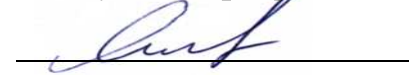


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій

"Допущений до захисту"

Завідувач кафедри ІУСТ



к.т.н, доц. Михелєв І.Л.

"22" червня 2026 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти "бакалавр"

на тему: **«Розроблення інформаційної системи виявлення фішингових
сайтів та підозрілих листів на основі ML»**

Виконав: студент групи 4144



Немченко Р. Д.
(підпис)

Керівник роботи:

СТ. ВИКЛАДАЧ

(посада, науковий ступінь вчене звання)



Баковська І.В.
(підпис)

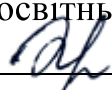
м. Миколаїв – 2026 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій
Спеціальність 126 "Інформаційні системи та технології"
Освітня програма "Інформаційні системи та технології"

"ЗАТВЕРДЖУЮ"

Гарант освітньої програми


к.т.н, доц. Кнirik Н.Р.

"25" березня 2026 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття ступеня вищої освіти "бакалавр"

Студенту Немченко Роману Денисовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Розроблення інформаційної системи виявлення фішингових сайтів та підозрілих листів на основі ML

Керівник роботи ст. викладач Баковська І.В

Затверджені наказом ректора № 270-уч від "31" березня 2026 року.

2. Термін подання роботи: 22 червня 2026 р.

3. Вихідні дані до проекту (роботи) ДСТУ щодо обробки інформації, літературні джерела, технічна документація на існуючі аналоги систем.

4. Перелік питань, що належать до розробки (найменування розділів) Аналіз предметної області та постановка задачі. Проектування системи виявлення фішингових сайтів та підозрілих листів на основі ML. Розробка системи виявлення фішингових сайтів та підозрілих листів на основі ML. Охорона праці.

5. Перелік презентаційних матеріалів Актуальність, концепція системи, Порівняльний аналіз систем, Діаграма Use Case, Діаграма Послідовності, ER-діаграми інформаційної бази даних, Структура технічних засобів інформаційної системи, Екранні форми розробки, Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Баковська І.В.	02.02.2026	23.02.2026
2	Баковська І.В.	23.02.2026	20.03.2026
3	Баковська І.В.	20.03.2026	24.04.2026
4	Баковська І.В.	24.04.2026	01.06.2026

7. Дата видачі завдання 02 лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

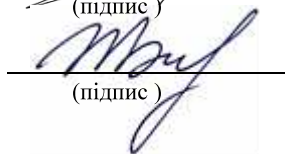
№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Аналіз предметної галузі	02.02.2026	
2	Постановка задачі	16.02.2026	
3	Розробка концепції	23.02.2026	
4	Розробка проекту ІС	23.03.2026	
5	Реалізація проекту	27.04.2026	
6	Розробка питань з охорони праці	01.06.2026	
7	Розробка графічних матеріалів	05.06.2026	
8	Подання роботи на перевірку на плагіат	08.06.2026	
9	Подання роботи рецензенту	15.06.2026	
10	Подання роботи на захист	22.06.2026	

Студент


(підпис)

Немченко Р.Д.
(ПІБ)

Керівник роботи


(підпис)

Баковська І.В.
(ПІБ)

АНОТАЦІЯ

Бакалаврська робота присвячена розробленню інформаційної системи виявлення фішингових сайтів та підозрілих електронних листів на основі методів машинного навчання. Актуальність теми обумовлена постійним зростанням кількості кіберзагроз, пов'язаних із фішинговими атаками, які спрямовані на отримання конфіденційної інформації користувачів, фінансових даних та облікових записів.

У роботі проведено аналіз сучасних методів виявлення фішингових ресурсів та шкідливих повідомлень, досліджено особливості застосування алгоритмів машинного навчання для класифікації вебресурсів і електронних листів. Розглянуто існуючі підходи до аналізу URL-адрес, вмісту вебсторінок та текстової інформації електронних повідомлень.

Запропоновано концепцію інформаційної системи, яка забезпечує автоматизоване виявлення потенційно небезпечних вебсайтів та підозрілих електронних листів шляхом аналізу набору ознак та використання моделей машинного навчання. Виконано проєктування архітектури системи, розроблено структуру бази даних, моделі взаємодії компонентів та алгоритми обробки інформації.

Результатом роботи є проєкт інформаційної системи, здатної здійснювати класифікацію вебресурсів і електронних повідомлень за рівнем загрози, формувати попередження для користувачів та підвищувати рівень інформаційної безпеки. Запропоноване рішення може бути використане в корпоративних мережах, освітніх установах та організаціях для протидії фішинговим атакам і зменшення ризиків компрометації даних.

Робота складається з 88 аркушів машинописного тексту, 12 рисунків, 7 таблиць, 11 літературних джерел, трьох додатків.

Ключові слова: інформаційна система, фішинг, фішинговий сайт, підозрілий електронний лист, машинне навчання, кібербезпека, класифікація даних, аналіз URL, інформаційна безпека, виявлення загроз.

ABSTRACT

The bachelor's thesis is devoted to the development of an information system for detecting phishing sites and suspicious emails based on machine learning methods. The relevance of the topic is due to the constant increase in the number of cyber threats associated with phishing attacks, which are aimed at obtaining confidential user information, financial data and account records.

The paper analyzes modern methods for detecting phishing resources and malicious messages, investigates the features of using machine learning algorithms for classifying web resources and emails. Existing approaches to analyzing URL addresses, web page content and text information of electronic messages are considered.

The concept of an information system is proposed, which provides automated detection of potentially dangerous websites and suspicious emails by analyzing a set of features and using machine learning models. The system architecture is designed, the database structure, component interaction models and information processing algorithms are developed.

The result of the work is a project of an information system capable of classifying web resources and electronic messages by threat level, generating warnings for users and increasing the level of information security. The proposed solution can be used in corporate networks, educational institutions and organizations to counteract phishing attacks and reduce the risks of data compromise.

The work consists of 88 sheets of typewritten text, 12 figures, 7 tables, 11 literary sources, and three appendices.

Keywords: information system, phishing, phishing site, suspicious email, machine learning, cybersecurity, data classification, URL analysis, information security, threat detection.

ЗМІСТ

АНОТАЦІЯ	4
ABSTRACT	5
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ	13
1.1 Аналіз предметної області	13
1.2 Недоліки в роботі	16
1.2.1 Недоліки традиційних методів виявлення фішингових сайтів	16
1.2.2 Недоліки виявлення підозрілих листів у сучасних поштових системах	18
1.2.3 Загальні проблеми існуючих систем захисту	19
1.2.4 Обґрунтування необхідності використання ML	19
1.3 Концепція системи	20
1.3.1 Обґрунтування вибору найкращої концепції	22
1.4 Математичне забезпечення розробки	23
1.4.2 Логістична регресія	24
1.4.3 Наївний байєсівський класифікатор (Naive Bayes)	24
1.4.4 TF-IDF для представлення тексту листів	25
1.4.5 Метрики оцінювання якості моделі	25
1.5 Постановка задачі	26
Висновки до розділу 1	29
РОЗДІЛ 2. ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ВІЯВЛЕННЯ ФІШИНГОВИХ САЙТІВ ТА ПІДОЗРІЛИХ ЛИСТІВ НА ОСНОВІ ML	31
2.1 Загальносистемні рішення	31
2.1.1 Призначення системи	32
2.1.2 Основні функції системи	32
2.1.3 Категорії користувачів та їх права доступу	32
2.1.4 Структура системи та основні модулі	33

2.1.5 Загальна логіка роботи системи	35
2.1.6 Вимоги до інформаційних потоків.....	35
2.1.7 Вимоги до масштабованості та розширюваності	35
2.2 Архітектура системи.....	36
2.2.1 Загальна структура архітектури	36
2.2.2 Рівень представлення	37
2.2.3 Серверний рівень (Application Layer)	37
2.2.4 Модуль попередньої обробки даних.....	38
2.2.5 Рівень машинного навчання ML Layer	38
2.2.6 Рівень зберігання даних	39
2.2.7 Взаємодія компонентів системи.....	40
2.2.8 Переваги обраної архітектури	40
2.2.9 Безпека системи	41
2.3 Взаємодії користувача із системою	42
2.4 Розробка бази даних системи	43
2.4.1 Основні вимоги до бази даних	44
2.4.2 Аналіз інформаційних потоків	44
2.4.3 Основні сутності бази даних.....	44
2.4.4 Логічна структура бази даних	47
2.4.5 Нормалізація бази даних	49
2.4.6 Забезпечення безпеки даних.....	49
2.4.7 Індексація та оптимізація.....	49
2.4.8 Масштабованість бази даних.....	50
Висновки до розділу 2	50
РОЗДІЛ 3. РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ВИЯВЛЕННЯ	
ФІШИНГОВИХ САЙТІВ ТА ПІДОЗРІЛИХ ЛИСТІВ НА ОСНОВІ ML	53
3.1 Вимоги до програмного забезпечення.....	53
3.1.1 Вимоги до операційної системи	53
3.1.2 Вимоги до серверного програмного забезпечення.....	54
3.1.3 Вимоги до програмного забезпечення машинного навчання	54

3.1.4	Вимоги до клієнтського програмного забезпечення	55
3.1.5	Вимоги до системи управління базами даних	56
3.1.6	Вимоги до мережевого програмного забезпечення	56
3.1.7	Вимоги до засобів розгортання та контейнеризації	57
3.1.8	Вимоги до безпеки програмного забезпечення	57
3.1.9	Вимоги до продуктивності системи	58
3.2	Вимоги до апаратного забезпечення	58
3.2.1	Вимоги до серверного обладнання	59
3.2.2	Вимоги до апаратного забезпечення ML-модуля	60
3.2.3	Вимоги до клієнтських робочих станцій	60
3.2.4	Вимоги до мережевого обладнання	61
3.2.5	Вимоги до систем резервного копіювання	61
3.2.6	Вимоги до масштабованості апаратної платформи	61
3.2.7	Вимоги до енергоефективності та надійності	62
3.3	Розгортання системи	62
3.3.1	Загальна схема розгортання системи	63
3.3.2	Клієнтський рівень	64
3.3.3	Web-сервер	64
3.3.4	Сервер прикладної логіки	64
3.3.5	ML-сервер	65
3.3.6	Сервер бази даних	65
3.3.7	Контейнеризація та хмарне розгортання	65
3.3.8	Забезпечення безпеки під час розгортання	66
3.3.9	Масштабованість системи	66
3.3.10	Переваги обраної моделі розгортання	66
	Висновки до розділу 3	67
РОЗДІЛ 4. ОХОРОНА ПРАЦІ		69
4.1	Загальні положення	69
4.2	Аналіз умов праці під час роботи з інформаційною системою	69
4.3	Вимоги до організації робочого місця	70

4.4 Вимоги до мікроклімату приміщення.....	70
4.5 Освітлення робочого місця	70
4.6 Електробезпека.....	71
4.7 Пожежна безпека	71
4.8 Режим праці та відпочинку	71
4.9 Заходи щодо покращення умов праці.....	72
Висновки до розділу 4	72
ВИСНОВКИ.....	73
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	75
ДОДАТОК А. ТЕХНІЧНЕ ЗАВДАННЯ НА РОЗРОБКУ ІНФОРМАЦІЙНОЇ СИСТЕМИ ВИЯВЛЕННЯ ФІШИНГОВИХ САЙТІВ ТА ПІДОЗРІЛИХ ЛИСТІВ НА ОСНОВІ ML	76
ДОДАТОК Б. ІНСТРУКЦІЯ КОРИСТУВАЧА ІНФОРМАЦІЙНОЇ СИСТЕМИ ВИЯВЛЕННЯ ФІШИНГОВИХ САЙТІВ ТА ПІДОЗРІЛИХ ЛИСТІВ НА ОСНОВІ ML	82
ДОДАТОК В. ЕКРАННІ ФОРМИ РОЗРОБКИ	88

ВСТУП

У сучасному цифровому суспільстві інформаційні технології стали невід'ємною складовою діяльності підприємств, державних установ та окремих користувачів. Разом із розвитком інформаційних систем і збільшенням обсягів електронного обміну даними зростає кількість кіберзагроз, серед яких особливе місце займають фішингові атаки. Фішинг є одним із найпоширеніших видів соціальної інженерії, що передбачає використання підроблених вебсайтів та електронних повідомлень для отримання конфіденційної інформації користувачів, зокрема логінів, паролів, банківських реквізитів та інших персональних даних.

Зловмисники постійно вдосконалюють методи створення фішингових ресурсів, що значно ускладнює їх виявлення традиційними засобами захисту. Сучасні фішингові сайти часто майже повністю копіюють зовнішній вигляд легітимних вебресурсів, а підозрілі електронні листи використовують складні методи маскування та персоналізації. У зв'язку з цим виникає необхідність розроблення інтелектуальних систем, здатних автоматично аналізувати характеристики вебресурсів та електронних повідомлень, виявляючи потенційні загрози на ранніх етапах.

Одним із перспективних напрямів вирішення даної проблеми є використання технологій машинного навчання. Алгоритми машинного навчання дозволяють аналізувати великі обсяги даних, виявляти приховані закономірності та здійснювати класифікацію об'єктів за визначеними ознаками. Застосування таких методів відкриває можливість створення ефективних систем автоматичного виявлення фішингових сайтів та підозрілих електронних листів із високою точністю прогнозування.

Актуальність теми бакалаврської роботи обумовлена необхідністю підвищення рівня інформаційної безпеки користувачів шляхом впровадження сучасних методів аналізу даних та машинного навчання для своєчасного виявлення кіберзагроз.

Мета роботи полягає у розробленні інформаційної системи виявлення фішингових сайтів та підозрілих листів на основі методів машинного навчання для підвищення ефективності захисту користувачів від кіберзагроз.

Для досягнення поставленої мети необхідно вирішити такі завдання:

1. Провести аналіз предметної області та сучасних методів виявлення фішингових атак.
2. Дослідити особливості функціонування фішингових вебсайтів та підозрілих електронних повідомлень.
3. Виконати аналіз алгоритмів машинного навчання, придатних для задач класифікації загроз.
4. Розробити концепцію інформаційної системи.
5. Спроектувати архітектуру програмного забезпечення та структуру бази даних.
6. Розробити математичне забезпечення системи.
7. Створити моделі взаємодії компонентів та UML-діаграми.
8. Реалізувати програмні модулі аналізу вебресурсів та електронних листів.
9. Оцінити ефективність запропонованого рішення.

Об'єктом дослідження є процеси виявлення та класифікації фішингових вебсайтів і підозрілих електронних повідомлень.

Предметом дослідження є методи та алгоритми машинного навчання, що використовуються для аналізу вебресурсів і електронних листів з метою виявлення ознак фішингу.

Методи дослідження включають методи системного аналізу, машинного навчання, класифікації даних, статистичного аналізу, проєктування інформаційних систем, моделювання бізнес-процесів та об'єктно-орієнтованого проєктування.

Практична цінність роботи полягає у створенні інформаційної системи, яка дозволяє автоматизувати процес виявлення потенційно небезпечних

вебсайтів та електронних повідомлень, підвищуючи рівень кібербезпеки користувачів та знижуючи ризики компрометації конфіденційної інформації.

Результати роботи можуть бути використані в організаціях різного профілю, навчальних закладах, корпоративних мережах та інших інформаційних системах, де існує потреба у своєчасному виявленні та попередженні фішингових атак.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної області

У сучасному цифровому середовищі інформаційні системи та мережеві технології активно використовуються в усіх сферах діяльності: банківській системі, електронній комерції, освіті, державному управлінні та корпоративному секторі. Разом із розвитком онлайн-сервісів зростає й кількість кіберзагроз, серед яких однією з найпоширеніших є *фішинг*. Фішингові атаки спрямовані на отримання конфіденційної інформації користувачів шляхом підробки вебсайтів або надсилання підозрілих електронних листів, які імітують повідомлення від легітимних організацій.

Фішинг є серйозною проблемою інформаційної безпеки, оскільки дозволяє зловмисникам отримувати логіни, паролі, банківські реквізити, персональні дані та доступ до корпоративних ресурсів. Внаслідок цього підприємства та користувачі зазнають фінансових втрат, витоку інформації та порушення безперервності бізнес-процесів. Тому актуальним є створення інформаційних систем, здатних автоматично виявляти фішингові ресурси та повідомлення, забезпечуючи попередження користувача про небезпеку.

Фішинг — це вид соціальної інженерії, що базується на маніпуляції користувачем з метою змусити його виконати небезпечні дії, а саме перейти за шкідливим посиланням, завантажити вкладення або ввести конфіденційні дані на підробленому сайті. Фішингові атаки можуть здійснюватися через електронну пошту email phishing, SMS-повідомлення smishing, телефонні дзвінки vishing, соціальні мережі та месенджери, підроблені вебсайти та рекламні посилання.

Найбільш поширеними є фішингові листи та фішингові вебсайти, які можуть мати візуальну схожість із реальними сервісами банки, служби доставки, державні портали, онлайн-магазини. Вони часто містять елементи

психологічного впливу таких як термінові повідомлення, погрози блокування акаунту, вигаш призду або необхідність підтвердження даних.

Фішинговий сайт — це вебресурс, створений з метою введення користувача в оману та викрадення його даних. Основними ознаками фішингових сайтів можуть бути використання доменних імен, схожих на оригінальні typosquatting. Наявність підозрілих піддоменів або довгих URL. Відсутність SSL-сертифікату або використання підробленого сертифікату. Використання IP-адреси замість доменного імені. Перенаправлення користувача через декілька сайтів. Копіювання дизайну офіційного сайту. Запити на введення конфіденційної інформації.

Виявлення таких ресурсів вручну є складним, оскільки зловмисники постійно вдосконалюють методи маскування.

Підозрілий електронний лист — це повідомлення, яке може містити фішингове посилання або вкладення зі шкідливим програмним забезпеченням. Основними характеристиками таких листів є невідповідність адреси відправника офіційному домену наявність граматичних помилок або дивного стилю написання, використання термінових закликів «терміново підтвердіть», «ваш акаунт буде заблоковано». Наявність підозрілих вкладень .exe, .zip, .js, .docm. Посилання на сторонні ресурси або скорочені URL. Використання підроблених логотипів компаній.

Фішингові листи часто використовуються для проникнення в корпоративні мережі, викрадення доступів до сервісів або розповсюдження вірусів.

Традиційні методи виявлення фішингових сайтів і листів включають:

- чорні списки — бази відомих фішингових доменів;
- фільтрацію за правилами — аналіз структури URL або вмісту листа за заздалегідь визначеними шаблонами;
- аналіз репутації домену — перевірка віку домену, країни реєстрації, DNS-записів;
- антивірусні системи та поштові фільтри.

Однак ці методи мають суттєві недоліки: вони часто не здатні виявляти нові, раніше невідомі фішингові атаки, оскільки залежать від оновлення баз або ручного введення правил.

У сучасних умовах ефективним підходом є використання методів машинного навчання Machine Learning, які дозволяють автоматично класифікувати сайти та листи як безпечні або фішингові. ML-підхід базується на аналізі великої кількості характеристик, таких як довжина URL, кількість піддоменів, наявність спеціальних символів @, -, цифри. Використання HTTPS, вік домену, частота використання ключових слів у тексті, наявність вкладень, статистичні параметри листа кількість посилань, слів, HTML-тегів. Лінгвістичні особливості повідомлення.

Моделі машинного навчання можуть навчатися на прикладах реальних фішингових та легітимних ресурсів. Найчастіше застосовуються алгоритми Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, Gradient Boosting, Neural Networks.

Використання ML дозволяє підвищити точність виявлення фішингових атак та зменшити кількість помилкових спрацювань.

В умовах зростання кількості кіберзагроз виникає необхідність створення спеціалізованої інформаційної системи, яка забезпечуватиме автоматичне виявлення фішингових сайтів за URL-адресою. Аналіз тексту електронних листів на наявність ознак фішингу. Перевірку вкладень та посилань, формування повідомлень для користувача про потенційну загрозу. Можливість навчання моделі на нових даних, ведення журналу перевірок та статистики.

Така система може використовуватися як для персонального захисту користувача, так і для корпоративного середовища наприклад, у навчальних закладах, банках або компаніях.

Таким чином, предметна область даної бакалаврської роботи охоплює проблематику інформаційної безпеки, зокрема виявлення фішингових вебсайтів та підозрілих електронних листів. Фішинг є актуальною загрозою

через постійне зростання кількості атак та їхню складність. Використання методів машинного навчання дозволяє автоматизувати процес виявлення таких загроз, підвищити точність класифікації та забезпечити ефективний захист користувачів і інформаційних систем. Розробка інформаційної системи виявлення фішингу є перспективним напрямом, що має практичну цінність та може бути використаний у різних сферах діяльності.

1.2 Недоліки в роботі

Фішингові атаки є однією з найпоширеніших загроз у сучасному цифровому середовищі. Їх метою є отримання конфіденційної інформації користувача шляхом підробки вебресурсів або надсилання підозрілих електронних листів, що містять шкідливі посилання чи вкладення. Незважаючи на значну кількість наявних засобів захисту (антивірусів, поштових фільтрів, браузерних механізмів безпеки, репутаційних сервісів), проблема виявлення фішингових сайтів та листів залишається актуальною. Це пояснюється тим, що існуючі системи мають низку суттєвих недоліків, які обмежують їх ефективність у реальних умовах.

1.2.1 Недоліки традиційних методів виявлення фішингових сайтів

1) Залежність від чорних списків

Багато сучасних систем кібербезпеки використовують чорні списки доменів та URL-адрес, які вже були визначені як небезпечні. Даний підхід є досить поширеним, оскільки дозволяє швидко блокувати відомі фішингові ресурси. Проте його головним недоліком є те, що він не забезпечує захист від нових сайтів, які ще не були виявлені.

Крім того, фішингові домени часто існують дуже короткий час (від кількох годин до кількох днів), після чого видаляються або змінюються. Через це навіть швидке оновлення баз даних не завжди дозволяє забезпечити своєчасний захист.

Основними недоліками є неможливість виявлення нових фішингових ресурсів zero-day phishing. Залежність від сторонніх сервісів репутації,

затримка між появою фішингового ресурсу та внесенням його до списку. Можливість обходу блокувань через використання нових доменів або коротких URL.

2) Використання схожих доменів та піддоменів

Фішингові сайти часто використовують доменні імена, які візуально схожі на справжні (наприклад, використання заміни букв, символів або схожих доменних зон). Це дозволяє вводити користувача в оману навіть при наявності базових систем захисту.

Недоліки існуючих систем полягають в складності автоматичного виявлення доменів-двійників, недостатньому аналізі структури URL, низької ефективності проти атак із використанням міжнародних доменів та схожих символів.

3) Недостатній аналіз контенту вебсторінки

Багато систем обмежуються лише аналізом URL-адреси, не враховуючи реальний вміст сторінки. Проте фішинговий сайт може мати цілком легітимний URL, але містити підроблену форму входу або небезпечні скрипти.

Недоліками є відсутність аналізу HTML-коду, скриптів і форм введення. Нemoжливість оцінки схожості сторінки до офіційного ресурсу. Слабка здатність виявляти складні фішингові сторінки з динамічним контентом.

4) Недостатня адаптивність до змін у фішингових атаках

Зловмисники постійно змінюють методи атак, застосовуючи редиректи, короткі посилання, динамічні домени, автоматичне генерування сторінок. Статичні методи аналізу не здатні швидко адаптуватися до таких змін.

Недоліками є потреба в ручному оновленні правил, низька точність при появі нових видів атак залежність від експертного втручання.

1.2.2 Недоліки виявлення підозрілих листів у сучасних поштових системах

1) Проблема соціальної інженерії

Фішингові листи зазвичай містять психологічний вплив: термінові повідомлення, залякування, обіцянки виграшу, повідомлення про блокування акаунту тощо. Більшість існуючих систем безпеки аналізують лише технічні характеристики листа, але не завжди здатні правильно інтерпретувати зміст повідомлення.

Як недоліки можна вказати слабкий семантичний аналіз тексту листа. Складність розпізнавання маніпулятивних фраз. Залежність від мови повідомлення.

2) Використання підроблених адрес відправника

Зловмисники можуть використовувати технології підміни адреси (spoofing) або домени, схожі на офіційні. У багатьох випадках користувач бачить на перший погляд легітимну адресу, що збільшує ймовірність переходу за шкідливим посиланням.

Недоліки це недостатній аналіз заголовків листа email headers. Обмежена перевірка SPF/DKIM/DMARC, складність автоматичного визначення підробки відправника.

3) Складність аналізу вкладень

Підозрілі листи можуть містити вкладення з шкідливим програмним забезпеченням або документами з макросами (наприклад, .docm, .xlsm). Частина поштових сервісів блокує такі вкладення, однак зловмисники використовують архіви або вкладення у вигляді посилань на хмарні сервіси.

Недоліки в недостатньому аналізі вкладень у форматах архівів, можливість прихованого шкідливого коду в документах, складність перевірки вкладень без використання sandbox-середовища.

4) Використання HTML та прихованих посилань

Фішингові листи часто містять HTML-розмітку, що дозволяє приховати справжнє посилання за красивою кнопкою або текстом. При

цьому користувач бачить одне посилання, але переходить на зовсім інший ресурс.

Недоліки в складності аналізу HTML-коду листа, можливість приховування URL через редиректи, обхід стандартних фільтрів.

1.2.3 Загальні проблеми існуючих систем захисту

1) Високий рівень помилкових спрацювань

Багато систем фільтрації видають значну кількість помилкових спрацювань (false positive), коли безпечні сайти або листи визначаються як фішингові. Це знижує довіру користувачів до системи та може призвести до її ігнорування.

2) Низький рівень виявлення прихованих атак

Деякі фішингові атаки використовують багатоступеневі редиректи, приховані JavaScript-скрипти або тимчасові домени. Такі методи ускладнюють виявлення загрози стандартними засобами.

3) Відсутність комплексного підходу

Часто існуючі рішення орієнтовані або на аналіз URL, або на аналіз листів, але не забезпечують інтегрованого підходу. В результаті частина загроз залишається непоміченою.

4) Відсутність автоматичного навчання та оновлення

Більшість класичних систем не підтримують механізм автоматичного оновлення логіки виявлення. Через це вони не здатні ефективно реагувати на нові методи атак без втручання спеціалістів.

1.2.4 Обґрунтування необхідності використання ML

З огляду на наведені недоліки можна зробити висновок, що традиційні методи захисту мають обмеження, особливо в умовах швидкої зміни фішингових атак. Саме тому доцільним є застосування методів машинного навчання, які дозволяють:

- аналізувати великі обсяги даних і виявляти приховані закономірності;

- автоматично класифікувати сайти та листи як безпечні або фішингові;
- адаптувати систему до нових типів атак шляхом перенавчання;
- зменшити кількість помилкових спрацювань;
- забезпечити комплексний підхід аналіз URL, контенту сайту, тексту листа, вкладень.

Таким чином, існуючі системи виявлення фішингових сайтів та підозрілих листів мають низку недоліків: залежність від чорних списків, недостатній аналіз контенту, складність виявлення нових загроз, проблеми з аналізом вкладень і HTML-листів, а також високу кількість помилкових спрацювань. Це підтверджує необхідність розроблення інформаційної системи нового покоління, яка базується на методах машинного навчання та забезпечує більш точне й адаптивне виявлення фішингових атак.

1.3 Концепція системи

З урахуванням виявлених недоліків існуючих рішень було розглянуто три можливі концепції побудови інформаційної системи виявлення фішингових сайтів і підозрілих листів, таблиця 1.1.

Варіант 1 – Система на основі чорних списків та репутаційних баз

Система виконує перевірку URL та адрес відправників за допомогою готових баз даних (blacklist, DNS reputation services). Вона визначає, чи присутній домен у списках небезпечних ресурсів.

Переваги у високій швидкості перевірки, простоті реалізації та інтеграції, мінімальних вимогах до обчислювальних ресурсів.

Недоліками є не виявляє нові фішингові сайти до внесення в blacklist, залежність від сторонніх сервісів, неможливість аналізу змісту листа чи поведінки сайту.

Варіант 2 – Rule-based система (перевірка за правилами та евристиками)

Система аналізує структуру URL та текст листа за набором правил: підозрілі слова, довжина URL, наявність спеціальних символів, вкладення, кількість посилань, HTML-розмітка тощо.

Перевагами є можливість працювати без великих наборів даних, легкість пояснення результатів перевірки, швидкість виконання аналізу.

Недоліками є висока залежність від якості правил, потреба постійного оновлення правил, низька ефективність проти сучасних адаптивних фішингових атак, велика кількість помилкових спрацювань.

Варіант 3 – Система на основі машинного навчання (ML-класифікація)

Система використовує алгоритми машинного навчання для класифікації сайтів і листів як безпечних або фішингових. Для цього аналізуються ознаки URL, метадані домену, вміст листа, кількість посилань, стилістика тексту тощо.

Переваги це висока точність класифікації при якісних даних, можливість виявлення нових атак zero-day phishing. Адаптивність та здатність до самонавчання. Можливість використання гібридного підходу URL + текст листа.

Недоліки в потребі у навчальних даних датасетах, складність реалізації та налаштування, потреба в ресурсах для навчання моделей, необхідність контролю якості даних.

№	Концепція системи	Опис	Переваги	Недоліки
1	Blacklist / репутаційні бази	Перевірка URL, доменів і адрес відправників за чорними списками та репутаційними сервісами	- висока швидкість перевірки- проста реалізація- мінімальні ресурси	- не виявляє нові загрози (zero-day)- залежність від оновлення баз- обмежена функціональність (без аналізу тексту листа)
2	Rule-based система (евристики)	Аналіз URL та листів за правилами: ключові слова, символи, структура URL, вкладення, кількість посилань	- швидка робота- зрозуміла логіка перевірки- не потребує навчальних даних	- потребує постійного оновлення правил- багато хибних спрацювань- слабка адаптація до нових атак
3	ML-система	Використання моделей ML для	- висока точність- здатність виявляти	- потрібні навчальні дані- складніша

№	Концепція системи	Опис	Переваги	Недоліки
	(машинне навчання)	класифікації сайтів і листів за набором ознак (URL, текст, метадані)	нові атаки-можливість самонавчання-комплексний аналіз (URL + текст)	реалізація- потребує ресурсів для навчання моделей

1.3.1 Обґрунтування вибору найкращої концепції

Порівняння трьох варіантів показало, що перші два підходи blacklist та rule-based мають переваги у простоті та швидкості реалізації, однак їх ефективність суттєво знижується в умовах появи нових фішингових сайтів та вдосконалення методів соціальної інженерії.

Найбільш перспективним є варіант 3 – система на основі машинного навчання, оскільки вона дозволяє аналізувати не лише домени, але й зміст листів. Виявляти нові фішингові ресурси навіть без внесення до blacklist;. Адаптуватися до змін у поведінці зловмисників, підвищувати точність за рахунок оновлення навчальної вибірки.

Крім того, ML-підхід дозволяє створити комплексну систему, яка може об'єднувати кілька методів аналізу URL, текстовий аналіз листа, перевірка вкладень, аналіз доменної інформації. Це забезпечує більш високий рівень захисту порівняно з традиційними рішеннями.

Таким чином, існуючі рішення з виявлення фішингових сайтів і підозрілих листів мають низку недоліків, зокрема залежність від чорних списків, необхідність ручного оновлення правил та недостатню адаптивність до нових загроз. Було розглянуто три варіанти концепції майбутньої системи: blacklist-підхід, rule-based фільтрація та ML-класифікація. На основі аналізу переваг і недоліків встановлено, що найкращим варіантом є застосування машинного навчання, оскільки воно забезпечує високу точність, здатність до самонавчання та ефективність проти сучасних фішингових атак.

1.4 Математичне забезпечення розробки

Математичне забезпечення інформаційної системи виявлення фішингових сайтів та підозрілих листів на основі методів машинного навчання включає сукупність математичних моделей, алгоритмів та методів, які застосовуються для аналізу даних, формування ознак та класифікації об'єктів. Основною метою математичного забезпечення є побудова моделі, яка здатна з високою точністю визначати, чи є вебсайт або електронний лист фішинговим.

Для розв'язання задачі виявлення фішингу використовується підхід бінарної класифікації, де кожному об'єкту URL або листу присвоюється один із двох класів:

- 1 – фішинговий Phishing
- 0 – легітимний Legitimate

Розробка ML-системи включає такі основні математичні етапи:

- формування вектора ознак;
- побудова моделі класифікації;
- оцінювання якості класифікатора.

1.4.1 Формування вектора ознак

Для машинного навчання кожен об'єкт (лист або сайт) представляється у вигляді вектора ознак:

$$X = (x_1, x_2, x_3, \dots, x_n)$$

де x_i — значення i -тої ознаки (довжина URL, кількість символів, наявність HTTPS, кількість посилань у листі тощо), а n — кількість ознак.

Клас об'єкта задається змінною:

$$y \in \{0, 1\}$$

1.4.2 Логістична регресія

Одним із базових методів класифікації є логістична регресія, яка обчислює ймовірність того, що об'єкт належить до класу фішингових.

Функція ймовірності задається сигмоїдною функцією:

$$P(y = 1|X) = \sigma(z) = \frac{1}{1 + e^{-z}}$$

де:

$$z = w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n$$

w_0 — зміщення (bias), w_i — вагові коефіцієнти ознак.

Рішення приймається за правилом:

$$\hat{y} = \begin{cases} 1, & P(y = 1|X) \geq 0.5 \\ 0, & P(y = 1|X) < 0.5 \end{cases}$$

1.4.3 Наївний байєсівський класифікатор (Naive Bayes)

Для аналізу тексту електронних листів часто застосовується наївний байєсівський класифікатор. Його основа — теорема Байєса:

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)}$$

де:

C — клас (фішинговий або безпечний),

X — набір ознак (слова, символи, фрази),

$P(C)$ — апіорна ймовірність класу,

$P(X/C)$ — ймовірність появи ознак у класі.

Класифікація виконується за правилом:

$$\hat{C} = \arg \max_C P(X|C) \cdot P(C)$$

1.4.4 TF-IDF для представлення тексту листів

Для перетворення тексту електронного листа у числові ознаки використовується метод TF-IDF, Term Frequency – Inverse Document Frequency. Формула TF-IDF:

$$TFIDF(t, d) = TF(t, d) \cdot IDF(t)$$

$$TF(t, d) = \frac{n_{t,d}}{\sum_k n_{k,d}}$$

$n_{t,d}$ — кількість входжень терміну t у документі d .

$$IDF(t) = \log \frac{N}{df(t)}$$

де:

N — загальна кількість документів,

$df(t)$ — кількість документів, що містять термін t .

Цей метод дозволяє підвищити вагу важливих слів, які рідко зустрічаються в загальному корпусі текстів.

1.4.5 Метрики оцінювання якості моделі

Для оцінювання ефективності класифікатора застосовуються метрики точності, повноти та F1-міри.

Точність (Accuracy)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

де:

TP — правильно визначені фішингові об'єкти,

TN — правильно визначені безпечні об'єкти,

FP — помилково визначені як фішингові,

FN — помилково визначені як безпечні.

Повнота

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

Показує, яку частку фішингових атак система змогла виявити.

Точність класифікації позитивного класу

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

Визначає, яка частина визначених фішингових об'єктів справді є небезпечною.

F1-міра

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

Ця метрика є збалансованим показником, який враховує як Precision, так і Recall.

Таким чином, математичне забезпечення розробки інформаційної системи виявлення фішингових сайтів і підозрілих листів базується на методах машинного навчання та математичних моделях класифікації. Основу становить формування вектора ознак, застосування алгоритмів логістичної регресії та наївного байєсівського класифікатора, використання TF-IDF для аналізу тексту, а також оцінювання якості моделі за метриками Accuracy, Precision, Recall та F1-score. Використання наведених математичних методів дозволяє створити ефективну систему автоматичного виявлення фішингових атак.

1.5 Постановка задачі

У сучасному інформаційному середовищі фішингові сайти та підозрілі електронні листи є однією з найбільш поширених загроз кібербезпеки. Вони використовуються для викрадення персональних даних, облікових записів, банківської інформації та доступу до корпоративних ресурсів. Зростання кількості фішингових атак, їх адаптивність і використання методів соціальної

інженерії зумовлюють необхідність створення ефективних автоматизованих засобів виявлення таких загроз.

Традиційні підходи до виявлення фішингу, зокрема використання чорних списків та фільтрації за правилами, не забезпечують достатнього рівня захисту через неможливість швидко реагувати на нові види атак. У зв'язку з цим актуальним є застосування методів машинного навчання, які дозволяють здійснювати аналіз даних, виділяти ознаки фішингових ресурсів та виконувати класифікацію об'єктів з високою точністю.

Метою бакалаврської роботи є розроблення інформаційної системи для автоматизованого виявлення фішингових сайтів та підозрілих електронних листів на основі методів машинного навчання, яка забезпечить підвищення рівня інформаційної безпеки користувачів шляхом швидкого аналізу вебресурсів і поштових повідомлень та формування результату оцінки ризику.

Об'єктом дослідження є процес виявлення фішингових атак у мережі Інтернет, зокрема аналіз вебсайтів та електронних листів на предмет наявності ознак шахрайства.

Предметом дослідження є методи, алгоритми та моделі машинного навчання, які застосовуються для класифікації вебресурсів і електронних повідомлень як фішингових або безпечних.

Для роботи інформаційної системи використовуються такі вхідні дані:

- URL-адреси вебсайтів;
- текст електронного листа тема, тіло повідомлення;
- метадані листа адреса відправника, домен, заголовки;
- наявні посилання та вкладення;
- дані про домен за можливості: вік домену, DNS-записи, SSL-сертифікат.

Результатом роботи системи є:

- класифікація об'єкта фішинговий / безпечний;
- рівень ризику низький, середній, високий;

- коротке пояснення причин підозри наявності небезпечних ознак;
- збереження результату перевірки у журналі системи.

Для досягнення поставленої мети необхідно вирішити такі задачі:

1. Провести аналіз предметної області та існуючих методів виявлення фішингових сайтів і підозрілих листів.
2. Визначити основні ознаки фішингових ресурсів та сформувати набір параметрів для аналізу URL і текстових повідомлень.
3. Обґрунтувати вибір підходів та алгоритмів машинного навчання для задачі класифікації.
4. Розробити концепцію інформаційної системи та визначити її функціональні можливості.
5. Спроекувати структуру бази даних для збереження інформації про перевірки, користувачів, підозрілі ресурси та результати класифікації.
6. Реалізувати модуль збору та попередньої обробки даних (очищення тексту, формування ознак, перетворення даних у числовий формат).
7. Побудувати та навчити модель машинного навчання для класифікації сайтів і листів.
8. Реалізувати модуль прогнозування, який забезпечує аналіз нового об'єкта та формування результату перевірки.
9. Розробити користувацький інтерфейс або API для взаємодії з системою.
10. Провести тестування системи та оцінити якість роботи моделі за метриками Accuracy, Precision, Recall та F1-score.
11. Забезпечити можливість оновлення навчальної вибірки та перенавчання моделі для підвищення актуальності системи.

Інформаційна система повинна відповідати таким вимогам:

- забезпечувати високу швидкість перевірки URL та листів;
- підтримувати масштабованість при збільшенні кількості запитів;
- зберігати результати аналізу та статистику у базі даних;

- мати зрозумілий інтерфейс або REST API;
- забезпечувати безпечну обробку даних користувачів;
- мати можливість розширення функціоналу (додавання нових моделей та ознак).

Очікуваним результатом виконання роботи є створення програмного продукту, який дозволяє автоматично виявляти фішингові сайти та підозрілі електронні листи на основі машинного навчання. Розроблена система повинна забезпечити підвищення рівня кіберзахисту користувачів, скоротити ризик витоку конфіденційної інформації та надати можливість ефективного аналізу загроз у реальному часі.

Таким чином, у підпункті сформульовано постановку задачі бакалаврської роботи, визначено мету, об'єкт і предмет дослідження, описано вхідні та вихідні дані, а також визначено перелік основних завдань, які необхідно виконати для створення інформаційної системи виявлення фішингових сайтів і підозрілих листів на основі методів машинного навчання.

Висновки до розділу 1

У першому розділі бакалаврської роботи було виконано аналіз предметної області, пов'язаної з проблемою виявлення фішингових сайтів та підозрілих електронних листів, що є однією з найбільш актуальних загроз сучасної кібербезпеки. Встановлено, що фішингові атаки активно застосовуються зловмисниками для отримання конфіденційної інформації користувачів та несанкціонованого доступу до інформаційних ресурсів, а їх кількість та складність постійно зростає.

У процесі дослідження було розглянуто основні ознаки фішингових сайтів і підозрілих листів, зокрема використання підроблених доменів, маніпулятивного тексту, шкідливих посилань, вкладень та прихованих механізмів перенаправлення. Проведений аналіз існуючих методів захисту показав, що традиційні підходи, засновані на чорних списках і евристичних правилах, мають суттєві недоліки: низьку ефективність проти нових атак,

залежність від оновлення баз даних, значну кількість помилкових спрацювань та обмежену здатність до адаптації.

Також у розділі було обґрунтовано доцільність використання методів машинного навчання як найбільш перспективного підходу для виявлення фішингу. Визначено, що ML-моделі дозволяють виконувати автоматизований аналіз великої кількості ознак, підвищувати точність класифікації та адаптуватися до нових схем атак шляхом перенавчання.

У підпункті математичного забезпечення розглянуто основні методи та формули, що застосовуються під час реалізації системи, зокрема формування вектора ознак, використання алгоритмів класифікації (логістична регресія, наївний байєсівський класифікатор), метод TF-IDF для перетворення текстової інформації у числовий формат, а також метрики оцінювання якості моделей Accuracy, Precision, Recall, F1-score. Це створює теоретичну основу для реалізації системи на практиці.

У підпункті постановки задачі визначено мету бакалаврської роботи, об'єкт та предмет дослідження, описано вхідні та вихідні дані, а також сформовано перелік основних завдань, необхідних для створення інформаційної системи виявлення фішингових ресурсів.

Таким чином, у першому розділі сформовано теоретичне підґрунтя для подальшого проєктування та розробки інформаційної системи, що дозволяє перейти до наступного етапу — проєктування архітектури, бази даних та реалізації програмного продукту із застосуванням методів машинного навчання.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ВИЯВЛЕННЯ ФІШИНГОВИХ САЙТІВ ТА ПІДОЗРІЛИХ ЛИСТІВ НА ОСНОВІ ML

2.1 Загальносистемні рішення

У процесі розроблення інформаційної системи виявлення фішингових сайтів та підозрілих електронних листів на основі методів машинного навчання необхідно визначити загальносистемні рішення, які описують принципи функціонування системи, її структуру, основні компоненти, ролі користувачів, логіку взаємодії модулів та вимоги до обробки даних. Загальносистемні рішення формують основу для подальшого проєктування архітектури, бази даних та програмної реалізації, див. рисунок 2.1.

Основна функція системи полягає у виконанні автоматичного аналізу введених даних (URL-адрес або тексту електронного листа) з метою визначення рівня небезпеки та класифікації об'єкта як фішингового або безпечного.

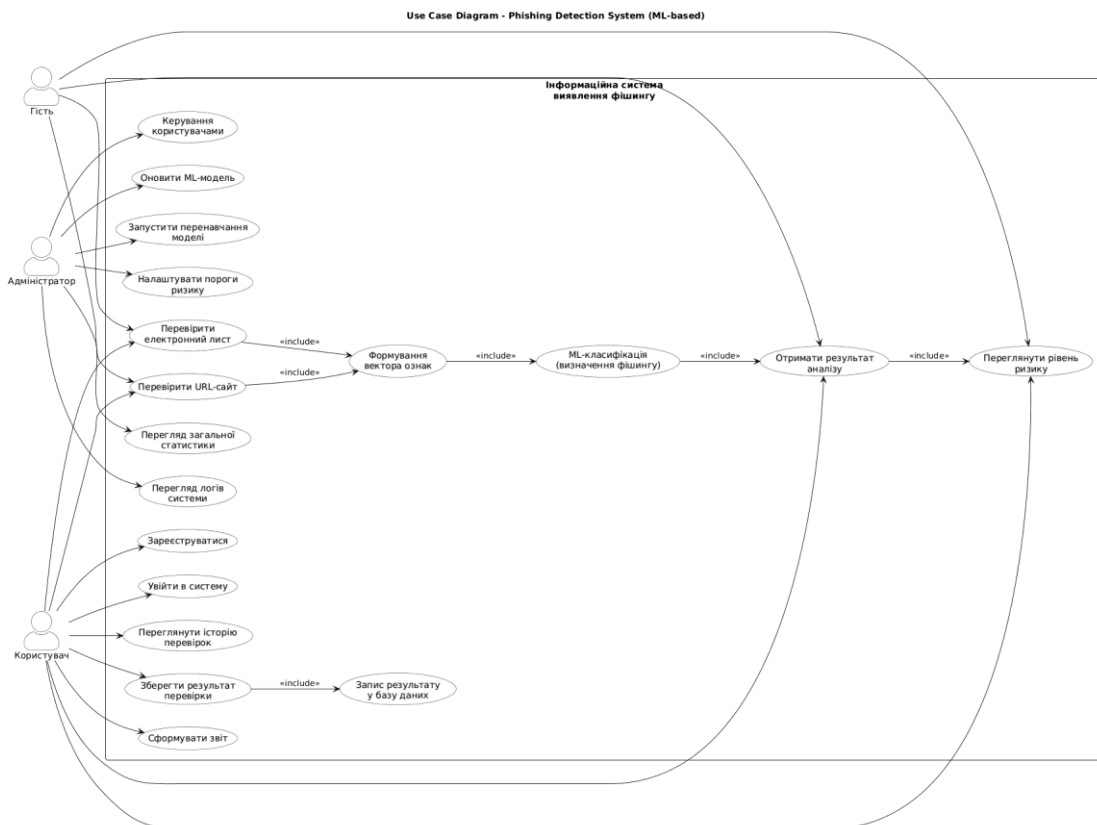


Рисунок 2.1 – Діаграма варіантів використання

2.1.1 Призначення системи

Розроблювана інформаційна система призначена для:

- автоматизованого виявлення фішингових вебсайтів;
- аналізу електронних листів на предмет наявності ознак фішингу;
- визначення рівня ризику повідомлення або ресурсу;
- надання користувачу результатів перевірки та рекомендацій щодо подальших дій;
- ведення журналу перевірок і формування статистики.

Система може бути використана як у персональному режимі для звичайних користувачів, так і у корпоративному середовищі для підвищення рівня інформаційної безпеки.

2.1.2 Основні функції системи

Інформаційна система повинна реалізовувати такі функції:

- введення URL-адреси для перевірки на фішинг;
- завантаження або введення тексту електронного листа для аналізу;
- автоматичне виділення ознак (features) з URL або тексту;
- класифікація об'єкта за допомогою ML-моделі;
- визначення ймовірності належності до класу фішингу;
- формування результату (фішинговий/безпечний);
- відображення рівня ризику (низький/середній/високий);
- збереження результатів перевірки у базі даних;
- ведення журналу подій та статистики;
- управління моделлю машинного навчання (оновлення, перенавчання).

2.1.3 Категорії користувачів та їх права доступу

Для забезпечення безпеки та зручності роботи передбачається розподіл користувачів на ролі:

- 1) Гість (неzareєстрований користувач)

Має можливість:

- виконувати перевірку URL або листа;
- отримувати результат аналізу.

Обмеження:

- немає доступу до історії перевірок;
- немає доступу до статистики.

2) Зареєстрований користувач

Має можливість:

- виконувати перевірку URL та листів;
- переглядати історію власних перевірок;
- зберігати результати аналізу;
- формувати звіти за результатами перевірок.

3) Адміністратор системи

Має можливість:

- керувати обліковими записами користувачів;
- переглядати загальну статистику перевірок;
- управляти базою підозрілих ресурсів;
- запускати оновлення або перенавчання ML-моделі;
- переглядати логи системи;
- налаштовувати порогові значення рівня ризику.

2.1.4 Структура системи та основні модулі

Для реалізації інформаційної системи пропонується модульна структура, яка включає такі основні компоненти:

1) Модуль введення та прийому даних

Забезпечує введення:

- URL-адрес вебсайтів;
- тексту листа;
- вкладень або посилань (за потреби).

2) Модуль попередньої обробки даних

Виконує очищення тексту від зайвих символів, нормалізує дані, виділяє ключові слова, аналізує структури URL. Виділяє статистичні параметри, а саме довжина URL, кількість символів, кількість посилань.

3) Модуль формування ознак

Формує вектор ознак для подальшої класифікації. Для листів може застосовуватися TF-IDF або Bag-of-Words, а для URL — набір структурних параметрів.

4) Модуль машинного навчання ML-модуль

Основний модуль системи, який виконує:

- завантаження навченої моделі;
- прогнозування результату;
- визначення ймовірності фішингу.

Може використовувати такі алгоритми:

- Logistic Regression;
- Random Forest;
- Gradient Boosting;
- Support Vector Machine.

5) Модуль оцінювання ризику

На основі ймовірності фішингу формує рівень ризику:

- низький 0–40%;
- середній 40–70%;
- високий 70–100%.

6) Модуль збереження та журналювання

Забезпечує:

- запис результатів перевірки у базу даних;
- ведення історії перевірок;
- збереження статистики та логів подій.

7) Модуль адміністративного керування

Дозволяє адміністратору:

- керувати користувачами;

- переглядати загальні результати роботи;
- запускати перенавчання моделей;
- оновлювати навчальні набори даних.

2.1.5 Загальна логіка роботи системи

Процес роботи системи можна описати наступними етапами:

1. Користувач вводить URL або текст електронного листа.
2. Система виконує попередню обробку інформації.
3. Формується вектор ознак для аналізу.
4. Дані передаються до ML-модуля.
5. ML-модель виконує класифікацію та обчислює ймовірність фішингу.
6. Модуль оцінювання ризику визначає рівень небезпеки.
7. Користувач отримує результат перевірки.
8. Результат зберігається у базі даних та додається до журналу.

2.1.6 Вимоги до інформаційних потоків

Система повинна забезпечувати обмін інформацією між модулями за такими принципами:

- використання API-запитів (REST);
- передача даних у форматі JSON;
- можливість інтеграції з поштовими сервісами або браузерами;
- захищене з'єднання HTTPS.

2.1.7 Вимоги до масштабованості та розширюваності

Оскільки фішингові атаки постійно змінюються, система повинна підтримувати:

- можливість оновлення набору ознак;
- додавання нових алгоритмів ML;
- перенавчання моделей на нових даних;
- зберігання історії перевірок для накопичення статистики.

Таким чином, у підпункті визначено загальносистемні рішення для розроблення інформаційної системи виявлення фішингових сайтів і підозрілих листів. Запропоновано функціональну структуру системи, визначено ролі користувачів, основні модулі, принципи обробки даних та логіку взаємодії компонентів. Отримані результати створюють основу для подальшого проєктування архітектури системи, моделі бази даних та реалізації програмного продукту із застосуванням методів машинного навчання.

2.2 Архітектура системи

Архітектура інформаційної системи є одним із ключових етапів проєктування програмного забезпечення, оскільки визначає структуру системи, взаємодію її компонентів, принципи обробки даних та механізми забезпечення масштабованості й надійності. Для системи виявлення фішингових сайтів та підозрілих електронних листів на основі методів машинного навчання архітектура повинна забезпечувати можливість швидкого аналізу даних, обробки великої кількості запитів, інтеграції ML-моделі та збереження результатів перевірки.

У процесі розроблення системи було обрано багаторівневу модульну архітектуру, рисунок 2.2, яка дозволяє розділити функціональні компоненти за їхнім призначенням та забезпечити гнучкість подальшого розвитку системи.

2.2.1 Загальна структура архітектури

Інформаційна система складається з таких основних рівнів:

1. Рівень представлення Presentation Layer
2. Серверний рівень Application Layer
3. Рівень машинного навчання ML Layer
4. Рівень зберігання даних Data Layer

Кожен із рівнів виконує окремі функції та взаємодіє з іншими компонентами через API-запити та внутрішні сервіси.

2.2.2 Рівень представлення

Рівень представлення забезпечує взаємодію користувача із системою. Він реалізується у вигляді вебінтерфейсу або клієнтського застосунку.

Основні функції:

- введення URL-адреси або тексту електронного листа;
- завантаження вкладень;
- відображення результатів перевірки;
- перегляд історії перевірок;
- відображення статистики та рівня ризику.

Для реалізації інтерфейсу можуть використовуватися:

- HTML5;
- CSS3;
- JavaScript;
- React або Vue.js.

Взаємодія з серверною частиною здійснюється через REST API із використанням формату JSON.

2.2.3 Серверний рівень (Application Layer)

Серверний рівень є центральною частиною системи та забезпечує обробку запитів користувачів, взаємодію з ML-модулем та базою даних.

Основні функції серверного рівня:

- прийом HTTP-запитів;
- валідація введених даних;
- керування логікою системи;
- виклик модулів попередньої обробки;
- взаємодія з ML-моделлю;
- збереження результатів перевірки;
- формування відповідей для користувача.

Для реалізації серверної частини можуть використовуватися:

- Python;

- Flask або FastAPI;
- Node.js альтернативно.

Основною перевагою серверної архітектури є можливість централізованого управління логікою системи та масштабування окремих компонентів.

2.2.4 Модуль попередньої обробки даних

Перед передачею інформації до ML-моделі необхідно виконати попередню обробку даних.

Для URL-адрес:

- аналіз структури URL;
- визначення довжини URL;
- перевірка HTTPS;
- підрахунок кількості піддоменів;
- виявлення спеціальних символів;
- аналіз домену та SSL-сертифіката.

Для електронних листів:

- очищення тексту;
- токенизація;
- видалення stop-слів;
- нормалізація тексту;
- аналіз HTML-коду;
- перевірка вкладень та посилань.

Результатом роботи модуля є формування набору ознак feature vector, який передається до ML-класифікатора.

2.2.5 Рівень машинного навчання ML Layer

ML-рівень є основним компонентом системи, який відповідає за класифікацію об'єктів.

Основні функції:

- завантаження навченої моделі;

- аналіз вектора ознак;
- прогнозування результату;
- визначення ймовірності фішингу;
- повернення результату серверному модулю.

Для реалізації ML-модуля можуть використовуватися:

- Scikit-learn;
- TensorFlow;
- Keras;
- PyTorch.

У системі можуть бути використані такі алгоритми:

- Logistic Regression;
- Random Forest;
- Support Vector Machine;
- Gradient Boosting;
- Neural Networks.

Після виконання класифікації система визначає:

- тип об'єкта (безпечний / фішинговий);
- рівень ризику;
- відсоток ймовірності фішингу.

2.2.6 Рівень зберігання даних

Рівень зберігання даних забезпечує роботу з інформацією системи та підтримує довготривале збереження результатів аналізу.

У базі даних зберігаються:

- облікові записи користувачів;
- результати перевірок;
- журнали подій;
- статистика роботи системи;
- історія класифікацій;
- набори навчальних даних.

Для реалізації можуть використовуватися:

- PostgreSQL;
- MySQL;
- MongoDB (для неструктурованих даних).

Основні вимоги до бази даних:

- підтримка швидкого пошуку;
- забезпечення цілісності даних;
- підтримка резервного копіювання;
- масштабованість.

2.2.7 Взаємодія компонентів системи

Взаємодія між компонентами відбувається за таким алгоритмом:

1. Користувач вводить URL або текст листа.
2. Інтерфейс надсилає запит до серверного API.
3. Сервер виконує перевірку коректності даних.
4. Дані передаються до модуля попередньої обробки.
5. Формується вектор ознак.
6. ML-модель виконує класифікацію.
7. Результат повертається серверу.
8. Сервер зберігає інформацію у базі даних.
9. Користувач отримує результат перевірки.

2.2.8 Переваги обраної архітектури

Запропонована архітектура має такі переваги:

- модульність і простота масштабування;
- можливість незалежного оновлення компонентів;
- підтримка інтеграції нових ML-моделей;
- централізоване управління логікою системи;
- підтримка хмарного розгортання;
- можливість роботи через REST API;
- зручність підтримки та тестування.

2.2.9 Безпека системи

Оскільки система працює з потенційно небезпечними даними, необхідно забезпечити:

- використання HTTPS;
- захист API-токенами;
- перевірку введених даних;
- обмеження доступу за ролями;
- журналювання подій;
- резервне копіювання бази даних.

Таким чином, у підпункті було розроблено архітектуру інформаційної системи виявлення фішингових сайтів та підозрілих електронних листів на основі ML. Запропонована багаторівнева модульна архітектура забезпечує ефективну взаємодію між компонентами системи, підтримує масштабованість, інтеграцію алгоритмів машинного навчання та безпечне зберігання даних. Отримані результати створюють основу для подальшого проєктування бази даних, UML-діаграм та реалізації програмної частини системи.



Рисунок 2.2 – Архітектура системи

2.3 Взаємодії користувача із системою

Діаграма послідовності відображає процес взаємодії користувача із системою під час перевірки URL-адреси або електронного листа. Основними учасниками процесу є користувач, вебінтерфейс системи, серверна частина, модуль обробки даних, ML-модель та база даних. У результаті система виконує класифікацію об'єкта, визначає рівень ризику та повертає результат користувачу.

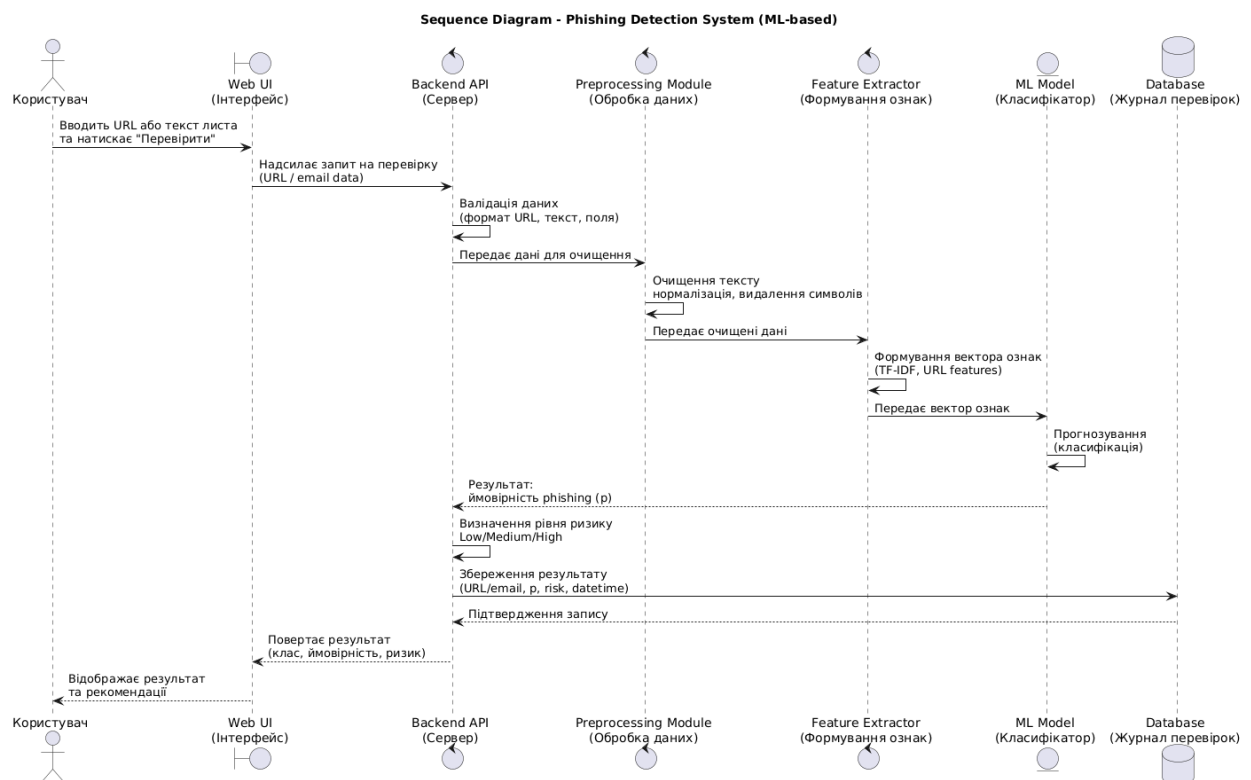


Рисунок 2.3 – Діаграма послідовності

Основні учасники процесу:

- Користувач – ініціює перевірку URL або листа;
- Web UI – забезпечує введення даних і відображення результатів;
- Backend API – приймає запит та керує виконанням операцій;
- Preprocessing Module – очищує дані та формує ознаки;
- ML Model – виконує класифікацію фішинг/безпечний;
- Database – зберігає історію перевірок і результати.

Основний сценарій роботи системи

1. Користувач вводить URL або текст листа у вебінтерфейсі.

2. Web UI передає дані на Backend API.
3. Backend API виконує первинну валідацію даних.
4. Дані передаються до модуля попередньої обробки.
5. Preprocessing Module формує вектор ознак.
6. ML Model отримує ознаки та виконує класифікацію.
7. Система визначає ймовірність фішингу та рівень ризику.
8. Результат зберігається у базі даних.
9. Backend API повертає результат на Web UI.
10. Web UI відображає висновок користувачу.

Таким чином, діаграма послідовності демонструє логіку функціонування системи та порядок взаємодії між її компонентами під час аналізу підозрілого ресурсу або листа. Це дозволяє чітко формалізувати процес перевірки та підтверджує модульну архітектуру системи із застосуванням ML-класифікації.

2.4 Розробка бази даних системи

База даних є одним із ключових компонентів інформаційної системи виявлення фішингових сайтів та підозрілих електронних листів, оскільки забезпечує централізоване зберігання, обробку та доступ до інформації, необхідної для функціонування системи. Від правильності проєктування бази даних залежить швидкість роботи системи, цілісність інформації, можливість масштабування та ефективність аналізу накопичених даних.

Основним призначенням бази даних є:

- збереження інформації про користувачів системи;
- ведення історії перевірок URL та електронних листів;
- зберігання результатів ML-класифікації;
- накопичення статистики та журналів подій;
- зберігання навчальних наборів даних;
- забезпечення можливості формування аналітичних звітів.

Для реалізації бази даних доцільно використовувати реляційну СУБД PostgreSQL або MySQL, оскільки вони забезпечують високу продуктивність, підтримку складних запитів, механізми індексації та надійне збереження інформації.

2.4.1 Основні вимоги до бази даних

База даних системи повинна відповідати таким вимогам:

- забезпечувати швидке збереження та пошук інформації;
- підтримувати цілісність і узгодженість даних;
- підтримувати багатокористувацький режим роботи;
- забезпечувати захист інформації;
- підтримувати резервне копіювання;
- забезпечувати можливість масштабування при збільшенні кількості перевірок;
- підтримувати журналювання подій системи.

2.4.2 Аналіз інформаційних потоків

У процесі роботи системи відбувається обробка таких інформаційних потоків:

- введення URL-адрес або тексту електронного листа;
- передача даних до ML-модуля;
- отримання результату класифікації;
- збереження результатів перевірки;
- формування історії перевірок;
- управління користувачами та ролями;
- ведення журналів системних подій.

На основі цих потоків було визначено основні сутності бази даних.

2.4.3 Основні сутності бази даних

У результаті аналізу предметної області було визначено такі основні сутності, рисунок 2.4.

1) Users

Таблиця містить інформацію про користувачів системи.

Основні атрибути:

user_id;

username;

email;

password_hash;

role;

created_at.

2) URL_Checks

Таблиця містить результати перевірки вебсайтів.

Основні атрибути:

check_id;

user_id;

url;

phishing_probability;

risk_level;

result;

checked_at.

3) Email_Checks

Таблиця призначена для збереження результатів перевірки електронних листів.

Основні атрибути:

email_check_id;

user_id;

sender_email;

subject;

email_text;

phishing_probability;

risk_level;

result;

checked_at.

4) ML_Models

Таблиця містить інформацію про моделі машинного навчання.

Основні атрибути:

model_id;

model_name;

algorithm;

version;

accuracy;

created_at.

5) Logs

Таблиця використовується для журналювання подій системи.

Основні атрибути:

log_id;

user_id;

action;

ip_address;

created_at.

6) Training_Datasets

Таблиця містить інформацію про навчальні набори даних.

Основні атрибути:

dataset_id;

dataset_name;

source;

records_count;

uploaded_at.

Conceptual ER Diagram - Phishing Detection System

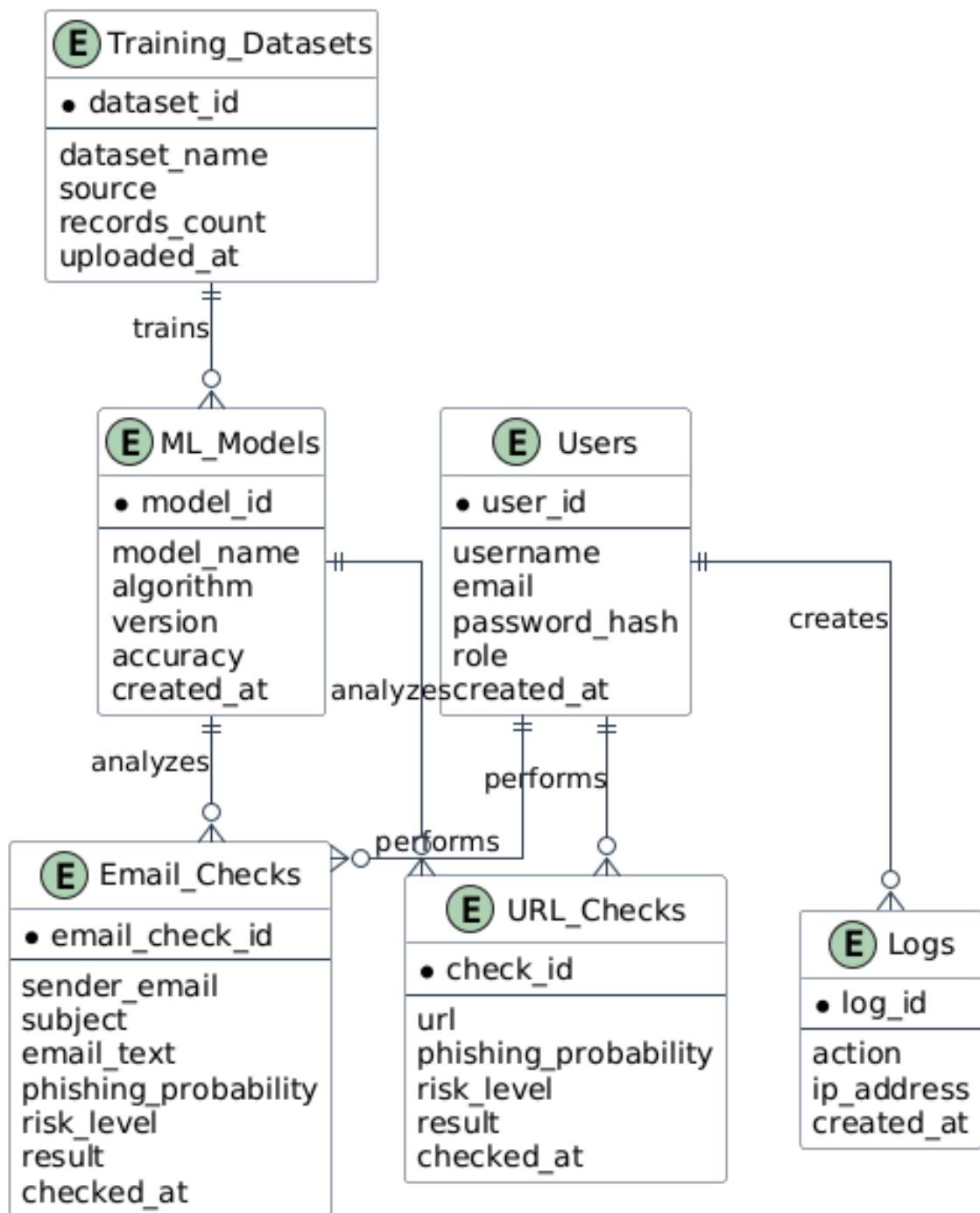


Рисунок 2.4 – Концептуальна модель даних

2.4.4 Логічна структура бази даних

Основою структури бази даних є зв'язки між таблицями, рисунок 2.5.

Logical ER Diagram - Phishing Detection System

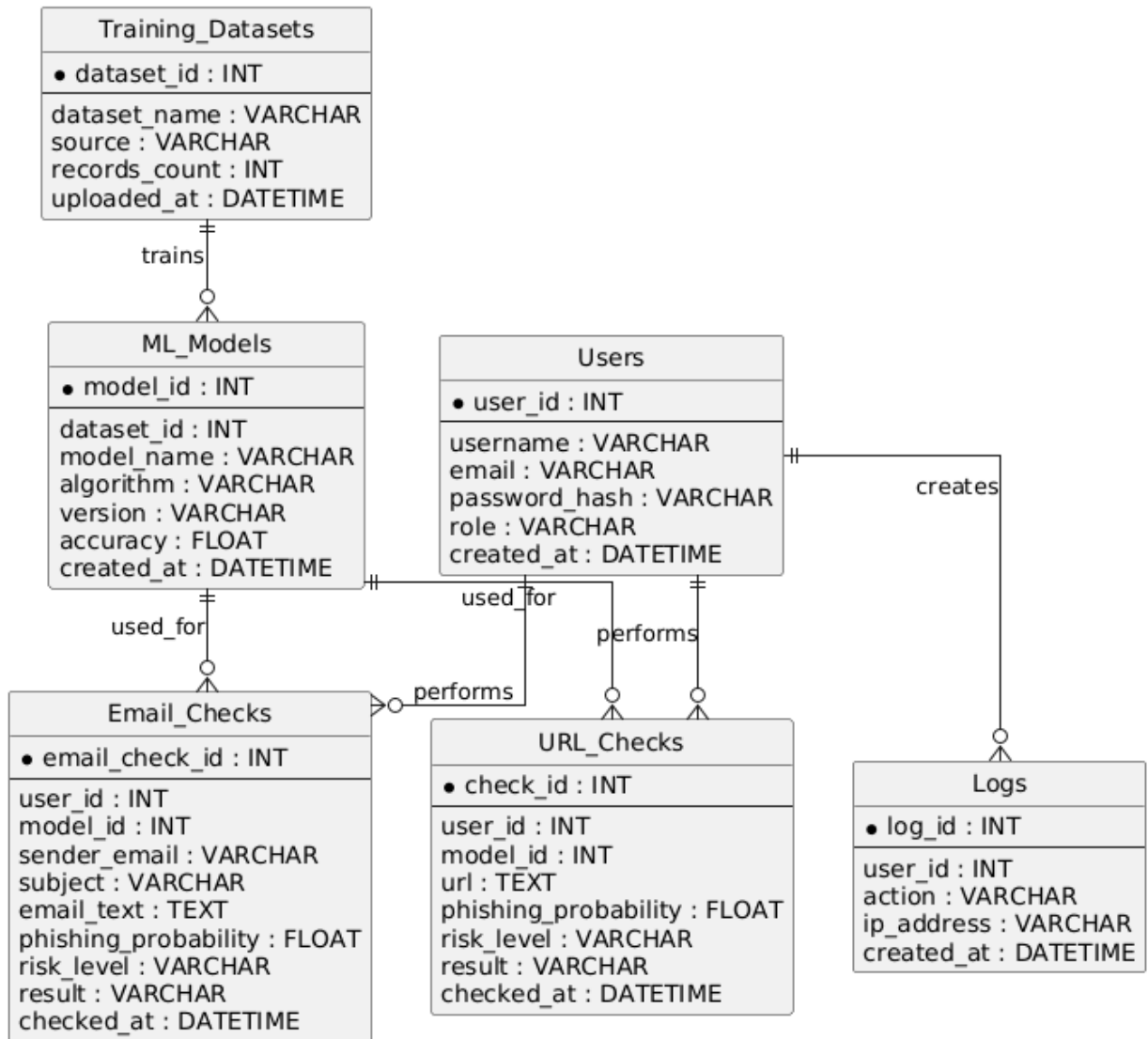


Рисунок 2.5 – Логічна модель даних

Основні зв'язки:

- один користувач може виконати багато перевірок URL;
- один користувач може перевірити багато електронних листів;
- одна ML-модель може використовуватися для багатьох перевірок;
- один користувач може створити багато записів журналу.

Типи зв'язків:

Users → URL_Checks : один-до-багатьох;

Users → Email_Checks : один-до-багатьох;

Users → Logs : один-до-багатьох;

ML_Models → URL_Checks : один-до-багатьох;

ML_Models → Email_Checks : один-до-багатьох.

2.4.5 Нормалізація бази даних

Для забезпечення цілісності даних база даних приводиться до третьої нормальної форми (3NF).

Це дозволяє:

- уникнути дублювання інформації;
- спростити оновлення даних;
- забезпечити узгодженість інформації;
- зменшити ризик виникнення аномалій.

Кожна таблиця містить первинний ключ (Primary Key), а зв'язки реалізуються через зовнішні ключі Foreign Key.

2.4.6 Забезпечення безпеки даних

Оскільки система працює з потенційно небезпечними ресурсами та персональними даними користувачів, необхідно реалізувати:

- хешування паролів;
- контроль доступу за ролями;
- резервне копіювання бази даних;
- аудит дій користувачів;
- захищене підключення до СУБД;
- обмеження SQL-доступу.

2.4.7 Індексція та оптимізація

Для підвищення продуктивності системи передбачається використання індексів для таких полів:

email;

username;

checked_at;

risk_level;

result.

Індексація дозволяє прискорити:

- пошук історії перевірок;
- формування статистики;
- фільтрацію результатів;
- генерацію звітів.

2.4.8 Масштабованість бази даних

У процесі експлуатації система може накопичувати великі обсяги даних, тому архітектура бази даних повинна підтримувати:

- горизонтальне масштабування;
- архівування старих записів;
- реплікацію даних;
- резервне копіювання;
- можливість інтеграції з Big Data-сховищами.

Таким чином, у підпункті було розроблено структуру бази даних інформаційної системи виявлення фішингових сайтів та підозрілих електронних листів. Визначено основні сутності, їх атрибути та взаємозв'язки, обґрунтовано вибір реляційної моделі та принципи нормалізації. Запропонована структура бази даних забезпечує ефективне зберігання результатів перевірок, підтримує роботу ML-модулів, дозволяє вести статистику та забезпечує масштабованість і безпеку системи.

Висновки до розділу 2

У другому розділі бакалаврської роботи було виконано проектування інформаційної системи виявлення фішингових сайтів та підозрілих електронних листів на основі методів машинного навчання. У процесі розробки визначено основні структурні та функціональні рішення, необхідні для реалізації ефективної системи аналізу та класифікації потенційно небезпечних ресурсів.

У підпункті загальносистемних рішень було визначено призначення системи, її функціональні можливості, ролі користувачів та основні модулі. Встановлено, що система повинна забезпечувати автоматизований аналіз URL-адрес і електронних листів, визначення рівня ризику, збереження результатів перевірки та підтримку роботи ML-моделей. Було сформовано модульну структуру системи, що забезпечує масштабованість, гнучкість та можливість подальшого розвитку програмного продукту.

У підпункті архітектури системи розроблено багаторівневу архітектуру, яка включає рівень представлення, серверний рівень, модуль машинного навчання та рівень зберігання даних. Запропонована архітектура забезпечує ефективну взаємодію між компонентами системи, підтримує використання REST API та дозволяє інтегрувати сучасні ML-бібліотеки й алгоритми класифікації. Також було визначено принципи взаємодії модулів, механізми обробки інформації та вимоги до безпеки системи.

Для формалізації логіки роботи системи було побудовано UML-діаграми, зокрема діаграму варіантів використання та діаграму послідовності. Діаграма варіантів використання дозволила визначити взаємодію користувачів із системою та основні функціональні сценарії роботи. Діаграма послідовності продемонструвала процес проходження даних через модулі системи — від введення URL або листа до отримання результату ML-класифікації та збереження інформації у базі даних.

У підпункті розробки бази даних було визначено основні сутності системи, їх атрибути та взаємозв'язки. Було спроектовано концептуальну та логічну моделі бази даних, що включають таблиці користувачів, результатів перевірок, ML-моделей, журналів подій та навчальних наборів даних. Обґрунтовано використання реляційної СУБД PostgreSQL або MySQL, а також принципи нормалізації та індексації даних для підвищення продуктивності системи.

У результаті виконання другого розділу сформовано повну проєктну основу для реалізації інформаційної системи виявлення фішингових сайтів та

підозрілих листів. Розроблені архітектурні рішення, UML-діаграми та модель бази даних створюють підґрунтя для подальшої програмної реалізації системи, інтеграції алгоритмів машинного навчання та проведення тестування програмного продукту.

РОЗДІЛ 3. РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ВИЯВЛЕННЯ ФІШИНГОВИХ САЙТІВ ТА ПІДОЗРІЛИХ ЛИСТІВ НА ОСНОВІ ML

3.1 Вимоги до програмного забезпечення

Для забезпечення ефективної роботи інформаційної системи виявлення фішингових сайтів та підозрілих електронних листів на основі методів машинного навчання необхідно визначити вимоги до програмного забезпечення, яке використовується під час розробки, розгортання та експлуатації системи. Правильно підібране програмне забезпечення забезпечує стабільність роботи системи, високу швидкодію, підтримку алгоритмів машинного навчання, масштабованість та безпеку обробки даних.

Розроблювана система є багатокomпонентним програмним продуктом, який включає вебінтерфейс, серверну частину, ML-модулі, базу даних та засоби адміністрування. У зв'язку з цим до програмного забезпечення висуваються вимоги щодо сумісності, продуктивності, підтримки сучасних технологій та можливості інтеграції між компонентами.

3.1.1 Вимоги до операційної системи

Для функціонування серверної частини системи доцільно використовувати сучасні операційні системи, які підтримують мережеві сервіси, ML-бібліотеки та засоби роботи з базами даних.

Рекомендовані операційні системи:

- Ubuntu Server 22.04 LTS;
- Debian 12;
- Windows Server 2019/2022;
- Linux-based системи.

Найбільш доцільним є використання Linux-систем, оскільки вони:

- мають високу стабільність;
- забезпечують кращу продуктивність серверних застосунків;
- підтримують Python та ML-фреймворки;

- мають вбудовані засоби безпеки;
- дозволяють ефективно працювати з Docker та хмарними сервісами.

3.1.2 Вимоги до серверного програмного забезпечення

Серверне програмне забезпечення повинно забезпечувати:

- обробку HTTP-запитів;
- взаємодію з ML-модулем;
- роботу REST API;
- обробку результатів класифікації;
- роботу з базою даних;
- автентифікацію користувачів.

Для реалізації серверної частини рекомендується використовувати:

Мови програмування:

- Python 3.11+;
- JavaScript (Node.js — альтернативно).
- Backend Framework:
- Flask;
- FastAPI;
- Django (за необхідності розширеної функціональності).

Для даної системи найбільш доцільним є використання FastAPI, оскільки він:

- має високу продуктивність;
- підтримує асинхронну обробку;
- зручний для реалізації REST API;
- добре інтегрується з ML-бібліотеками.

3.1.3 Вимоги до програмного забезпечення машинного навчання

Оскільки система базується на ML-алгоритмах, необхідно використовувати бібліотеки для:

- обробки даних;

- навчання моделей;
- тестування;
- прогнозування;
- оцінювання точності моделей.

Рекомендовані бібліотеки:

- Scikit-learn;
- TensorFlow;
- Keras;
- PyTorch;
- NumPy;
- Pandas;
- NLTK;
- SpaCy.

Для задач класифікації URL та текстових повідомлень найбільш доцільним є використання Scikit-learn, оскільки бібліотека:

- містить готові алгоритми класифікації;
- має простий API;
- підтримує TF-IDF;
- забезпечує швидке навчання моделей.

3.1.4 Вимоги до клієнтського програмного забезпечення

Клієнтська частина системи повинна забезпечувати:

- введення URL або тексту листа;
- відображення результатів аналізу;
- перегляд історії перевірок;
- роботу через веббраузер.

Для реалізації інтерфейсу можуть використовуватися:

- HTML5;
- CSS3;
- JavaScript;
- React;

- Vue.js.

Користувач повинен мати можливість працювати із системою через сучасні браузерери:

- Google Chrome;
- Mozilla Firefox;
- Microsoft Edge;
- Opera.

3.1.5 Вимоги до системи управління базами даних

Для збереження результатів перевірок та службової інформації необхідно використовувати СУБД, яка підтримує:

- багатокористувацький режим;
- індексацію;
- транзакції;
- резервне копіювання;
- механізми безпеки.

Рекомендовані СУБД:

- PostgreSQL;
- MySQL;
- MariaDB.

Найбільш доцільним є використання PostgreSQL через:

- високу стабільність;
- підтримку складних SQL-запитів;
- високу продуктивність;
- підтримку JSON-полів;
- механізми масштабування.

3.1.6 Вимоги до мережевого програмного забезпечення

Для забезпечення доступу користувачів до системи необхідно використовувати:

- вебсервер;

- API-сервіси;
- засоби захищеного мережевого з'єднання.

Рекомендоване ПЗ:

- Nginx;
- Apache HTTP Server;
- Gunicorn/Uvicorn.

Для захисту даних необхідно:

- використовувати HTTPS;
- застосовувати SSL/TLS-сертифікати;
- обмежувати доступ до API;
- використовувати JWT-автентифікацію.

3.1.7 Вимоги до засобів розгортання та контейнеризації

Для спрощення розгортання системи рекомендується використання контейнеризації.

Рекомендовані засоби:

- Docker;
- Docker Compose;
- Kubernetes (для масштабованих рішень).

Контейнеризація дозволяє:

- ізолювати компоненти системи;
- спростити оновлення;
- забезпечити переносимість;
- прискорити розгортання.

3.1.8 Вимоги до безпеки програмного забезпечення

Програмне забезпечення повинно забезпечувати:

- автентифікацію користувачів;
- авторизацію за ролями;
- захист паролів шляхом хешування;
- захист від SQL-ін'єкцій;

- журналювання дій користувачів;
- перевірку вхідних даних;
- резервне копіювання.

Також необхідно забезпечити:

- регулярне оновлення бібліотек;
- моніторинг роботи сервера;
- обмеження доступу до адміністративних функцій.

3.1.9 Вимоги до продуктивності системи

Інформаційна система повинна:

- обробляти запити у реальному часі;
- підтримувати одночасну роботу кількох користувачів;
- забезпечувати швидку класифікацію URL та листів;
- підтримувати масштабування при збільшенні навантаження.

Рекомендований час відповіді системи:

- до 2–3 секунд для перевірки URL;
- до 5 секунд для аналізу електронного листа.

Таким чином, у підпункті було визначено вимоги до програмного забезпечення інформаційної системи виявлення фішингових сайтів та підозрілих електронних листів на основі ML. Було обґрунтовано вибір операційних систем, серверних технологій, ML-бібліотек, СУБД та засобів мережевої взаємодії. Визначені вимоги забезпечують стабільність, масштабованість, безпечність та ефективність роботи системи, а також створюють основу для подальшого розгортання та тестування програмного продукту.

3.2 Вимоги до апаратного забезпечення

Для забезпечення стабільної та ефективної роботи інформаційної системи виявлення фішингових сайтів та підозрілих електронних листів на основі методів машинного навчання необхідно визначити вимоги до апаратного забезпечення. Враховуючи використання ML-моделей, обробку

текстових даних, мережеву взаємодію та роботу з базами даних, апаратна платформа повинна забезпечувати достатню продуктивність, надійність та можливість масштабування.

Апаратне забезпечення системи включає серверну частину, клієнтські робочі станції, мережеве обладнання, засоби резервного зберігання даних.

Правильно підібрані апаратні ресурси забезпечують швидке виконання алгоритмів класифікації, стабільну роботу вебсервісів та підтримку багатокористувацького режиму роботи.

3.2.1 Вимоги до серверного обладнання

Серверне обладнання повинно забезпечувати обробку HTTP-запитів, виконання ML-алгоритмів, взаємодію з базою даних, одночасну роботу декількох користувачів, зберігання результатів перевірок.

Мінімальні характеристики сервера. Таблиця 3.1

Компонент	Мінімальні вимоги
Процесор	Intel Core i5 / AMD Ryzen 5
Кількість ядер	4 ядра
Оперативна пам'ять	8 ГБ RAM
Накопичувач	SSD 256 ГБ
Мережевий адаптер	1 Gbit/s
Операційна система	Linux Ubuntu Server

Рекомендовані характеристики сервера. Таблиця 3.2

Компонент	Рекомендовані вимоги
Процесор	Intel Xeon / AMD EPYC
Кількість ядер	8–16 ядер
Оперативна пам'ять	16–32 ГБ RAM
Накопичувач	NVMe SSD 512 ГБ – 1 ТБ
GPU (для ML)	NVIDIA RTX / Tesla
Мережа	1–10 Gbit/s

Використання SSD-накопичувачів дозволяє суттєво прискорити доступ до бази даних, запуск ML-моделей, обробку великих наборів даних, формування статистики.

3.2.2 Вимоги до апаратного забезпечення ML-модуля

Оскільки система використовує алгоритми машинного навчання, важливе значення має продуктивність обчислювального обладнання.

Для навчання та тестування ML-моделей рекомендується багатоядерний процесор, великий обсяг оперативної пам'яті, використання графічного процесора (GPU).

Рекомендовані характеристики ML-сервера. Таблиця 3.3

Компонент	Характеристика
CPU	Intel Xeon / Ryzen 7+
RAM	32 ГБ і більше
GPU	NVIDIA RTX 3060/4060 або вище
Відеопам'ять	від 8 ГБ VRAM
SSD	NVMe SSD 1 ТБ

GPU-прискорення дозволяє скоротити час навчання моделей, швидше виконувати прогнозування, ефективніше працювати з нейромережами.

3.2.3 Вимоги до клієнтських робочих станцій

Клієнтські пристрої використовуються для роботи з вебінтерфейсом, введення URL або тексту листа, перегляду результатів перевірки, формування звітів.

Мінімальні характеристики клієнтського ПК. Таблиця 3.4

Компонент	Мінімальні вимоги
Процесор	Intel Core i3
Оперативна пам'ять	4 ГБ
Накопичувач	SSD 128 ГБ
Браузер	Chrome / Firefox
Мережа	Доступ до Інтернету

Рекомендовані характеристики. Таблиця 3.5

Компонент	Рекомендовані вимоги
Процесор	Intel Core i5 / Ryzen 5
Оперативна пам'ять	8 ГБ

Компонент	Рекомендовані вимоги
Накопичувач	SSD 256 ГБ
Монітор	Full HD
Мережа	Стабільне широкосмугове підключення

Оскільки система працює через браузер, високих вимог до клієнтського обладнання не висувається.

3.2.4 Вимоги до мережевого обладнання

Для забезпечення стабільної роботи системи необхідне мережеве обладнання, яке підтримує швидку передачу даних, захищене мережеве з'єднання, багатокористувацький режим роботи.

Основними вимогами є швидкість мережі не менше 100 Мбіт/с підтримка TCP/IP, підтримка HTTPS, використання маршрутизаторів із функціями захисту, підтримка VPN (за необхідності).

Для серверного обладнання рекомендується використання Gigabit Ethernet, резервні мережеві канали, мережевий firewall.

3.2.5 Вимоги до систем резервного копіювання

Оскільки система зберігає результати перевірок та журнали подій, необхідно забезпечити резервне копіювання даних.

Рекомендується використання зовнішніх NAS-сховищ, RAID-масивів, хмарних сервісів резервування, автоматичних backup-систем.

Резервне копіювання повинно виконуватися щоденно для бази даних, щотижнево для системних файлів, перед оновленням ML-моделей.

3.2.6 Вимоги до масштабованості апаратної платформи

У процесі експлуатації кількість користувачів та обсяг даних можуть збільшуватися, тому система повинна підтримувати горизонтальне масштабування, підключення додаткових серверів, балансування навантаження, використання хмарної інфраструктури.

Для масштабованих рішень було використано: AWS, Microsoft Azure, Google Cloud Platform.

3.2.7 Вимоги до енергоефективності та надійності

Серверне обладнання повинно підтримувати безперервну роботу, мати систему охолодження, забезпечувати захист від перепадів напруги, підтримувати джерела безперебійного живлення (UPS).

Для серверних рішень рекомендується використання серверних блоків живлення, моніторинг температури, автоматичне відновлення після збоїв.

Таким чином, у підпункті було визначено вимоги до апаратного забезпечення інформаційної системи виявлення фішингових сайтів та підозрілих електронних листів на основі ML. Було встановлено мінімальні та рекомендовані характеристики серверного обладнання, клієнтських пристроїв, ML-сервера та мережевої інфраструктури. Запропоновані вимоги забезпечують стабільну роботу системи, швидке виконання ML-алгоритмів, підтримку багатокористувацького режиму та можливість масштабування програмного продукту в майбутньому.

3.3 Розгортання системи

Розгортання інформаційної системи є важливим етапом реалізації програмного продукту, який визначає принципи фізичного розміщення компонентів системи, їх взаємодію у мережевому середовищі та способи забезпечення стабільної роботи сервісів. Для інформаційної системи виявлення фішингових сайтів та підозрілих електронних листів на основі ML процес розгортання повинен забезпечувати високу доступність, масштабованість, безпечність та ефективну взаємодію між усіма програмними компонентами.

Розроблювана система є багаторівневою web-oriented системою, рисунок 3.1 , яка включає:

- клієнтський інтерфейс;
- сервер прикладної логіки;
- ML-модуль;
- сервер бази даних;

- мережеві сервіси та засоби безпеки.

Для забезпечення стабільної роботи системи використовується клієнт-серверна архітектура із можливістю хмарного або локального розгортання.

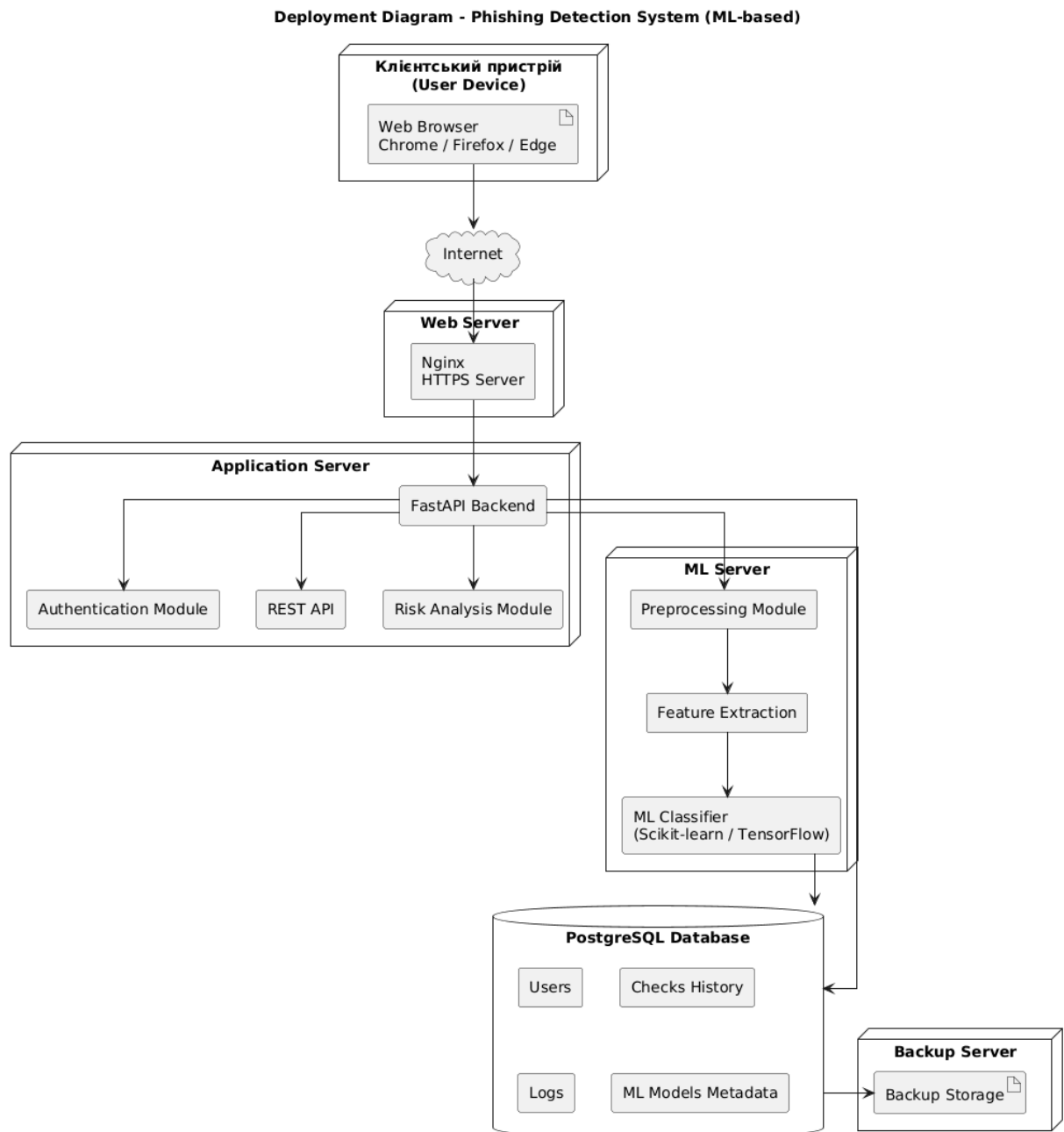


Рисунок 3.1 – Діаграма розгортання системи

3.3.1 Загальна схема розгортання системи

Система складається з таких основних вузлів:

1. Клієнтський пристрій користувача;
2. Web-сервер;
3. Сервер прикладної логіки;

4. ML-сервер;
5. Сервер бази даних;
6. Система резервного копіювання.

Усі компоненти взаємодіють між собою через захищені мережеві протоколи.

3.3.2 Клієнтський рівень

Клієнтський рівень представлений персональними комп'ютерами або мобільними пристроями користувачів.

Основні функції клієнтського рівня це введення URL або тексту електронного листа, відображення результатів аналізу, перегляд історії перевірок, взаємодія із сервером через браузер.

Для роботи користувачу необхідноступ до мережі Інтернет, сучасний веббраузер, підтримка JavaScript та HTTPS.

Взаємодія із сервером здійснюється через HTTP/HTTPS-запити, REST API, JSON-формат обміну даними.

3.3.3 Web-сервер

Web-сервер забезпечує прийом HTTP-запитів, маршрутизацію запитів, передачу даних між клієнтом і сервером, захист мережевої взаємодії.

Для реалізації web-сервера було використано Nginx через високу продуктивність, підтримку великої кількості одночасних з'єднань, інтеграцію з FastAPI та Docker, підтримку SSL/TLS.

Web-сервер також виконує балансування навантаження, кешування запитів, захист від DDoS-атак.

3.3.4 Сервер прикладної логіки

Сервер прикладної логіки є центральним компонентом системи та забезпечує обробку бізнес-логіки, взаємодію з ML-модулем, керування перевітками URL та листів, формування відповідей користувачам, взаємодію з базою даних.

Для реалізації серверної частини використовуються Python, FastAPI, Uvicorn/Gunicorn.

Основні функції це автентифікація користувачів, обробка REST API, валідація даних, передача даних до ML-модуля, формування рівня ризику.

3.3.5 ML-сервер

ML-сервер виконує завантаження навченої моделі, обробку ознак, виконання класифікації, прогнозування рівня ризику.

На ML-сервері розміщуються моделі машинного навчання, бібліотеки TensorFlow, Scikit-learn або PyTorch, модулі попередньої обробки тексту.

ML-сервер може працювати як окремий сервіс, у Docker-контейнері, у хмарному середовищі.

Для підвищення продуктивності, при необхідності, можливе використання GPU-прискорення, окремих серверів обчислень, кластерних рішень.

3.3.6 Сервер бази даних

Сервер бази даних забезпечує збереження результатів перевірок, зберігання інформації про користувачів, ведення журналів подій, накопичення статистики, збереження інформації про ML-моделі.

Для реалізації використовуються PostgreSQL, MySQL.

Сервер БД може бути як локальним так і віддаленим та хмарним.

Для забезпечення надійності використовуються резервне копіювання, реплікація, RAID-масиви, механізми журналювання транзакцій.

3.3.7 Контейнеризація та хмарне розгортання

Для спрощення процесу розгортання рекомендується використання контейнеризації.

Основні технології Docker, Docker Compose, Kubernetes.

Контейнеризація дозволяє ізолювати сервіси, спростити оновлення системи, швидко масштабувати компоненти, забезпечити переносимість між серверами.

Система може бути розгорнута як локально так і у приватній мережі і у хмарному середовищі AWS, Azure або Google Cloud.

3.3.8 Забезпечення безпеки під час розгортання

Оскільки система працює з потенційно небезпечними ресурсами, необхідно реалізувати HTTPS-з'єднання, SSL/TLS-сертифікати, firewall, JWT-автентифікацію, захист API, журналювання дій користувачів, моніторинг серверів.

Для захисту серверної інфраструктури рекомендується використання VPN, багаторівнева авторизація, регулярне оновлення ПЗ, антивірусний захист.

3.3.9 Масштабованість системи

Запропонована архітектура підтримує горизонтальне масштабування, додавання нових серверів, балансування навантаження, розподілену обробку запитів, хмарне масштабування ML-сервісів.

Це дозволяє системі ефективно працювати навіть при значному збільшенні кількості користувачів та обсягу перевірок.

3.3.10 Переваги обраної моделі розгортання

Основними перевагами запропонованої моделі є модульність, гнучкість, підтримка контейнеризації, висока продуктивність, масштабованість, підтримка хмарних сервісів, централізоване адміністрування, можливість інтеграції нових ML-моделей.

Таким чином, у підпункті було розглянуто процес розгортання інформаційної системи виявлення фішингових сайтів та підозрілих електронних листів на основі ML. Визначено основні серверні компоненти, принципи взаємодії між вузлами системи, особливості контейнеризації та хмарного розгортання. Запропонована модель розгортання забезпечує стабільну роботу системи, підтримку масштабування, ефективну взаємодію між компонентами та високий рівень захисту даних.

Висновки до розділу 3

У третьому розділі бакалаврської роботи було визначено основні вимоги до програмного та апаратного забезпечення інформаційної системи виявлення фішингових сайтів та підозрілих електронних листів на основі методів машинного навчання, а також розглянуто особливості розгортання системи та організації взаємодії її компонентів.

У підпункті, присвяченому вимогам до програмного забезпечення, було обґрунтовано вибір сучасних програмних засобів для реалізації системи. Визначено вимоги до операційних систем, серверного програмного забезпечення, клієнтської частини, ML-бібліотек та систем управління базами даних. Для реалізації серверної логіки запропоновано використання Python та FastAPI, а для побудови ML-модулів — бібліотек Scikit-learn, TensorFlow та інших інструментів машинного навчання. Також визначено вимоги до безпеки програмного забезпечення, підтримки REST API та контейнеризації.

У підпункті вимог до апаратного забезпечення було встановлено мінімальні та рекомендовані характеристики серверного обладнання, ML-сервера, клієнтських робочих станцій та мережевої інфраструктури. Було визначено вимоги до продуктивності обчислювальної системи, використання SSD-накопичувачів, багатоядерних процесорів та GPU-прискорення для ефективного виконання ML-алгоритмів. Також розглянуто питання масштабованості, резервного копіювання та забезпечення безперервної роботи системи.

У підпункті розгортання системи було розроблено модель фізичного розміщення компонентів інформаційної системи. Визначено основні вузли системи: клієнтський рівень, web-сервер, сервер прикладної логіки, ML-сервер та сервер бази даних. Було обґрунтовано використання клієнт-серверної архітектури, контейнеризації на основі Docker та можливості хмарного розгортання. Також розглянуто механізми забезпечення

інформаційної безпеки, балансування навантаження та масштабування системи.

Для візуалізації фізичної структури системи було розроблено UML Deployment Diagram, яка демонструє взаємодію між серверними компонентами, базою даних, ML-модулем та клієнтськими пристроями користувачів. Побудована діаграма дозволяє наочно представити процес функціонування системи у мережевому середовищі та підтверджує модульність і масштабованість запропонованого рішення.

Таким чином, у третьому розділі було сформовано технічну основу для практичної реалізації інформаційної системи виявлення фішингових сайтів та підозрілих електронних листів. Визначені програмні та апаратні вимоги, а також модель розгортання забезпечують стабільну роботу системи, підтримку ML-модулів, захищену обробку даних та можливість подальшого розвитку програмного продукту.

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

4.1 Загальні положення

Під час розроблення інформаційної системи виявлення фішингових сайтів та підозрілих електронних листів на основі методів машинного навчання значна частина робіт виконується із застосуванням комп'ютерної техніки та спеціалізованого програмного забезпечення. Робота програміста, аналітика даних або системного адміністратора належить до категорії розумової праці та пов'язана з тривалим перебуванням за персональним комп'ютером, що може негативно впливати на здоров'я працівника.

Основними завданнями охорони праці є забезпечення безпечних умов праці, збереження здоров'я працівників, попередження професійних захворювань, зниження ризику виробничого травматизму, створення комфортного робочого середовища.

Організація робочого місця повинна відповідати вимогам Закону України «Про охорону праці», ДСанПіН щодо роботи з персональними електронно-обчислювальними машинами, правил пожежної безпеки, вимог електробезпеки.

4.2 Аналіз умов праці під час роботи з інформаційною системою

Робота над розробленням інформаційної системи включає програмування, аналіз даних, тестування програмного забезпечення, роботу з базами даних, використання ML-моделей, адміністрування серверного середовища.

Основними небезпечними та шкідливими факторами є тривале статичне навантаження, перенапруження зору, електромагнітне випромінювання, психоемоційне навантаження, недостатнє освітлення, шум від комп'ютерної техніки, можливість ураження електричним струмом.

Найбільший вплив на працівника має тривала робота за монітором, що може спричиняти втому очей, головний біль, порушення постави, захворювання опорно-рухового апарату, зниження концентрації уваги.

4.3 Вимоги до організації робочого місця

Робоче місце користувача повинно забезпечувати ергономічні та безпечні умови праці.

Основні вимоги полягають в тому що площа робочого місця — не менше 6 м², об'єм приміщення — не менше 20 м³, відстань від очей до монітора — 50–70 см, верхній край монітора повинен бути на рівні очей, клавіатура повинна розташовуватись на відстані 10–15 см від краю столу, робоче крісло повинно бути регульованим.

Монітор необхідно розташовувати перпендикулярно до вікна, без відблисків, із достатнім рівнем яскравості та контрастності.

Для зменшення навантаження на хребет рекомендується використання ергономічного крісла, підтримка правильної постави, регулярні перерви під час роботи.

4.4 Вимоги до мікроклімату приміщення

Для комфортної роботи необхідно підтримувати оптимальні параметри мікроклімату.

Рекомендовані параметри такі як температура повітря: 20–24 °С, відносна вологість: 40–60%, швидкість руху повітря: до 0,2 м/с.

Приміщення повинно регулярно провітрюватися, мати систему вентиляції, забезпечувати достатню циркуляцію повітря.

Для підтримки комфортних умов, при необхідності, можуть використовуватись кондиціонери, зволожувачі повітря, системи клімат-контролю.

4.5 Освітлення робочого місця

Освітлення має велике значення під час роботи з комп'ютерною технікою.

Рекомендується використовувати природне освітлення, комбіноване штучне освітлення, LED-лампи з нейтральною температурою світла.

Нормований рівень освітленості робочого місця складає 300–500 лк.

Недостатнє або надмірне освітлення може призводити до швидкої втоми очей, погіршення зору, зниження продуктивності праці.

4.6 Електробезпека

Комп'ютерне обладнання належить до електроустановок, тому необхідно дотримуватися правил електробезпеки.

Основні вимоги до електроустановок це використання справної техніки, наявність заземлення, використання стабілізаторів напруги, заборона роботи з пошкодженими кабелями, відключення обладнання під час технічного обслуговування.

Для захисту від аварійних ситуацій рекомендується використання джерел безперебійного живлення (UPS), автоматичних вимикачів, мережевих фільтрів.

4.7 Пожежна безпека

У приміщенні, де експлуатується комп'ютерна техніка, необхідно забезпечити дотримання правил пожежної безпеки.

Основні причини виникнення пожеж це коротке замикання, перевантаження електромережі, несправність обладнання, порушення правил експлуатації техніки.

Для забезпечення пожежної безпеки необхідно використовувати сертифіковане обладнання, регулярно перевіряти електропроводку, не допускати перевантаження мережі, мати вогнегасники, забезпечити вільний доступ до евакуаційних виходів.

Рекомендовано використовувати порошкові або вуглекислотні вогнегасники.

4.8 Режим праці та відпочинку

Для зменшення навантаження на організм необхідно дотримуватись режиму праці та відпочинку.

Рекомендації перерва 10–15 хв після кожної години роботи, виконання вправ для очей, зміна положення тіла, короткі фізичні вправи.

Раціональна організація праці дозволяє зменшити втому, підвищити продуктивність, знизити ризик професійних захворювань.

4.9 Заходи щодо покращення умов праці

Для підвищення безпеки та комфорту праці рекомендується використання сучасної комп'ютерної техніки, застосування ергономічних меблів, автоматизація резервного копіювання, моніторинг стану серверного обладнання, використання антивірусного програмного забезпечення, регулярне оновлення програмного забезпечення.

Також доцільно проводити інструктажі з охорони праці, контролювати стан електромережі, організовувати навчання з пожежної безпеки.

Висновки до розділу 4

У розділі було розглянуто основні аспекти охорони праці під час розроблення інформаційної системи виявлення фішингових сайтів та підозрілих електронних листів на основі ML. Проаналізовано шкідливі та небезпечні фактори, які можуть виникати під час роботи з комп'ютерною технікою, а також визначено вимоги до організації робочого місця, мікроклімату, освітлення, електробезпеки та пожежної безпеки.

Запропоновані заходи дозволяють створити безпечні та комфортні умови праці для розробників і користувачів системи, знизити ризик професійних захворювань та забезпечити стабільну роботу програмно-апаратного комплексу.

ВИСНОВКИ

У бакалаврській роботі було розглянуто актуальну задачу розроблення інформаційної системи виявлення фішингових сайтів та підозрілих електронних листів на основі методів машинного навчання. Актуальність теми обумовлена стрімким зростанням кількості кіберзагроз, поширенням фішингових атак та необхідністю автоматизації процесів виявлення небезпечних вебресурсів і електронних повідомлень.

У процесі виконання роботи було проведено аналіз предметної області, досліджено сучасні методи та засоби виявлення фішингових ресурсів, визначено основні проблеми існуючих систем захисту та обґрунтовано необхідність використання методів машинного навчання для підвищення ефективності виявлення загроз. Було встановлено, що традиційні підходи до фільтрації небезпечного контенту мають низку недоліків, зокрема недостатню адаптивність до нових типів атак, залежність від сигнатурних баз та складність оперативного оновлення.

У першому розділі роботи було сформовано концепцію інформаційної системи, визначено її основні функції та вимоги до роботи. Також було розроблено математичне забезпечення системи, описано основні алгоритми класифікації та оцінювання ефективності моделей машинного навчання. У результаті було сформульовано постановку задачі та визначено основні етапи реалізації програмного продукту.

У другому розділі було виконано проектування системи. Розроблено загальносистемні рішення, визначено архітектуру програмного забезпечення та побудовано UML-діаграми, зокрема діаграми варіантів використання, послідовності, станів і розгортання. Також було спроектовано структуру бази даних, визначено основні сутності та взаємозв'язки між ними. Розроблена архітектура забезпечує модульність, масштабованість та можливість інтеграції сучасних ML-моделей.

У третьому розділі було визначено вимоги до програмного та апаратного забезпечення системи, обґрунтовано вибір серверних технологій,

ML-бібліотек, систем управління базами даних та засобів контейнеризації. Було розроблено модель розгортання системи, яка забезпечує стабільну роботу сервісів, захищену взаємодію між компонентами та підтримку масштабування в умовах зростання навантаження.

У четвертому розділі було розглянуто питання охорони праці під час роботи з комп'ютерною технікою та серверним обладнанням. Проаналізовано потенційно небезпечні фактори виробничого середовища, визначено вимоги до організації робочого місця, електробезпеки, пожежної безпеки та режиму праці користувачів і розробників системи.

У результаті виконання бакалаврської роботи було досягнуто поставленої мети — спроектовано інформаційну систему виявлення фішингових сайтів та підозрілих електронних листів на основі методів машинного навчання. Запропоноване рішення дозволяє автоматизувати процес аналізу вебресурсів та електронних повідомлень, підвищити швидкість виявлення загроз, зменшити ризик кібератак та забезпечити більш ефективний рівень інформаційної безпеки.

Практична цінність роботи полягає у можливості використання розроблених рішень для створення реальної системи кіберзахисту в організаціях, навчальних закладах або корпоративних мережах. Подальший розвиток роботи може бути пов'язаний із вдосконаленням ML-моделей, інтеграцією нейромережевих алгоритмів, використанням Big Data-технологій та розширенням функціональних можливостей системи.

СПИСОК ВИКОРИСТНИХ ДЖЕРЕЛ

1. Aggarwal, C. C. (2023). *Neural networks and deep learning: A textbook* (2nd ed.). Springer.
2. Alzahrani, A., Alenazi, M., & Alshammari, S. (2022). Machine learning techniques for phishing detection: A review. *Electronics*, 11(19), 3125. <https://doi.org/10.3390/electronics11193125>
3. Aurangzeb, K., Aleem, M., Iqbal, M. A., & Islam, M. A. (2021). A comprehensive review of phishing attack detection techniques using machine learning. *Sensors*, 21(16), 5605. <https://doi.org/10.3390/s21165605>
4. Chollet, F. (2021). *Deep learning with Python* (2nd ed.). Manning Publications.
5. Géron, A. (2022). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow* (3rd ed.). O'Reilly Media.
6. Goodfellow, I., Bengio, Y., & Courville, A. (2021). *Deep learning*. MIT Press.
7. Khonji, M., Iraqi, Y., & Jones, A. (2021). Phishing detection: A literature survey. *IEEE Communications Surveys & Tutorials*, 23(2), 1287–1310. <https://doi.org/10.1109/COMST.2021.3051354>
8. Mitchell, T. M. (2021). *Machine learning*. McGraw-Hill Education.
9. Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, 2(3), 160. <https://doi.org/10.1007/s42979-021-00592-x>
10. Scikit-learn Developers. (2024). *Scikit-learn: Machine learning in Python*. <https://scikit-learn.org>
11. TensorFlow Team. (2024). *TensorFlow documentation*. <https://www.tensorflow.org>

ДОДАТОК А. ТЕХНІЧНЕ ЗАВДАННЯ НА РОЗРОБКУ ІНФОРМАЦІЙНОЇ СИСТЕМИ ВИЯВЛЕННЯ ФІШИНГОВИХ САЙТІВ ТА ПІДОЗРЛИХ ЛИСТІВ НА ОСНОВІ ML

1 Загальні відомості

1.1 Найменування системи «Інформаційна система виявлення фішингових сайтів та підозрілих електронних листів на основі методів машинного навчання».

1.2 Підстава для розробки є тема бакалаврської роботи за спеціальністю

«Інформаційно управляючі системи та технології».

1.3 Розробник системи здобувач спеціальності «Інформаційно управляючі системи та технології» **Немченко Р.**

1.4 Призначення системи

Система призначена для:

- автоматизованого аналізу URL-адрес;
- виявлення фішингових вебресурсів;
- аналізу електронних листів;
- визначення рівня ризику повідомлень;
- використання ML-моделей для класифікації;
- збереження результатів перевірок;
- підвищення рівня інформаційної безпеки користувачів.

2 Мета розробки системи

Метою розробки є створення інформаційної системи, яка забезпечує автоматичне виявлення фішингових сайтів та підозрілих електронних листів із використанням алгоритмів машинного навчання для підвищення ефективності кіберзахисту.

3 Характеристика об'єкта автоматизації

Об'єктом автоматизації є процес:

- перевірки вебресурсів;
- аналізу електронних листів;

- класифікації потенційно небезпечного контенту;
- збереження результатів аналізу;
- моніторингу підозрілих дій.

Система повинна забезпечувати:

- швидку обробку запитів;
- автоматичне прогнозування рівня ризику;
- підтримку багатокористувацького режиму.

4 Вимоги до системи

4.1 Функціональні вимоги

Система повинна забезпечувати:

Для користувача:

- введення URL для перевірки;
- введення тексту електронного листа;
- отримання результатів аналізу;
- відображення рівня ризику;
- перегляд історії перевірок;
- авторизацію у системі.

Для адміністратора:

- керування користувачами;
- перегляд журналів подій;
- оновлення ML-моделей;
- адміністрування бази даних;
- керування доступом.

4.2 Нефункціональні вимоги

Система повинна забезпечувати:

- високу швидкодію;
- масштабованість;
- надійність;
- безпечність передачі даних;
- підтримку резервного копіювання;

- стабільну роботу при багатокористувацькому режимі.

Час обробки одного запиту:

- до 3 секунд для URL;
- до 5 секунд для електронного листа.

4.3 Вимоги до надійності

Система повинна:

- забезпечувати безперервну роботу;
- підтримувати резервне копіювання;
- відновлювати роботу після збоїв;
- вести журнал помилок та подій.

4.4 Вимоги до безпеки

Система повинна забезпечувати:

- автентифікацію користувачів;
- авторизацію за ролями;
- шифрування даних;
- HTTPS-з'єднання;
- захист від SQL-ін'єкцій;
- захист API;
- журналювання дій користувачів.

5 Вимоги до програмного забезпечення

Для реалізації системи необхідно використовувати:

Серверне програмне забезпечення:

- Python 3.11+;
- FastAPI;
- Uvicorn;
- Docker.

ML-бібліотеки:

- Scikit-learn;
- TensorFlow;
- Pandas;

- NumPy;
- NLTK.

База даних:

- PostgreSQL або MySQL.

Web-сервер:

- Nginx.

Клієнтська частина:

- HTML5;
- CSS3;
- JavaScript;
- React.

6 Вимоги до апаратного забезпечення

Мінімальні характеристики сервера:

- CPU: 4 ядра;
- RAM: 8 ГБ;
- SSD: 256 ГБ;
- мережа: 1 Gbit/s.

Рекомендовані характеристики:

- CPU: 8–16 ядер;
- RAM: 16–32 ГБ;
- NVMe SSD;
- GPU NVIDIA RTX для ML.

7 Вимоги до інформаційного забезпечення

Система повинна працювати з:

- URL-адресами;
- текстами електронних листів;
- наборами навчальних даних;
- журналами подій;
- інформацією про користувачів.

Формати даних:

- JSON;
- CSV;
- TXT;
- SQL.

8 Вимоги до організації робіт

Етапи розробки:

1. Аналіз предметної області;
2. Формування вимог;
3. Проектування архітектури;
4. Розробка бази даних;
5. Реалізація серверної частини;
6. Реалізація ML-модуля;
7. Розробка web-інтерфейсу;
8. Тестування системи;
9. Розгортання системи;
10. Підготовка документації.

9 Етапи тестування системи

Система повинна проходити:

- модульне тестування;
- інтеграційне тестування;
- функціональне тестування;
- навантажувальне тестування;
- тестування безпеки.

10 Вимоги до документації

У комплект документації повинні входити:

- технічне завдання;
- опис архітектури системи;
- UML-діаграми;
- інструкція користувача;
- інструкція адміністратора;

- опис бази даних;
- звіт із тестування.

11 Очікувані результати розробки

У результаті виконання роботи повинна бути створена інформаційна система, яка:

- автоматично визначає фішингові сайти;
- аналізує підозрілі електронні листи;
- використовує ML-алгоритми;
- забезпечує швидке отримання результатів;
- підтримує масштабування та модернізацію;
- підвищує рівень кібербезпеки користувачів.

ДОДАТОК Б. ІНСТРУКЦІЯ КОРИСТУВАЧА ІНФОРМАЦІЙНОЇ СИСТЕМИ ВИЯВЛЕННЯ ФІШИНГОВИХ САЙТІВ ТА ПІДОЗРІЛИХ ЛИСТІВ НА ОСНОВІ ML

1 Загальні відомості

Інформаційна система виявлення фішингових сайтів та підозрілих електронних листів призначена для автоматизованого аналізу вебресурсів та електронних повідомлень із використанням методів машинного навчання.

Система дозволяє:

- перевіряти URL-адреси;
- аналізувати тексти електронних листів;
- визначати рівень ризику;
- отримувати результати ML-класифікації;
- переглядати історію перевірок;
- адмініструвати користувачів та систему.

2 Призначення системи

Система використовується для:

- виявлення фішингових вебсайтів;
- аналізу підозрілих електронних листів;
- автоматичного визначення небезпечного контенту;
- підвищення рівня інформаційної безпеки користувачів.

3 Системні вимоги

Для роботи із системою необхідно:

Мінімальні вимоги:

- операційна система Windows 10 / Linux / macOS;
- оперативна пам'ять — від 4 ГБ;
- сучасний браузер;
- доступ до мережі Інтернет.

Рекомендовані браузери:

- Google Chrome;
- Mozilla Firefox;

- Microsoft Edge;
- Opera.

4 Вхід у систему

Для початку роботи необхідно:

1. Відкрити веббраузер;
2. Перейти за адресою системи;
3. Увести логін та пароль;
4. Натиснути кнопку «Увійти».

Після успішної авторизації користувач переходить до головного меню системи.

5 Головне меню системи

Головне меню містить:

- модуль перевірки URL;
- модуль аналізу електронних листів;
- історію перевірок;
- профіль користувача;
- налаштування системи;
- адміністративний модуль (для адміністратора).

6 Перевірка URL-адреси

Для перевірки вебресурсу необхідно:

1. Відкрити вкладку «Перевірка URL»;
2. Ввести URL-адресу сайту;
3. Натиснути кнопку «Перевірити».

Після обробки система відображає:

- рівень ризику;
- результат класифікації;
- ознаки фішингу;
- рекомендації користувачу.

Приклад результату перевірки. Таблиця Б.1

Параметр	Значення
URL	http://example-site.com
Статус	Підозрілий
Рівень ризику	Високий
ML-класифікація	Phishing
Рекомендація	Не переходити за посиланням

7 Аналіз електронного листа

Для перевірки електронного листа необхідно:

1. Перейти до вкладки «Аналіз листів»;
2. Вставити текст листа;
3. Натиснути кнопку «Аналізувати».

Система виконує:

- обробку тексту;
- пошук підозрілих ознак;
- ML-класифікацію;
- оцінку ризику.

Результати аналізу листа

Користувач отримує:

- статус повідомлення;
- рівень небезпеки;
- список підозрілих фрагментів;
- рекомендації щодо безпеки.

8 Історія перевірок

У розділі «Історія» користувач може:

- переглядати попередні перевірки;
- шукати результати;
- сортувати записи;

- переглядати дату та час перевірки.

Історія містить:

- URL або текст листа;
- результат перевірки;
- рівень ризику;
- дату аналізу.

9 Робота адміністратора

Адміністратор системи має доступ до:

- керування користувачами;
- перегляду журналів подій;
- оновлення ML-моделей;
- резервного копіювання;
- налаштування системи.

Основні функції адміністратора

Адміністратор може:

- створювати нових користувачів;
- блокувати облікові записи;
- змінювати права доступу;
- переглядати системні журнали;
- запускати оновлення моделей машинного навчання.

10 Повідомлення системи

Система може відображати такі повідомлення. Таблиця Б. 2

Повідомлення	Опис
URL Safe	Сайт безпечний
Suspicious URL	Виявлено підозрілі ознаки
Phishing Detected	Виявлено фішинговий ресурс
Mail Analyzed	Аналіз листа завершено
Access Denied	Недостатньо прав доступу

11 Рекомендації з безпеки

Під час роботи із системою рекомендується:

- не відкривати підозрілі посилання;
- не вводити конфіденційні дані на невідомих сайтах;
- регулярно змінювати пароль;
- використовувати двофакторну автентифікацію;
- перевіряти електронні листи перед відкриттям вкладень.

12 Можливі помилки та способи їх усунення

Проблема	Причина	Спосіб усунення
Неможливо увійти	Невірний пароль	Перевірити логін і пароль
Не виконується аналіз	Відсутнє з'єднання	Перевірити Інтернет
Помилка сервера	Перевантаження системи	Повторити запит пізніше
Не завантажується сторінка	Проблеми браузерa	Оновити браузер

13 Завершення роботи

Для завершення роботи необхідно:

1. Натиснути кнопку «Вийти»;
2. Закрити браузер.

Це дозволяє уникнути несанкціонованого доступу до системи.

14 Технічна підтримка

У разі виникнення помилок або проблем користувач повинен звернутися до адміністратора системи або служби технічної підтримки.

Інструкція користувача описує основні принципи роботи з інформаційною системою виявлення фішингових сайтів та підозрілих електронних листів на основі ML. Використання системи дозволяє автоматизувати процес перевірки вебресурсів та електронних повідомлень,

підвищити рівень інформаційної безпеки та зменшити ризик фішингових атак.

ДОДАТОК В. ЕКРАННІ ФОРМИ РОЗРОБКИ

Екранна форма 1 — Перевірка URL-адреси, рисунок В.1

Дана екранна форма призначена для аналізу вебпосилань на наявність ознак фішингу. Користувач вводить URL-адресу у відповідне поле та запускає перевірку. Система відображає результат аналізу, рівень ризику, ML-класифікацію та перелік виявлених підозрілих ознак. Також форма містить рекомендації щодо подальших дій користувача та детальну інформацію про перевірений сайт.

Екранна форма 2 — Аналіз електронного листа, рисунок В.1

Дана форма використовується для перевірки тексту електронного листа на наявність фішингових або шахрайських ознак. Користувач вставляє текст листа у спеціальне поле, після чого система виконує ML-аналіз повідомлення. У результаті відображаються рівень небезпеки, класифікація листа, знайдені підозрілі елементи та рекомендації щодо безпечної роботи з повідомленням. Форма також дозволяє переглядати виявлені посилання та службову інформацію про лист.

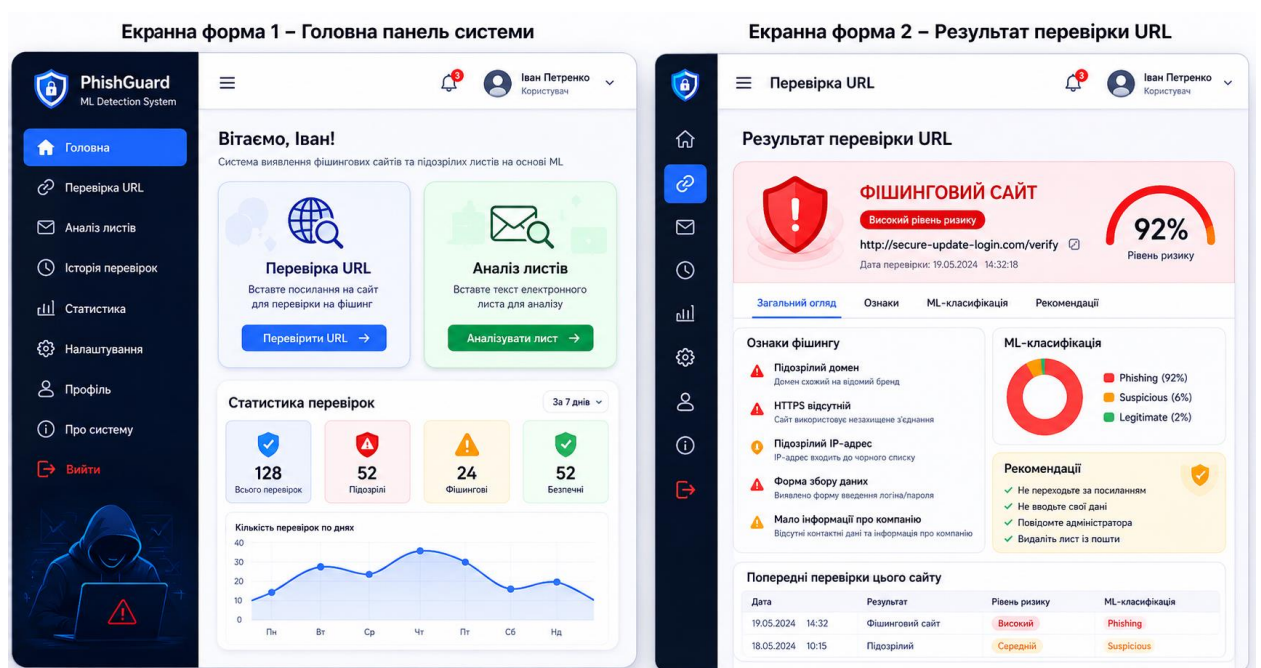


Рисунок В.1 -Екранні форми розробки