скорочень	2	
3	А4	Зміст	2	
4	А4	Вступ	2	
5	А4	1 Розділ	17	
6	А4	2 Розділ	21	
7	А4	3 Розділ	14	
8	А4	Висновок	1	
9	А4	Перелік посилань	2	
10	А4	Додаток А	1	
11	А4	Додаток Б	1	
12	А4	Додаток В	1	
13	А4	Додаток Г	1	

ДОДАТОК Б. Перелік документів на оптичному носії

Шалагінов_ПЗ.docx

Шалагінов_ПЗ.pdf

Шалагінов_ДМ.pptx

ДОДАТОК В. Відгук керівника економічного розділу

Економічний розділ виконаний відповідно до вимог, які ставляться до кваліфікаційних робіт, та заслуговує на оцінку ____ б. («_____»).

Керівник розділу _____ доц. Пілова Д.П.

(підпис) (ініціали, прізвище)

ДОДАТОК Г. ВІДГУК

на кваліфікаційну роботу бакалавра на тему:

студента групи 125-20-1
Шалагінова Іллі Олеговича

Пояснювальна записка складається з титульного аркуша, завдання, реферату, списку умовних скорочень, змісту, вступу, трьох розділів, висновків, переліку посилань та додатків, розташованих на 69 сторінках та містить 15 рисунків, 24 джерела та 4 додатка.

Об'єктом розробки є технології штучного інтелекту для виявлення та нейтралізації вебатак.

Предметом розробки є засоби застосування технологій штучного інтелекту для виявлення та нейтралізації вебатак.

Метою роботи є підвищення рівня захисту інформаційної системи за рахунок впровадження технологій штучного інтелекту для виявлення та нейтралізації вебатак.

У першому розділі розглянуто класифікацію, методи здійснення вебатак, їхню природу, механізми дії та вплив на безпеку інформаційних систем. Аналізуючи атаки, було визначено їх типи та механізми, а також способи впливу на системи та потенційні небезпеки.

У другому розділі досліджується використання штучного інтелекту для виявлення та нейтралізації вебатак. Показано, як ШІ допомагає системам адаптуватися до нових загроз та виявляти відомі атаки через самонавчання.

Студент показав достатній рівень володіння теоретичними положеннями з обраної теми, показав здатність формувати власну точку зору (теоретичну позицію).

Робота оформлена та написана грамотною мовою, відповідає вимогам положення про систему запобігання та виявлення плагіату у Національному технічному університеті «Дніпровська політехніка». Містить необхідний ілюстрований матеріал. Автор добре знає проблему, уміє формулювати наукові та практичні завдання і знаходить адекватні засоби для їх вирішення.

В цілому робота задовольняє усім вимогам і може бути допущена до захисту, а його автор заслуговує на оцінку «_____».

Керівник Максим ТКАЧ