Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій

Бакалаврська дипломна робота на тему:

«ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПЕРЕДБАЧЕННЯ ФІШИНГОВИХ САЙТІВ»

Виконав: студент 4 курсу, групи 2ІСТ-20б спеціальності 126 – «Інформаційні системи та технології»
_____Данило ЛИТВИНЕНКО
Керівник: к.т.н., доц. каф. САІТ
_____Олексій КОЗАЧКО
«31» 05 2024 р.
Рецензент: к.т.н., доц. каф.
_____Володимир ОЗЕРАНСЬКИЙ
«10» 06 2024 р.

Допущено до захисту

Завідувач кафедри САІТ

_____ д.т.н., проф. Віталій МОКІН

«03» 06 2024 р.

Вінниця ВНТУ – 2024 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій
Рівень вищої освіти – перший (бакалаврський)
Галузь знань – 12 Інформаційні технології
Спеціальність 126 – «Інформаційні системи та технології»
Освітньо-професійна програма – Прикладні інформаційні технології

ЗАТВЕРДЖУЮ
Завідувач кафедри САІТ
_____ д.т.н., проф. Віталій МОКІН
"05" 03 2024 року

ЗАВДАННЯ
НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ
Литвиненку Данилу Олександровичу

1. Тема роботи: «Інформаційна технологія передбачення фішингових сайтів»
керівник роботи: Козачко О. М., к.т.н., доц. каф. САІТ
затверджені наказом ВНТУ від «11» 05 2024 року №80
2. Термін подання студентом роботи 31.05
3. Вихідні дані до роботи:
1) Набір даних фішингу веб-сторінок.
4. Зміст текстової частини:
1) Загальна характеристика об'єкту досліджень;
2) Вибір та аналіз оптимальних інформаційних технологій;
3) Розроблення інформаційної технології для передбачення фішингових сайтів.
5. Перелік ілюстративного матеріалу:
1) Огляд даних датасету;
2) Візуалізація даних датасету;
3) Розвідувальний аналіз даних;
4) Блок-схема алгоритму інформаційної технології;
5) Результат найкращої моделі.

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 11.03

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Аналіз предметної області	11.03	31.03	вик
2	Вибір оптимальних інформаційних технологій	01.04	15.04	вик
3	Розвідувальний аналіз даних	16.04	30.04	вик
4	Розроблення інформаційної технології	01.05	15.05	вик
5	Оформлення матеріалів до захисту БДР	16.05	31.05	вик

Студент _____ Данило ЛИТВИНЕНКО

Керівник роботи _____ Олексій КОЗАЧКО

АНОТАЦІЯ

Бакалаврська дипломна робота складається з 69 сторінок формату А4, на яких є 34 рисунка, та 20 найменувань використаних джерел літератури.

Метою роботи є розроблення інформаційної технології передбачення фішингових сайтів.

Застосовано методи машинного навчання та аналізу даних для виявлення підозрілих веб-ресурсів. Розроблено функціонал інформаційної технології на платформі Kaggle. Інформаційна технологія дозволяє автоматизувати процеси виявлення фішингових сайтів, що значно підвищує рівень безпеки користувачів в інтернеті.

Ключові слова: фішинг, інформаційна технологія, веб-сторінки, кібербезпека, шахрайство.

ABSTRACT

The bachelor's thesis consists of 69 A4 pages with 34 figures and 20 references.

The aim of the work is to develop an information technology for predicting phishing sites.

Machine learning and data analysis methods are used to identify suspicious web resources. The functionality of an information system based on the Kaggle platform was developed and tools for analyzing web pages were integrated. The information system allows to automate the processes of detecting phishing sites, which significantly increases the level of user security on the Internet.

Keywords: phishing, information system, web pages, cybersecurity, fraud.

ЗМІСТ

ВСТУП

Актуальність теми. Актуальність роботи досліджень фішингових сайтів обумовлена постійним збільшенням числа подібних загроз в цифровому просторі. Незважаючи на зростаючу обізнаність користувачів про методи фішингу, зловмисники постійно вдосконалюють свої методи, що ускладнює їх виявлення. Людський фактор продовжує залишатися ключовим фактором, який кіберзлочинці використовують для досягнення своїх цілей. Навіть сучасні системи безпеки не завжди запобігають помилкам користувачів, які можуть стати жертвами фішингових атак через недбалість або незнання.

Фішинг – це форма інтернет-шахрайства, яка стала досить поширеною в останні роки. За цією схемою зловмисники, вдаючись до соціальної інженерії, використовують підроблені веб-сайти та електронні повідомлення, щоб виманити конфіденційну інформацію від потенційних жертв. У цьому контексті важливо розрізняти фішингові атаки від достовірних запитів, щоб уникнути неприємних наслідків для особистої безпеки та фінансового стану [1].

Для виявлення фішингових сайтів в роботі пропонується використати такі методи машинного навчання як логістична регресія, дерево рішень, випадковий ліс та модель градієнтного бустингу.

Мета і задачі дослідження. Метою дослідження є розроблення інформаційної технології для передбачення фішингових сайтів.

Для досягнення поставленої мети необхідно розв’язати такі задачі:

– виконати аналіз проблеми фішингових сайтів та методів їх виявлення;

– здійснити вибір оптимального рішення та налаштувань для розв’язання задачі;

– розробити технологію для аналізу та виявлення фішингових сайтів з використанням методів машинного навчання та аналізу даних.

Об’єктом дослідження є процес створення інформаційної технології передбачення фішингових сайтів.

Предметом дослідження є моделі та методи для аналізу та передбачення фішингових сайтів.

Публікації. За результатами даної роботи була зроблена доповідь на тему «Інформаційна технологія передбачення фішингових сайтів» на міжнародній науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи (Вінниця, 2023-2024 рр.)» з публікацією тез [2].

1 ЗАГАЛЬНА ХАРАКТЕРИСТИКА ОБ’ЄКТУ ДОСЛІДЖЕНЬ

1.1 Опис об’єкта досліджень

В умовах науково-технічного прогресу інформація стає об'єктом певних суспільних відносин, що виникають з моменту її створення, в процесі накопичення, зберігання, обробки і використання, формування, переробки і використання інформаційних ресурсів товарного вигляду відбувається в процесі комп'ютеризації різних галузей людської діяльності. діяльність, що здійснюється з використанням сучасних інформаційних технологій. Однак використання сучасних інформаційних технологій, крім усього позитивного в рамках розвитку суспільства, вимагає інтенсивного впровадження використання сучасних комп'ютерних технологій в корисливих цілях, особливо в кредитній і банківській діяльності, що призвело до появи нового класу злочинів в області комп'ютерної злочинності [3].

У той же час злочинці видають підроблені веб-сайти за добре відомі та надійні веб-ресурси. Для цього максимально скопіюйте дизайн та URL-адресу та додайте лише 1 або 2 символи, які відрізняють його від оригіналу. Неозброєним оком різницю помітити дуже складно.

Щоб зрозуміти, що таке фішинговий сайт, досить згадати велику кількість спам-листів, в яких обіцялися призи, як тільки ви перейдете за посиланням і введете свої особисті дані. Звичайно, якщо ви йому довіряєте, злочинець дійсно отримає приз. Така схема здійснюється за допомогою телефонних дзвінків, повідомлень і навіть онлайн-реклами.

Крім того, існує безліч причин: підтвердження схвалення на сайті, відписка від спаму, установка дуже зручних програм, значні знижки на товари або їх подарунки. Близько 80% підроблених веб-ресурсів пропонують послуги з поповнення мобільного рахунку. Зрозуміло, кошти ви не отримаєте.

Жодна країна світу не застрахована від комп'ютерних злочинів. Шахрайство є найбільш поширеним явищем на сучасному етапі комп'ютеризації суспільства.

Термін "Інтернет-шахрайство" зазвичай відноситься до будь-якого виду шахрайства, при якому один або кілька елементів Інтернету (чати, електронна пошта, дошки оголошень або веб-сайти) використовуються для залучення потенційних жертв, здійснення шахрайських операцій або надсилання повідомлень від шахраїв фінансовим установам або іншим особам [4].

Проблема шахрайства в Інтернеті стає дуже актуальною в міру розвитку ситуації. Дані, накопичені Інтерполом, підтверджують той факт, що в останні роки World Wide Web став зоною дуже активного зростання злочинності в порівнянні з усіма існуючими галузями людської діяльності [5]. На рисунку 1.1 продемонстровано дія фішингу.

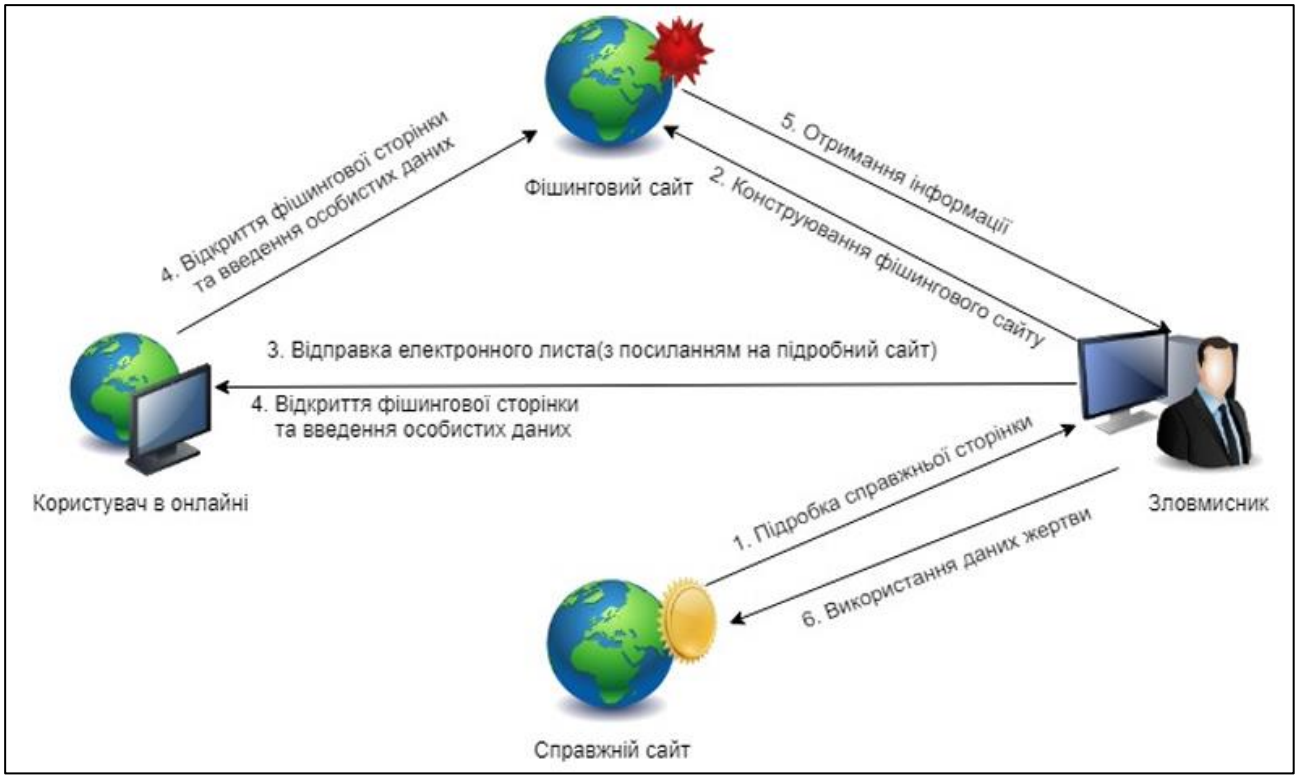

Рисунок 1.1 – Приклад дії фішингу

В Україні також зафіксована нова форма мобільного фішингу: шахраї добровільно переводять гроші зі свого рахунку на особистий рахунок іншого шахрайського абонента.

Про це повідомив Віктор Гоцуленко, експерт зі зв'язків з громадськістю "Київстар".

Суть афери полягає в тому, що шахрай відправляє потенційній жертві електронне повідомлення і відкриває популярний веб-сайт про можливість поповнення рахунку, що завдасть шкоди оператору мобільного зв'язку.

При цьому пропонується впровадити спеціальний набір USSD-команд, який призведе до зняття грошей з рахунків довірливих абонентів, але при цьому постраждають самі абоненти.

Шахраї намагаються заохочувати перекази невеликих сум 10-20 гривень, щоб вони не завдавали істотної шкоди абоненту, але деякі експерти стверджують, що такі гроші того не варті.

Anti-Phishing Working Group(APWG) аналізує фішингові атаки та інші методи крадіжки персональних даних, про які APWG отримує інформацію від компаній-членів, глобальних дослідницьких партнерів, а також електронною поштою. APWG вимірює еволюцію, поширення та розповсюдження методів крадіжки персональних даних, спираючись на дослідження наших компаній-членів та галузевих експертів. Фішинг – це злочин, що використовує як соціальну інженерію, так і технічні хитрощі для викрадення персональних даних споживачів та облікових даних фінансових рахунків.

Схеми соціальної інженерії полюють на необережних жертв, змушуючи їх повірити, що вони мають справу з надійною, законною стороною, наприклад, за допомогою обманних адрес електронної пошти та електронних повідомлень. Вони призначені для того, щоб привести споживачів на підроблені веб-сайти, які обманом змушують одержувачів розголошувати фінансові дані, такі як імена користувачів і паролі. Схеми технічного обману встановлюють шкідливе програмне забезпечення на комп'ютери для безпосередньої крадіжки облікових даних, часто з використанням систем, які перехоплюють імена користувачів та паролі облікових записів або перенаправляють користувачів на підроблені веб-сайти [6].

Огляд тенденцій фішингової активності:

– Телефонний фішинг, який безпосередньо залучає жертв, безконтрольно поширюється. Телефонні номери, що використовуються для

шахрайства, становили понад 20% активів, пов'язаних з шахрайством, виявлених OpSec у першому кварталі 2024 року;

– Фішинг з використанням телефонних дзвінків – так званий голосовий фішинг або «вішинг» – зростає з кожним кварталом;

– У 1 кварталі 2024 року APWG зафіксувала 963 994 фішингові атаки, що є найнижчим квартальним показником з 4 кварталу 2021 року;

– Платформи соціальних мереж були найбільш часто атакованим сектором, на який було спрямовано 37,4% всіх фішингових атак в 1 кварталі 2024 року. Кількість фішингових атак у банківському сегменті продовжила знижуватися - до 9,8%;

– Середня сума банківського переказу, запитувана в BEC-атаках у першому кварталі 2024 року, становила $84 059, що майже на 50% вище за середній показник попереднього кварталу.

APWG відстежує:

– Унікальні фішингові сайти. Це основний показник фішингу, про який повідомляється в усьому світі. Він визначається за унікальними базами фішингових URL-адрес, знайдених у фішингових електронних листах, про які повідомляється в репозиторії APWG. (Один фішинговий сайт може рекламуватися як тисячі індивідуальних URL-адрес, які ведуть до одного й того ж місця призначення). Таким чином, APWG вимірює фішингові сайти, про які повідомляється, що є більш релевантною метрикою, ніж URL-адреси;

– Унікальні теми фішингових листів. Враховуються електронні листи-приманки, які мають різні теми листів. Деякі фішингові кампанії можуть використовувати однакові теми, але рекламувати різні фішингові сайти. Цей показник є загальним показником різноманітності фішингових атак і може бути приблизною оцінкою кількості фішингових атак;

– APWG також підраховує кількість атакованих брендів, вивчаючи звіти про фішинг, що надходять на електронну біржу злочинів APWG, і нормалізує написання назв брендів.

В таблиці 1.1 представлено загальну кількість фішингових атак

Таблиця 1.1. Загальна кількість фішингових атак

Назва	Січень	Лютий	Березень
Кількість виявлених унікальних фішингових сайтів (атак)	358,107	314,974	290,913
Унікальні фішингові розсилки	50,837	24,086	41,550
Кількість брендів, на які спрямовані фішингові компанії	314	309	301

На рисунку 1.2 показано графік зафіксованих фішингових атак з 2023 по 2024 роки.

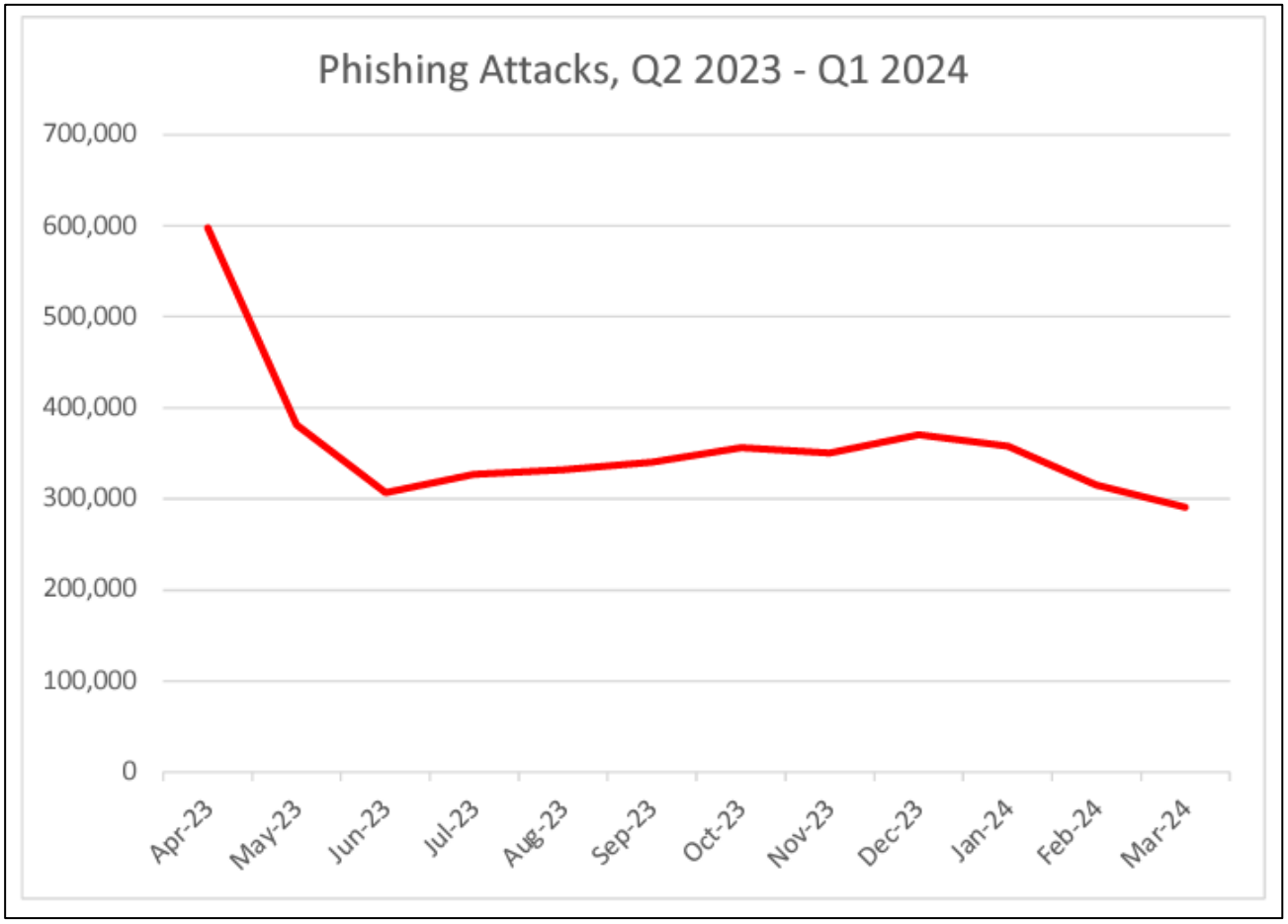

Рисунок 1.2 – Графік фішингових атак

Член-засновник APWG, компанія OpSec Security, виявила, що кількість телефонних номерів, які використовуються для здійснення шахрайських дій, різко зросла за останні три роки. Телефонні номери, що використовуються для

шахрайства, становлять понад 20 відсотків усіх активів, пов'язаних з шахрайством, які OpSec виявила в першому кварталі 2024 року. OpsSec підраховує шахрайські активи, включаючи шахрайські URL-адреси (наприклад, фішингові URL-адреси), телефонні номери, які використовуються в шахрайстві, та електронні поштові скриньки, які використовуються для здійснення шахрайства (включаючи ті, що використовуються для BEC-атак, шахрайства з оголошеннями про роботу тощо).

На рисунку 1.3 представлено графік зростання телефонного фішингу з 2021 року до 2023 року

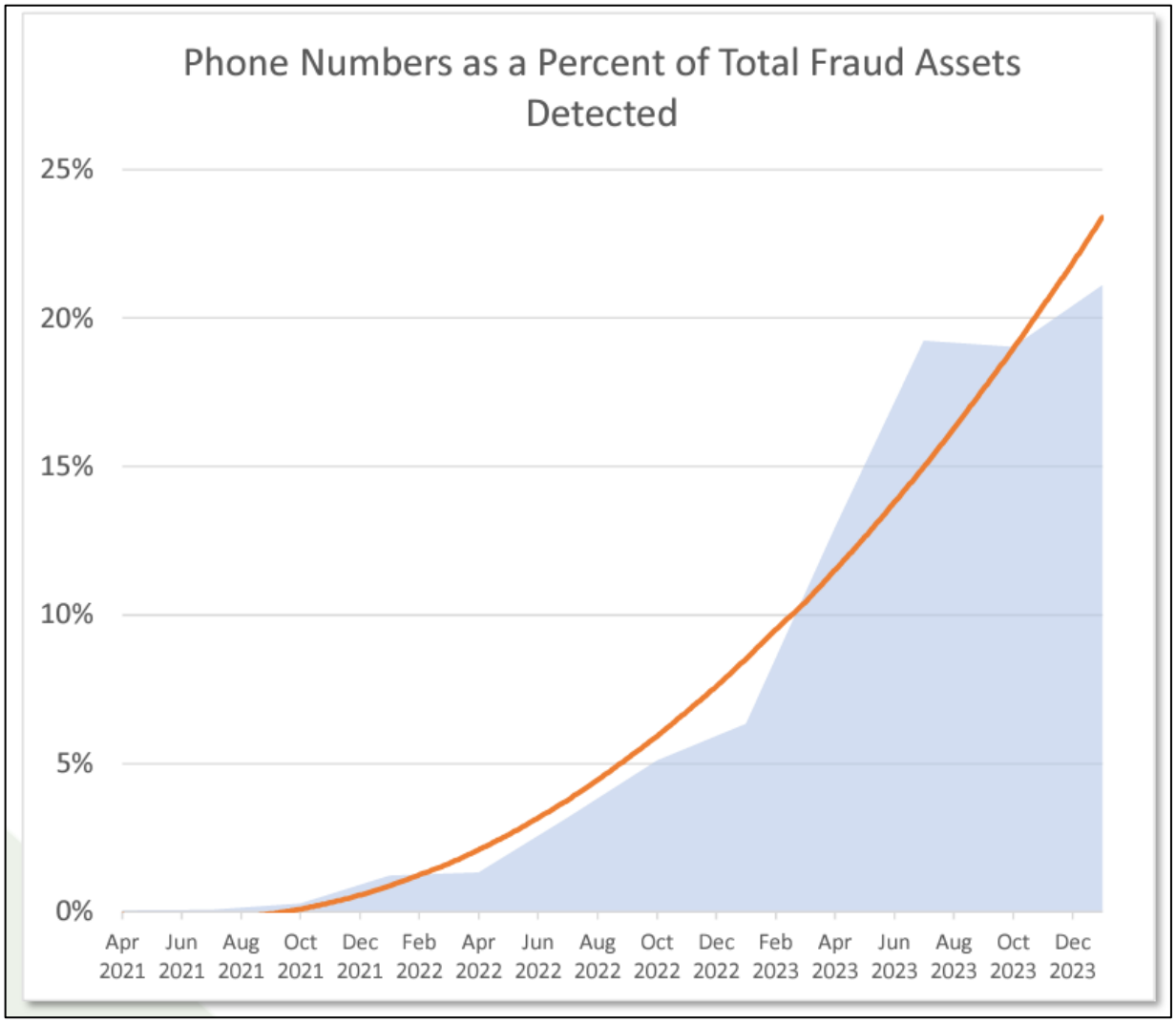

Рисунок 1.3 – Графік зростання телефонного фішингу

На рисунку 1.4 представлено графік розподілу фішингових атак в різних сферах.

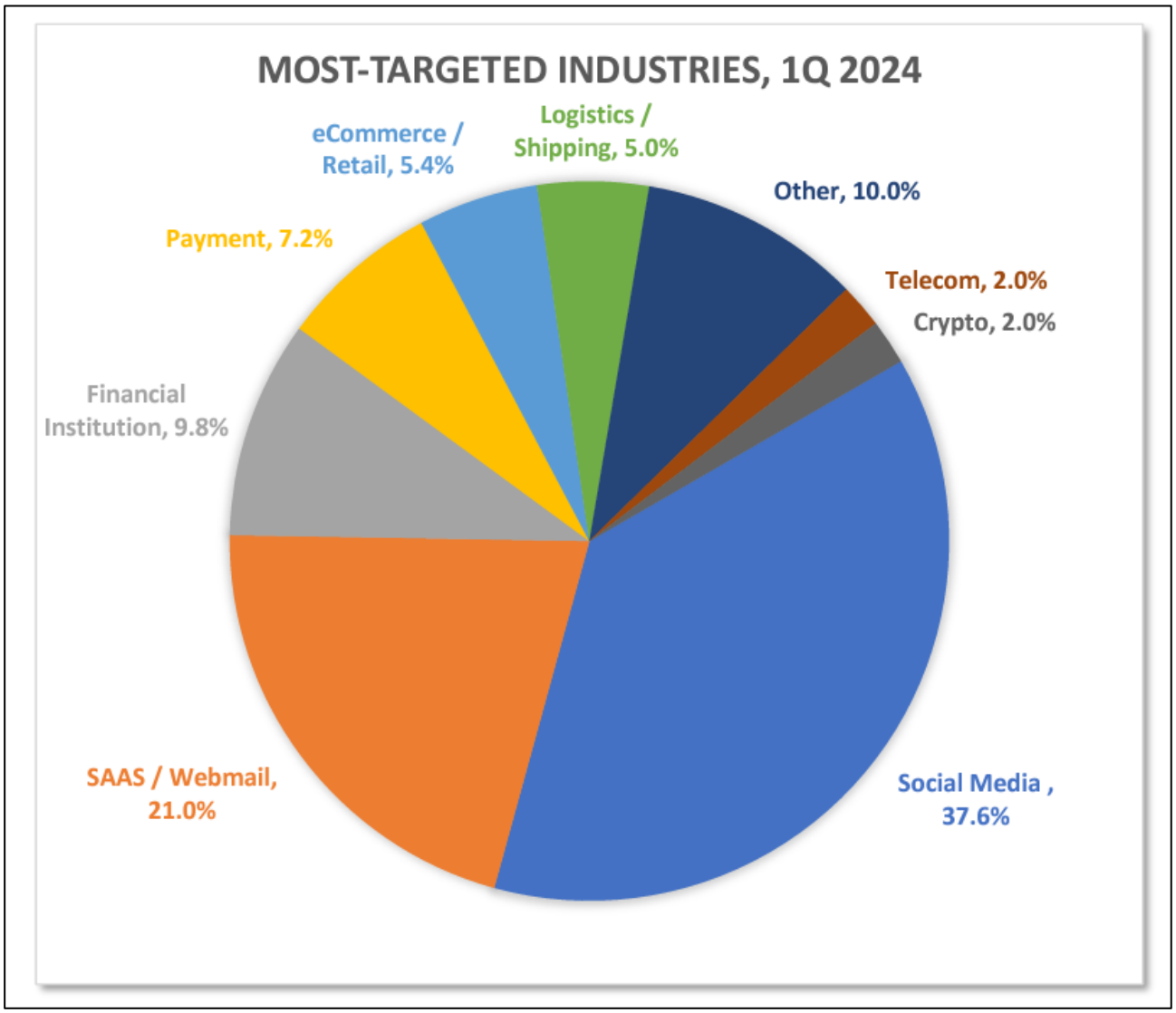

Рисунок 1.4 – Графік розподілу фішингу

«Майже 60% шкідливих повідомлень, що надійшли на корпоративні поштові скриньки в першому кварталі 2024 року, намагалися викрасти облікові дані для входу в систему, тоді як 40% були засновані на реагуванні», - сказав Джон Вілсон, старший науковий співробітник відділу досліджень загроз компанії Fortra. «Менше половини відсотка шкідливих повідомлень, які потрапляли в корпоративні поштові скриньки, намагалися доставити шкідливе програмне забезпечення. Ці цифри свідчать про те, що корпоративним поштовим фільтрам все ще важко відловлювати фішинг з використанням облікових даних та повідомлення про атаки, засновані на відповіді» [7].

На рисунку 1.5 представлено графік розподілу фішингових атак на окремі сервіси.

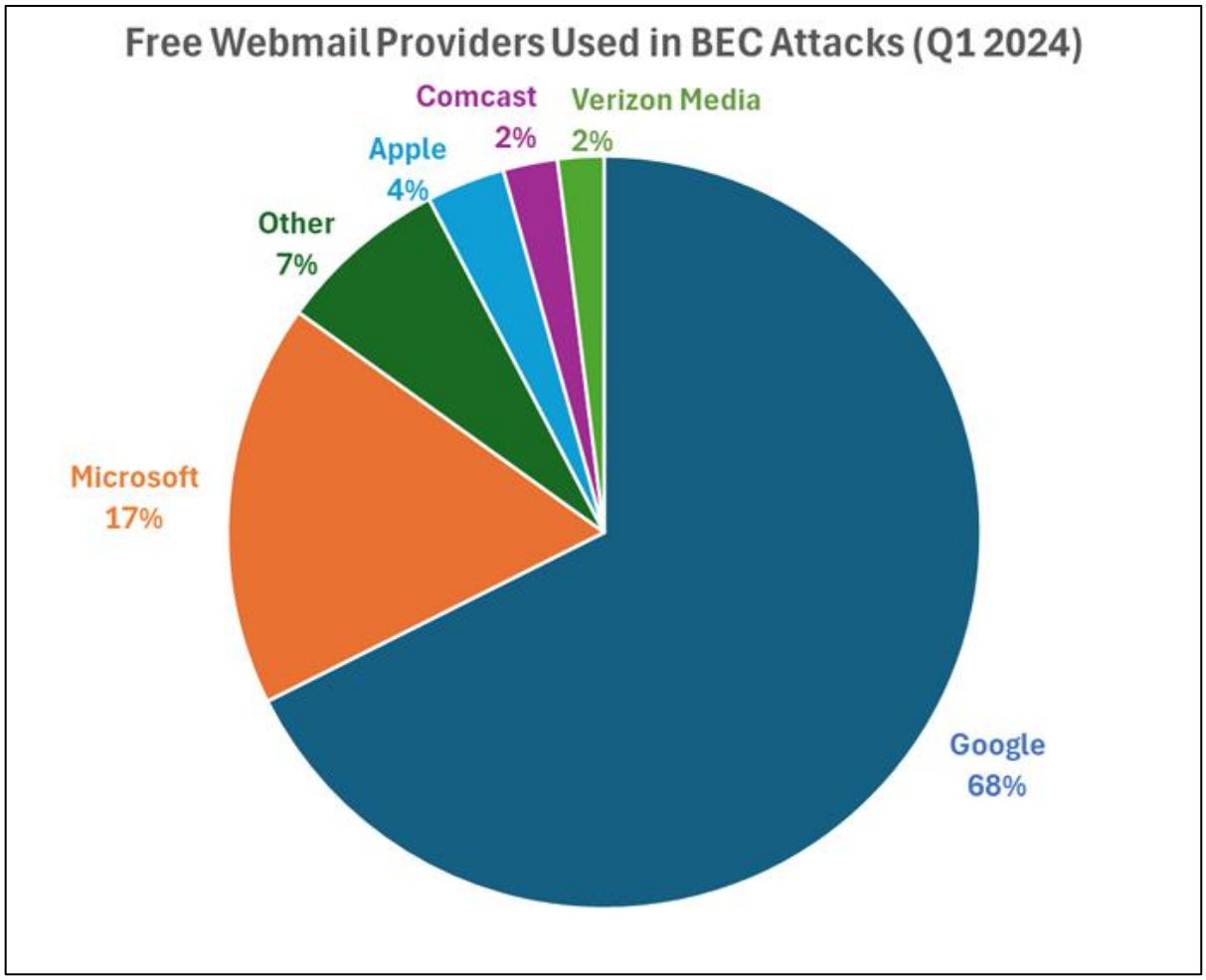

Рисунок 1.5 – Графік розподілу фішингу окремих сервісів

Fortra виявила, що 73% BEC-атак у першому кварталі 2024 року були розпочаті з використанням безкоштовних поштових доменів, що трохи більше, ніж у попередньому кварталі (68%). Решта 27% BEC-атак у першому кварталі 2024 року використовували комбінацію зловмисно зареєстрованих доменів та скомпрометованих поштових акаунтів. Google був найпопулярнішим провайдером безкоштовної веб-пошти серед шахраїв, на частку якого припадає 68% акаунтів безкоштовної веб-пошти, що використовувалися в першому кварталі 2024 року в шахрайських діях. На поштові сервіси Microsoft припало 17% атак з використанням веб-пошти в першому кварталі, за ними йшов довгий хвіст інших провайдерів веб-пошти.

На рисунку 1.6 представлено графік розподілу реєстраторів, які використовують для реєстрації доменів BEC.

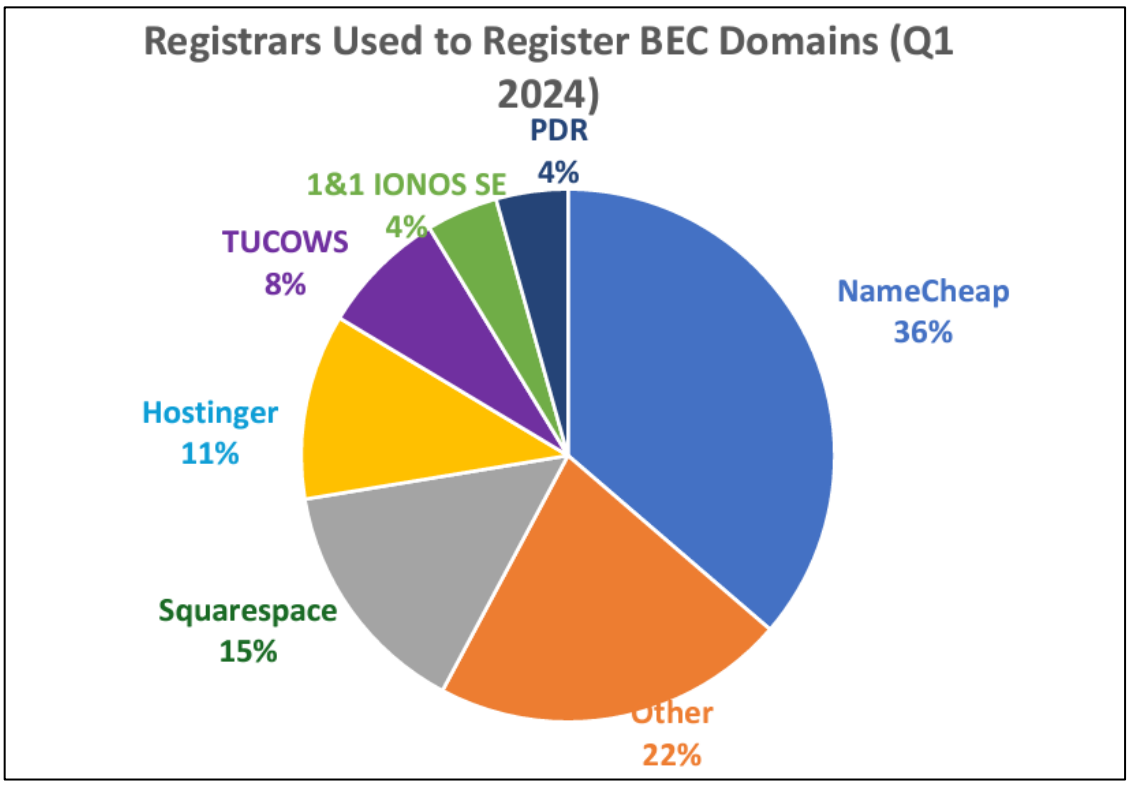

Рисунок 1.6 – Графік розподілу реєстраторів домену

1.2 Види та методи розпізнавання фішингових сайтів

На сьогоднішній день ми можемо виділити 3 види фішингу: поштовий, онлайн і комбінований.

Перший – найстаріший, який полягає у відправці жертві спеціального електронного листа з проханням надіслати відповідну інформацію. І як наочний приклад можемо навести найпримітивніший спосіб отримання даних для доступу в інтернет. Зловмисник просто представляється співробітником провайдера інтернет-користувача і розповідає історію про збій в роботі бази даних, а для ще більшої переконливості, повідомляючи останньому своє ім'я користувача, шахрай може використовувати поштову скриньку з назвою сервера, схожим на назву сервера провайдера.

В рамках дослідження проблеми поширення шахрайства в Інтернеті Інститут Ponemon, який займається питаннями конфіденційності та громадянських свобод, опитав 1335 американських інтернет-користувачів і з'ясував, що 76% респондентів знайомі з електронною поштою обману. 70% з цього числа відвідали підроблені веб-сайти, на які посилаються підробленими листами, 15% залишили на цьому сайті особисті дані і 2% зазнали економічних втрат в результаті фішингу. Крім того, Інститут Ponemon у своєму звіті називає суму в 500 мільйонів доларів збиток, який можуть понести інтернет-користувачі в результаті такого роду шахрайства.

Згідно з опитуванням Javelin Strategy & Research, в якому взяли участь 4000 осіб, шахраї все частіше використовують Інтернет для крадіжки персональних даних.

Онлайновий фішинг. Існує кілька сайтів, на яких проводяться онлайн-фішингові операції, афери з цінними паперами, магазини. У той же час вони використовують подібний дизайн ідентифікації доменного імені.

Число жертв таких злочинних дій постійно зростає, оскільки зловмисники часто пропонують різні товари буквально за демпінговими цінами. А підозри, які могли виникнути у потенційних жертв, були розвіяні популярністю скопійованих сайтів.

Тому користувачі інтернету вважають, що вони перебувають на веб-сервері в реальному магазині з хорошою репутацією. Прийнявши рішення про покупку товару, інтернет-користувач реєструється в системі. При цьому, цілком ймовірно, що для цього він буде використовувати той же пароль, що і для інших сервісів. Крім того, для придбання продукту користувачі вводять номер своєї пластикової картки та інші дані. Це саме те, що потрібно зловмиснику, і він може негайно скористатися кредитною карткою жертви.

Обидва описаних виду фішингу існують вже давно. Вони досить примітивні, тому уважні та скептичні люди, особливо останнім часом, через загальний рівень освіти користувачів Інтернету в галузі інформаційної безпеки дещо змінився, тому Електронна пошта та онлайн-фішинг перетворилися на неефективний вид шахрайства в Інтернеті.

Але на зміну їм прийшов 3-й тип фішингу (комбінований), який практично відразу став популярним. Суть цього методу обману полягає в тому, що зловмисник створює підроблений веб-сайт організації та заманює користувача на цей веб-сайт за допомогою приманки. Головна небезпека комбінованого фішингу полягає в його високій актуальності. І справді, ніхто не хоче, щоб людина надсилала пароль електронною поштою. Робоча група з боротьби з фішингом (APWG) назвала цей тип шахрайства "фермерством".

Серед численних технологій, представлених на ринку для забезпечення безпеки інформаційних технологій від фішингових атак неможливо виділити найбільш ефективну і надійну з них. Ось представлено найбільш популярні та масові інформаційні технології з забезпеченням безпеки користувачів:

– Сервіз забезпечення безпеки – Google Safe Browsing, Yandex Safe Browsing, OpenDNS, Phishtank;

– Програмне забезпечення для забезпечення безпеки у вигляді встановленого додатка-антивірусне програмне забезпечення, додаток для веб-браузера.

Google Safe Browsing допомагає захищати понад п’ять мільярдів пристроїв щодня, показуючи попередження користувачам, коли вони намагаються перейти на небезпечні сайти або завантажити небезпечні файли. Безпечний перегляд також сповіщає веб-майстрів, коли їхні веб-сайти скомпрометовані зловмисниками, і допомагає їм діагностувати та вирішити проблему, щоб їхні відвідувачі залишалися в безпеці. Захист безпечного перегляду працює в усіх продуктах Google і забезпечує безпечніший перегляд в Інтернеті [8].

Безпечний перегляд був запущений у 2005 році, щоб захистити користувачів у мережі від фішингових атак, і розвинувся, щоб надати користувачам інструменти для захисту від веб-загроз, таких як зловмисне програмне забезпечення, небажане програмне забезпечення та соціальна інженерія на настільних і мобільних платформах.

Користувачі, які потребують або бажають вищого рівня безпеки під час перегляду веб-сторінок, можуть увімкнути Enhanced Safe Browsing.

Користувачі, які налаштували Enhanced Safe Browsing для свого облікового запису Google або браузера Chrome, отримають найвищий рівень захисту.

У Google Chrome користувачі Enhanced Safe Browsing отримають переваги від таких додаткових засобів захисту:

– Перевірки в режимі реального часу зі списками відомих фішингових і шкідливих сайтів;

– Можливість попросити Google виконати глибше сканування завантажених файлів на наявність зловмисного програмного забезпечення та вірусів;

– Захист від раніше невідомих атак під час переходу на сайти;

– Індивідуальний захист на основі вашого рівня ризику.

В інших продуктах Google користувачі Enhanced Safe Browsing отримають додаткові засоби захисту:

– Посилений захист у GMail, включаючи додаткові перевірки вкладень і веб-посилань;

– Індивідуальний захист у разі виявлення атаки на обліковий запис.

Вибираючи Enhanced Safe Browsing, користувачі будуть ділитися додатковою інформацією, пов’язаною з безпекою, щоб покращити свій онлайн-захист. Ці дані використовуються лише з метою безпеки та видаляються через короткий проміжок часу. Надсилаючи додаткову інформацію про потенційно ризиковані події Chrome дозволяє безпечному перегляду покращити свою здатність виявляти зловмисний вміст в Інтернеті, щоб краще захистити користувачів у всій мережі.

На рисунку 1.7 Представлено блок-схему алгоритму обробки запитів Google Safe Browsing

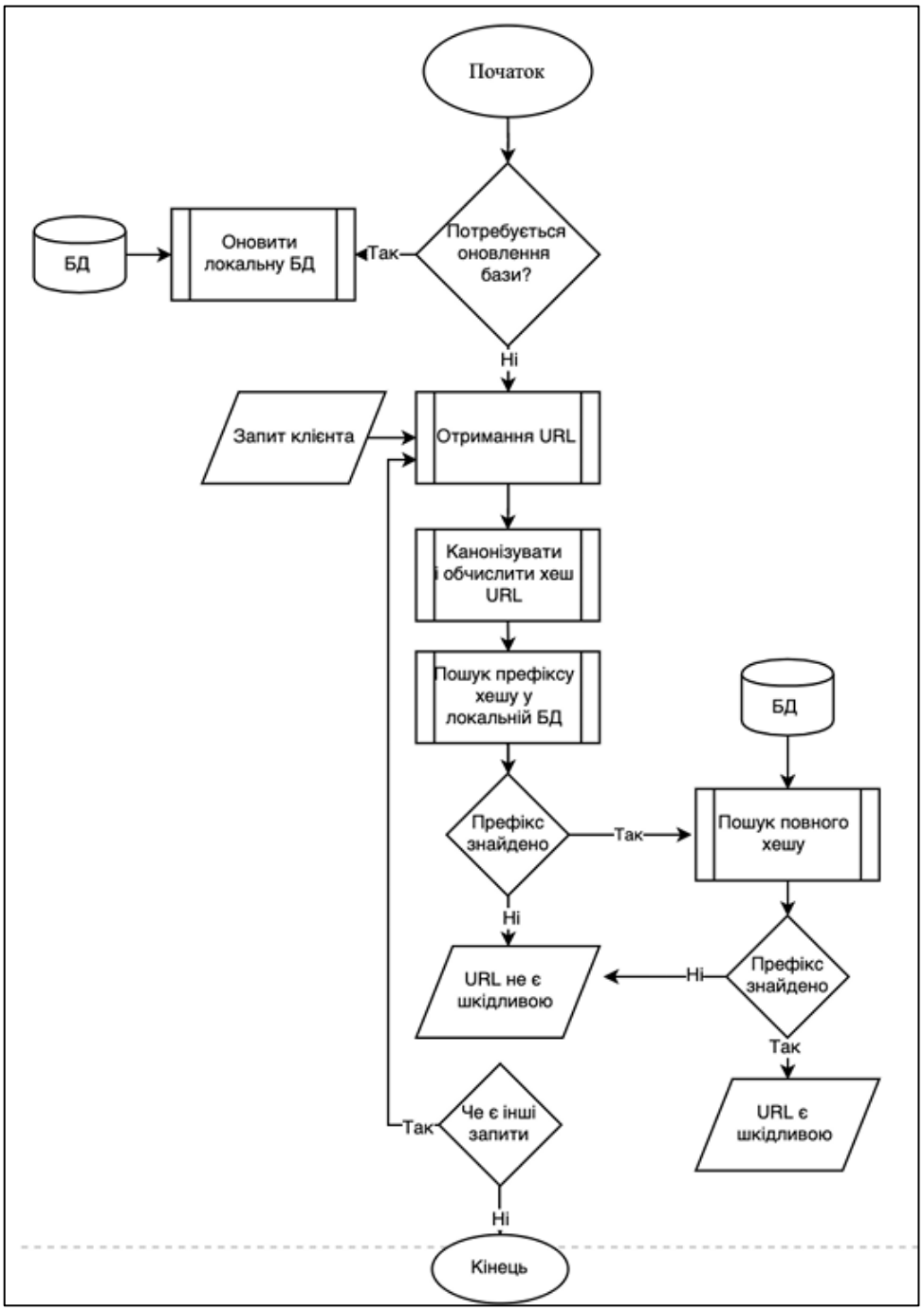

Рисунок 1.7 – Блок-схема алгоритму

1.3 Вибір мови програмування та середовища розробки

Вибір мови програмування є важливим кроком у розробці програмного забезпечення. Існує безліч мов, кожна з яких має свої особливості, переваги та сфери застосування. Однією з найбільш популярних мов програмування є Python, яка зарекомендувала себе як універсальний інструмент для вирішення різноманітних задач [9].

Python — це мова програмування високого рівня, яка дозволяє виконувати програми без попередньої компіляції завдяки інтерпретації. Вона використовує динамічну строгу типізацію даних і має легкий синтаксис, що робить її привабливою для розробників. Величезна стандартна бібліотека Python дозволяє вирішувати широкий клас задач, від веб-розробки до аналізу даних. Окрім стандартної бібліотеки, Python має репозиторій зовнішніх бібліотек, які розробляються та підтримуються спільнотою розробників.

Python активно використовується для розробки веб-додатків, системних утиліт, а також для аналізу даних. Це дозволяє значно прискорювати обчислення великих масивів даних. Простість синтаксису та велика кількість бібліотек роблять Python популярним серед науковців, що сприяло створенню потужних фреймворків для роботи з масивами даних та створення нейронних мереж [10].

Особливості Python:

– Простота використання. Python відомий своїм зрозумілим і легким для читання синтаксисом, що дозволяє розробникам швидко освоїти мову;

– Ефективні структури даних. Python має ефективні вбудовані структури даних, такі як списки, множини, словники, які полегшують розробку програм;

– Об'єктно-орієнтоване програмування. Python підтримує об'єктно-орієнтоване програмування, що сприяє модульності та повторному використанню коду;

– Інтерпретована мова. Python є інтерпретованою мовою, що дозволяє заощадити час на компіляцію і швидко тестувати код;

– Багатоплатформеність. Python працює на всіх основних платформах, таких як Windows, UNIX, Macintosh, OS/2.

Плюси:

– Її легко читати, вчити та писати. Це мова програмування високого рівня з англійським синтаксисом. Це полегшує читання та розуміння коду. Її дійсно легко зрозуміти і вивчити, тому багато людей рекомендують Python новачкам. Вам потрібно менше рядків коду для виконання того ж завдання в порівнянні з іншими основними мовами, такими як C/C++ та Java

– Підвищує продуктивність. Це дуже продуктивна мова. Завдяки її простоті розробники можуть зосередитися на розв’язанні проблеми. Їм не потрібно витрачати багато часу на розуміння синтаксису або поведінку мови програмування. Ви пишете менше коду та виконуєте більше завдань

– Інтерпретована мова. Python мова, що інтерпретується, а це означає, що вона безпосередньо виконує код по рядку. Якщо сталася помилка, вона зупиняє подальше виконання та повідомляє про її виникнення. Вона показує лише одну помилку, навіть якщо у програмі їх кілька. Це спрощує налагодження

– Динамічно типізована. Python не визначає тип змінної, доки ми не запустимо код. Вона автоматично надає тип даних, коли відбувається процес виконання. Фахівець може не турбуватися про оголошення змінних та типи даних

– Безкоштовна та з відкритим вихідним кодом. Ця мова постачається під схваленою OSI ліцензією з відкритим вихідним кодом. Це робить його безкоштовним для використання та розповсюдження. Ви можете завантажити вихідний код, змінити його та навіть розповсюджувати свою версію. Це корисно для організацій, які хочуть використати свою версію для розробки

– Підтримка великих бібліотек. Стандартна бібліотека Python є величезною, ви можете знайти майже всі функції, необхідні для вашого завдання. Таким чином ви не залежите від зовнішніх бібліотек

– Портативність. У багатьох мовах, таких як C/C++, потрібно змінити свій код, щоб запустити програму на різних платформах. З Python все інакше. Ви тільки пишете один раз і запускаєте її будь-де

Мінуси:

– Низька швидкість. Вище ми обговорювали, що це інтерпретована мова з динамічною типізацією. Порядкове виконання коду часто призводить до повільного виконання. Динамічна природа Python також є причиною її низької швидкості, оскільки їй доводиться виконувати додаткову роботу при виконанні коду;

– Неефективна для пам’яті. Ця мова програмування використовує великий обсяг пам’яті, це може бути недоліком при створенні програм, коли віддають перевагу оптимізації пам’яті;

– Слабка у мобільних обчисленнях. Python зазвичай використовується у серверному програмуванні. Ми не бачимо її на стороні клієнта або в мобільних програмах з таких причин: вона не заощаджує пам’ять і має повільну обчислювальну потужність у порівнянні з іншими мовами;

– Доступ до бази даних. Програмувати на цій мові легко, але коли ми взаємодіємо з базою даних, її не вистачає. Рівень доступу до бази даних у Python примітивний та недостатньо розвинений у порівнянні з іншими популярними технологіями;

– Помилки виконання. Це мова з динамічною типізацією, тому тип даних змінної може змінюватись у будь-який час. Змінна, що містить ціле число, у майбутньому може містити рядок, що може призвести до помилок виконання.

Середовищем розробки було вибрано платформу Kaggle. Kaggle — це онлайн-платформа, яка надає можливості для змагань з аналізу даних, хостингу наборів даних, створення спільнот та навчання машинному навчанню. Вибір Kaggle для реалізації проекту має численні переваги, особливо якщо метою є вирішення складних задач аналізу даних, машинного навчання або штучного інтелекту [11].

Переваги:

– Доступ до якісних даних. Kaggle надає великий вибір якісних і різноманітних наборів даних, що є ключовим фактором для успішної реалізації проектів у сфері аналізу даних і машинного навчання;

– Високий рівень конкуренції. Участь у змаганнях Kaggle дозволяє перевірити свої навички та моделі в умовах високої конкуренції. Це сприяє вдосконаленню своїх методів і підходів;

– Можливість навчання. Kaggle — це чудова платформа для навчання та самовдосконалення. Ви можете навчитися новим методам, переглядаючи рішення інших учасників, та отримати поради від експертів;

– Kaggle підтримує функції для спільної роботи над проектами. Ви можете співпрацювати з іншими учасниками, ділитися своїм кодом та отримувати відгуки;

– Репутація та кар'єрні можливості. Участь і перемоги в конкурсах Kaggle можуть значно підвищити вашу репутацію в спільноті дата-сайентістів. Це може сприяти кар'єрному зростанню та залученню до цікавих проектів у майбутньому.

Недоліки:

– Високий рівень конкуренції. Участь у змаганнях Kaggle вимагає високого рівня знань та навичок, що може бути складним для новачків;

– Обмеження ресурсів. Хоча Kaggle надає велику кількість ресурсів, іноді виникають обмеження щодо обчислювальних потужностей, що може вплинути на швидкість виконання великих моделей;

– Часові витрати. Участь у змаганнях та робота над проектами на Kaggle може вимагати значних часових витрат, особливо якщо ви прагнете досягти високих результатів.

1.4 Опис моделей, які будуть використовуватись для передбачення даних

Машинне навчання застосовує різні математичні алгоритми для класифікації об'єктів і прийняття рішень. В залежності від конкретної задачі та вхідних даних, різні алгоритми можуть демонструвати різну точність рішень і вимагати різної кількості обчислювальних ресурсів для виконання. Нижче наведено опис деяких з найпоширеніших алгоритмів прийняття рішень, які використовуються в сучасному аналізі даних [12].

Логістична регресія – це статистичний метод аналізу взаємозв'язку між незалежними змінними (регресія) та залежними змінними шляхом моделювання ймовірності події. Цей метод використовується для бінарної класифікації, коли залежна змінна має 2 можливі значення. При множинній регресії передбачається, що залежна змінна є лінійною функцією незалежної змінної:

$$y = b_1 x_1 + b_2 x_2 + \ldots + b_n x_n + uy = b_1 x_1 + b_2 x_2 + \ldots + b_n x_n + u\,,$$

де y – залежна змінна,

xi – пояснююча змінна,

bi – коефіцієнт змінної,

u – випадкова похибка.

У векторному вигляді це виражається як:

$$y = b'x + uy = b'x + u\,,$$

де x – вектор пояснюючих змінних,

b′ – транспонований вектор параметрів.

Умовна ймовірність події розраховується за формулою:

$$P(y \mid x) = b'xP(y \mid x) = b'x\,.$$

Логістична регресія має недоліки, такі як обмежена кількість вхідних факторів, і вимагає додаткового експертного підходу для вибору найбільш важливих характеристик. Аномальні дані можуть спотворити результати, тому часто потрібно усунути викиди, застосувати регуляризацію або виключити певні функції для підвищення точності моделі. Порівняно з методом найменших квадратів, логістична регресія споживає більше ресурсів і вимагає більше часу та обчислювальних ресурсів для побудови та навчання моделей. Більше того, оцінки ймовірності можуть бути недостатньо точними, особливо для незбалансованих даних або коли функція правдоподібності не є експоненціальною [13].

Незважаючи на ці обмеження, логістична регресія продовжує залишатися потужним інструментом для моделювання ймовірностей, що належать до певного класу, на основі змінних, як і не є основоположними. Це дозволяє робити прогнози, класифікувати клієнтів, оцінювати ризики, прогнозувати ймовірність захворювання та вирішувати інші завдання. Ретельно аналізуючи результати та вживаючи заходів для усунення аномалій, можна підвищити якість прогнозування та класифікації.

Однією з головних переваг логістичної регресії є можливість моделювати ймовірність настання події, що дозволяє зрозуміти вплив різних факторів. Це

важливо для аналізу взаємозалежності між змінними та обчислення ймовірностей для різних значень вхідних факторів. Наприклад, у фінансовому аналізі можна використовувати цей метод для прогнозування ймовірності успіху за замовчуванням клієнта або інвестиційного проекту. Крім того, логістична регресія дозволяє враховувати взаємодію між змінними та виявляти складні нелінійні залежності.

Успішне використання логістичної регресії включає високоякісну підготовку даних, облік деталей моделі та правильну інтерпретацію результатів. Забезпечити достовірність результатів логістичної регресії можна, лише беручи до уваги всі ці аспекти, які також звертають увагу на можливі обмеження цього методу, такі як обмежена кількість вхідних факторів і, в деяких випадках, можливість недостатньо точної оцінки ймовірностей.

Метод дерев рішень є одним з найбільш поширених способів вирішення задач класифікації та прогнозування. Цей метод видобутку даних також відомий як дерево вирішальних правил або дерево класифікації та регресії. Він використовується для вирішення проблем класифікації, коли цільова змінна має дискретні значення, і для прогнозування проблем, коли цільова змінна є безперервною. У цих випадках дерево рішень встановлює залежність цієї змінної від незалежної змінної. У своїй найпростішій формі дерево рішень-це спосіб подання правил у вигляді ієрархічної послідовної структури. Основою цієї структури є відповідь "так" або "ні" на кілька питань, які допоможуть визначити, як приймати рішення [14].

У процесі побудови моделі алгоритм ітеративно визначає ступінь впливу кожного вхідного атрибуту на значення вихідного атрибуту. Він використовує атрибут з найбільшим впливом для розбиття вузла дерева рішень. Вузол верхнього рівня показує розподіл значень вихідного атрибуту для всього набору даних. Кожен наступний вузол відображає розподіл вихідного атрибуту, враховуючи умови на вхідні атрибути, які відповідають цьому вузлу. Модель росте доти, доки розбиття вузла на подальші вузли підвищує ймовірність того, що вихідний атрибут прийме певне значення, порівняно з іншими значеннями, тобто розбиття покращує якість прогнозу. Алгоритм шукає атрибути та їх

значення, розбиття за якими дозволяє з більшою ймовірністю правильно передбачити значення вихідного атрибуту.

Переваги методу дерев рішень:

– Простота інтерпретації. Дерева рішень легко зрозуміти і візуалізувати, оскільки вони представляють правила у вигляді ієрархічної, послідовної структури. Це робить результати інтуїтивно зрозумілими навіть для тих, хто не має глибоких знань у галузі машинного навчання;

– Універсальність. Дерева рішень можуть використовуватись як для задач класифікації (коли цільова змінна має дискретні значення), так і для задач регресії (коли цільова змінна є безперервною). Це робить їх корисними для різних типів даних і задач;

– Обробка різноманітних типів даних. Вони можуть працювати з категоріальними та числовими змінними, використовуючи різні правила розбиття для кожного типу даних. Це дозволяє моделі ефективно працювати з широким спектром інформації;

– Мінімальні вимоги до підготовки даних. Дерева рішень потребують мінімальної попередньої обробки даних. Вони не вимагають нормалізації або масштабування даних, що зменшує час і зусилля, необхідні для підготовки даних;

– Стійкість до шуму та викидів. Дерева рішень відносно стійкі до шуму та викидів у даних, що дозволяє їм залишатись ефективними навіть при наявності аномалій у наборі даних;

– Виявлення важливих змінних. Дерева рішень здатні виявляти найбільш важливі змінні для задачі класифікації або прогнозування. Це допомагає краще зрозуміти взаємозв'язки в даних і може бути використано для вдосконалення моделей;

– Підтримка великих наборів даних. Вони можуть ефективно обробляти великі обсяги даних і працювати з великими наборами змінних, що робить їх популярними для задач Data Mining;

– Швидке навчання і прогнозування. Алгоритми дерев рішень зазвичай швидко навчаються і можуть швидко робити прогнози, що є важливим для додатків, де час має критичне значення;

– Можливість інтеграції з іншими методами. Дерева рішень можуть бути основою для ансамблевих методів, таких як випадковий ліс і бустинг, що дозволяє підвищити точність і надійність моделей.

На рисунку 1.8 представлено приклад блок-схеми методу дерев рішень.

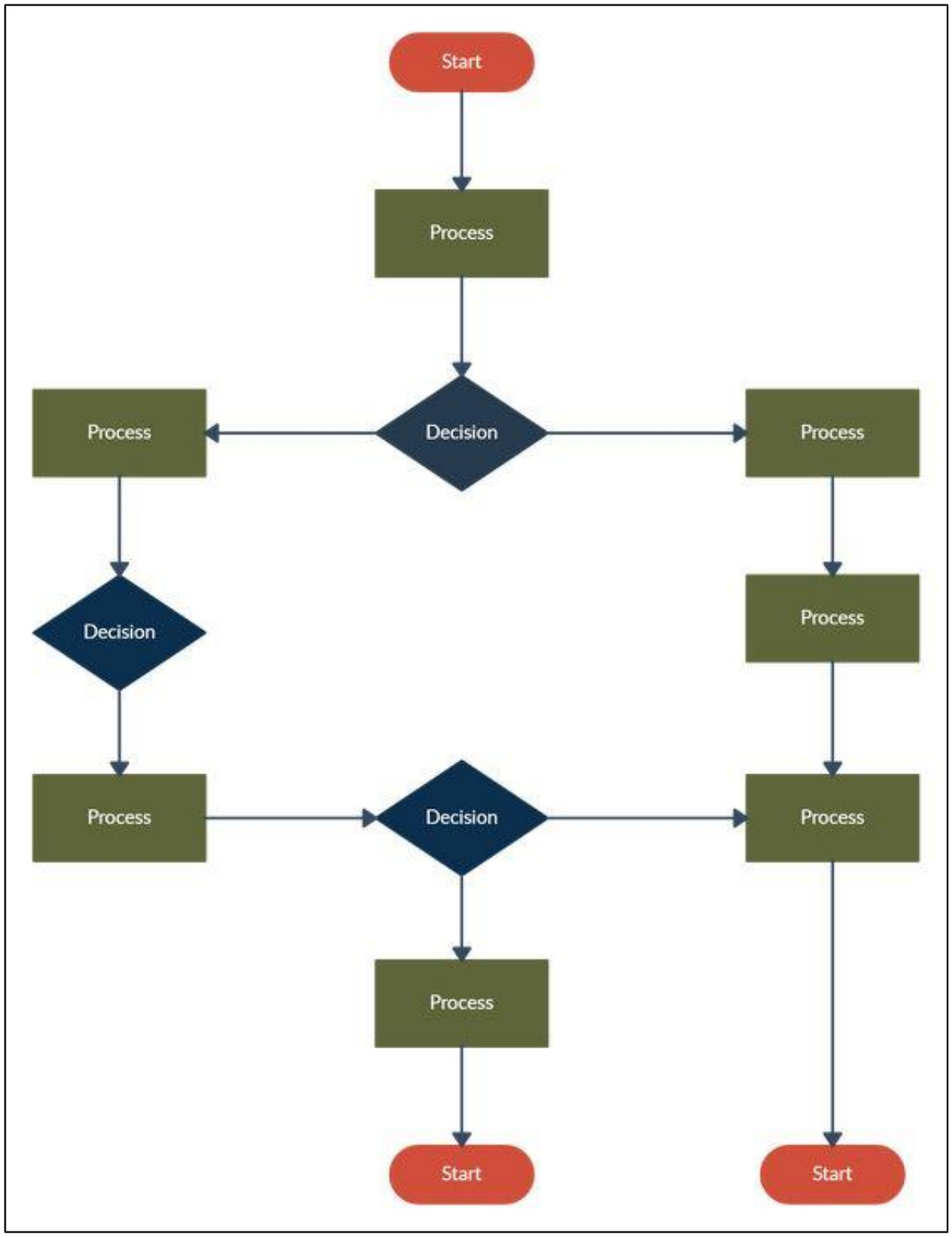

Рисунок 1.8 – Блок-схема методу дерев рішень

Випадковий ліс, є ансамблевим методом машинного навчання, що використовується для класифікації, регресії та інших завдань. Цей метод працює, будуючи багато дерев рішень під час навчання моделі та утворюючи моду для класів (у випадку класифікації) або усереднений прогноз (для регресії) з побудованих дерев. Недоліком цього методу є тенденція до перенавчання [15].

Лео Брейман та Адель Катлер запропонували розширення цього алгоритму, а «Випадковий ліс» став їхньою торговою маркою. Алгоритм поєднує дві основні ідеї: метод беггінга Бреймана і метод випадкових підпросторів, який запропонував Tin Kam Ho [16].

Алгоритм навчання класифікатора такий:

Нехай навчальна вибірка містить N прикладів, простір ознак має розмірність M, і заданий параметр m (зазвичай $m \approx \sqrt{M}$).

Усі дерева в комітеті будуються незалежно одне від одного за такою процедурою:

– Генерується випадкова підвибірка з повторенням розміром n з навчальної вибірки;

– Випадковим чином обираються m ознак з M;

– Побудова дерева рішень, яке класифікує приклади цієї підвибірки, де під час кожного розгалуження вибирається ознака для поділу не з усіх M, а лише з m випадково вибраних;

– Дерево будується до повного вичерпання підвибірки і не обрізається;

– Повертаємося до кроку 1, генеруємо нову вибірку і повторюємо побудову дерев.

Оптимальна кількість дерев підбирається так, щоб мінімізувати помилку класифікатора на тестовій вибірці або оцінку помилки out-of-bag. Класифікація об'єктів проводиться шляхом голосування: кожне дерево комітету класифікує об'єкт, і перемагає клас, за який проголосувало найбільше число дерев.

Оцінка важливості змінних полягає в тому, щоб визначити, які змінні найбільш впливають на результат моделі. Це можна зробити шляхом вимірювання зміни у відповіді моделі при зміні конкретної змінної.

Переваги методу випадковий ліс:

– Висока точність. Завдяки ансамблевому підходу, який об'єднує прогнози багатьох дерев рішень, метод зазвичай демонструє високу точність у задачах класифікації та регресії;

– Стійкість до перенавчання. Використання багатьох дерев і техніки випадкової вибірки зменшує ризик перенавчання, що робить модель більш узагальненою і стійкою до нових даних;

– Обробка великої кількості ознак. Метод ефективно працює з великими наборами даних і може обробляти безліч ознак, включаючи як числові, так і категоріальні;

– Мінімальна потреба в налаштуванні. Метод потребує менше налаштувань гіперпараметрів порівняно з іншими методами машинного навчання, що спрощує його застосування;

– Гнучкість. Метод може використовуватись як для класифікаційних, так і для регресійних задач, що робить його універсальним інструментом для різних типів проблем;

– Природна можливість для паралелізації. Побудова окремих дерев є незалежною, що дозволяє легко розпаралелити процес навчання і скоротити час обробки великих даних.

На рисунку 1.9 Представлено приклад блок-схеми методу випадковий ліс.

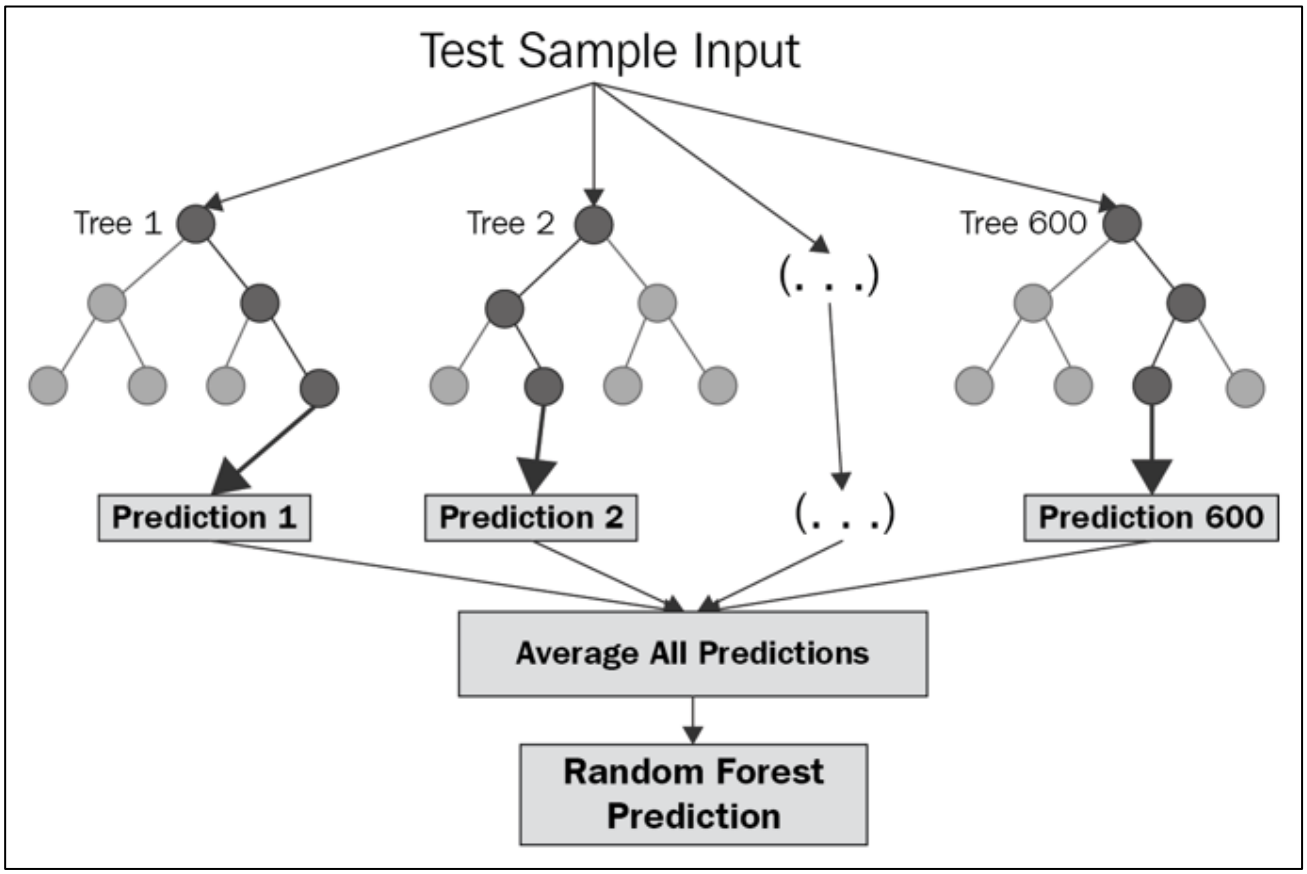

Рисунок 1.9 – Блок-схема методу випадковий ліс

Ансамблеве навчання – це метод, що об’єднує прогнози декількох окремих моделей для підвищення точності та надійності. Завдяки поєднанню кількох моделей, цей підхід стає більш гнучким і менш чутливим до коливань у даних. Найпоширеніші методи ансамблевого навчання – це багінг та бустинг [17].

Багінг передбачає паралельне навчання багатьох моделей на різних підмножинах даних, обраних випадковим чином. Одним із прикладів застосування багінгу є випадкові ліси. У випадкових лісах створюється серія дерев рішень, кожне з яких навчається на випадковій вибірці даних, а остаточний прогноз визначається шляхом усереднення результатів усіх дерев.

Бустинг – це метод послідовного навчання моделей, де кожна наступна модель намагається виправити помилки попередньої. Дерева рішень з градієнтним бустингом є яскравим прикладом цього підходу. В основі бустингу лежить ідея поступового накопичення слабких моделей, кожна з яких перевершує випадкові вгадування, і поступового вдосконалення їх на основі зауважень попереднього етапу.

Градієнтний бустинг – це метод машинного навчання для задач класифікації та регресії, який об’єднує кілька слабких моделей для створення однієї сильної. У цьому підході кожне дерево рішень намагається зменшити помилку попереднього. Хоча такі моделі можуть бути важкими для навчання, їх поступовий ітеративний підхід дозволяє досягти високої точності [18].

Слабкі моделі інтегруються таким чином, що кожна наступна модель враховує залишкові помилки попереднього етапу. Фінальна модель комбінує результати всіх етапів, що забезпечує високу точність. Втрати визначаються за допомогою функції втрат, яка використовується для оцінки помилок.

Оптимізація градієнтного бустингу

Методи градієнтного бустингу можуть бути схильні до перенавчання, що погіршує їх продуктивність на тестових наборах даних.

Метод Градієнтний бустинг має кілька ключових переваг, які роблять його потужним інструментом для задач класифікації та регресії:

– Висока точність. Метод може досягати високої точності прогнозування завдяки поєднанню багатьох слабких учнів (наприклад, дерев рішень) в один сильний ансамбль. Це дозволяє моделі виявляти складні патерни в даних;

– Обробка різноманітних типів даних. Метод добре працює як з числовими, так і з категоріальними даними, що робить його універсальним інструментом для різних типів задач;

– Гнучкість. Метод дозволяє налаштовувати багато параметрів (наприклад, кількість дерев, глибину дерев, швидкість навчання), що дає можливість оптимізувати модель під конкретну задачу та дані;

– Обробка пропущених значень. Метод може автоматично обробляти пропущені значення в наборах даних, що спрощує процес підготовки даних;

– Масштабованість. Метод може бути застосований до великих наборів даних і добре масштабується на великі обсяги інформації, що робить його придатним для промислових застосувань;

– Стійкість до перенавчання. Завдяки методам регуляризації, таким як усереднення та випадкова вибірка, може зменшувати ризик перенавчання, що забезпечує кращу узагальнюваність на нових даних;

– Інтерпретованість. Метод складається з дерев рішень, які відносно легко інтерпретувати. Це дозволяє розуміти вплив окремих змінних на результати моделі;

– Автоматичне врахування взаємодій між ознаками. Здатний автоматично виявляти та використовувати взаємодії між ознаками, що дозволяє будувати більш точні прогнози;

– Стійкість до надмірної кількості параметрів. Метод добре працює навіть при великій кількості вхідних параметрів, що знижує потребу в сильній попередній обробці даних;

– Можливість паралелізації. Процес навчання методу може бути паралелізований, що дозволяє використовувати обчислювальні ресурси ефективніше та скорочувати час навчання моделі.

Ці переваги роблять Градієнтний бустинг потужним і ефективним методом для широкого спектру задач у машинному навчанні та аналізі даних.

На рисунку 1.10 Представлено приклад блок-схеми методу градієнтного бустингу

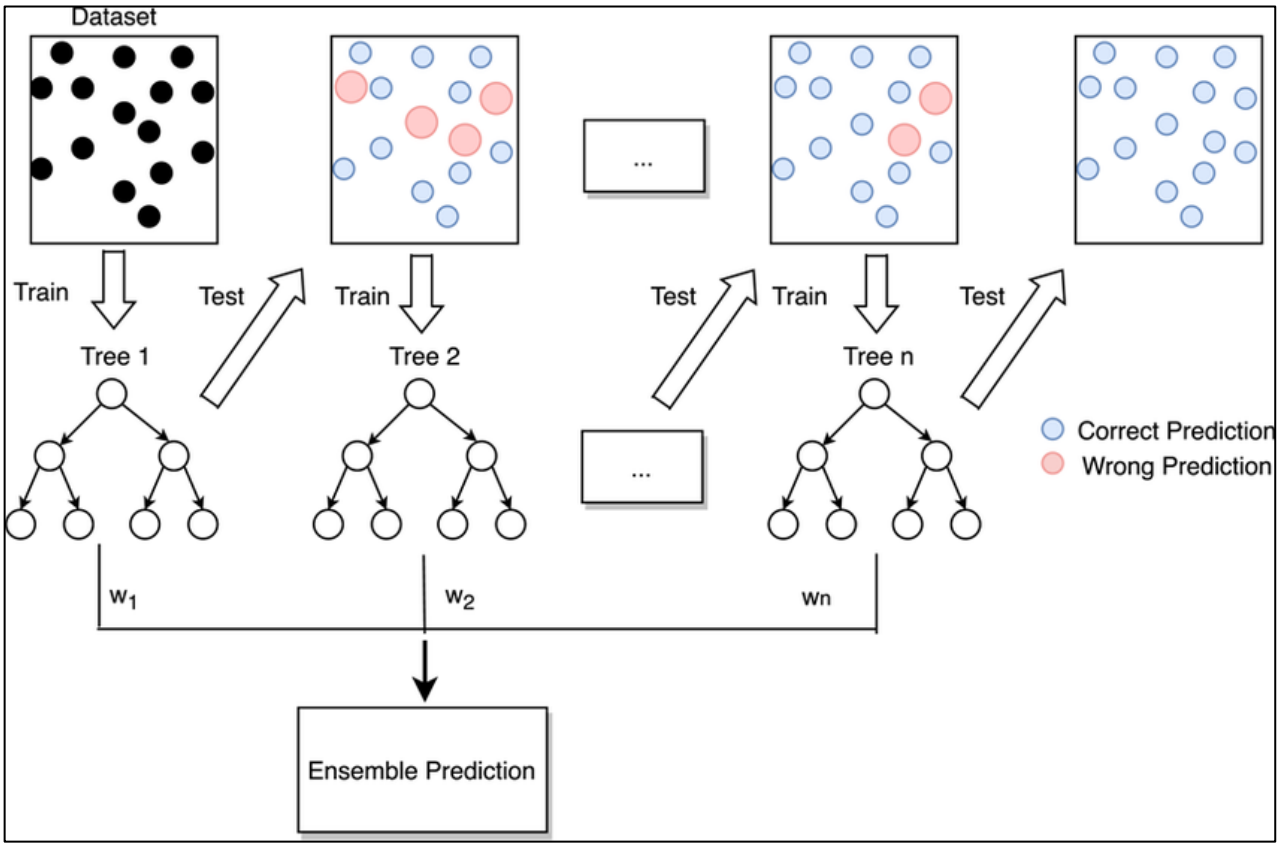

Рисунок 1.10 – Блок-схема методу градієнтний бустинг

1.5 Висновки

В даному розділі було проаналізовано та описано об’єкт дослідження предметної області, а саме проблема фішингового шахрайства в інтернеті. Описано основні види фішингу: поштовий, онлайн і комбінований. Також описано та вибрано мову програмування – Python та середовище розробки – Kaggle. Також описано недоліки та переваги методів інформаційної технології розв’язання задачі передбачення фішингових сайтів. Для розв’язання поставлених задач обрано такі методи як логістична регресія, дерево рішень, випадковий ліс та градієнтний бустинг.

2 ВИБІР ТА АНАЛІЗ ОПТИМАЛЬНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

2.1 Опис датасету для розробки технології передбачення фішингових сайтів

В даній роботі використано датасет даних про фішинг веб-сторінки.

Цей набір даних є результатом об’єднання двох наборів даних з ідентичними характеристиками. Однак не всі ознаки з оригінальних наборів даних були збережені в об’єднаному наборі даних. Це вибіркове включення ознак було зроблено, щоб зосередитися на найбільш відповідних даних і уникнути надмірності. Отриманий набір даних надає всебічне уявлення про спільні характеристики двох вихідних наборів даних, зберігаючи впорядкований і цілеспрямований набір ознак [19].

Цей набір даних містить такі ознаки:

– url_length: довжина URL-адреси. Ця ознака визначає загальну кількість символів в URL-адресі, включаючи всі символи, такі як літери, цифри, спеціальні символи тощо. Наприклад, для URL-адреси http://example.com/path?query=value, довжина буде 33 символи;

– n_dots: кількість символів «.» в URL-адресі. Ця ознака рахує кількість крапок (.) в URL-адресі. Наприклад, в адресі http://example.com/path.to/resource, кількість крапок буде 3;

– n_hypens: кількість символів «-» в URL-адресі. Ця ознака рахує кількість дефісів (-) в URL-адресі. Наприклад, в адресі http://example-site.com/path-to-resource, кількість дефісів буде 2;

– n_underline: кількість символів «_» в URL-адресі. Ця ознака рахує кількість символів підкреслення (_) в URL-адресі. Наприклад, в адресі http://example.com/path_to_resource, кількість підкреслень буде 1;

– n_slash: кількість символів '/' в URL-адресі. Ця ознака рахує кількість косих рисок (/) в URL-адресі. Наприклад, в адресі http://example.com/path/to/resource, кількість косих рисок буде 4;

– n_questionmark: кількість «?» символів в URL-адресі. Ця ознака рахує кількість знаків питання (?) в URL-адресі. Наприклад, в адресі http://example.com/path?query=value, кількість знаків питання буде 1;

– n_equal: кількість символів «=» в URL-адресі. Ця ознака рахує кількість знаків рівності (=) в URL-адресі. Наприклад, в адресі http://example.com/path?query=value, кількість знаків рівності буде 1;

– n_at: кількість символів «@» в URL-адресі. Ця ознака рахує кількість символів собачки (@) в URL-адресі. Наприклад, в адресі http://example.com/user@domain, кількість символів собачки буде 1;

– n_and: кількість символів «&» в URL-адресі. Ця ознака рахує кількість амперсандів (&) в URL-адресі. Наприклад, в адресі http://example.com/path?query1=value1&query2=value2, кількість амперсандів буде 1;

– n_exclamation: кількість "!" символів в URL-адресі. Ця ознака рахує кількість знаків оклику (!) в URL-адресі. Наприклад, в адресі http://example.com/path!resource, кількість знаків оклику буде 1;

– n_space: кількість символів " " в URL-адресі. Ця ознака рахує кількість пробілів в URL-адресі. Пробіли в URL можуть бути закодовані як %20. Наприклад, в адресі http://example.com/path%20to%20resource, кількість пробілів буде 2;

– n_tilde: кількість символів «~» в URL-адресі. Ця ознака рахує кількість тильд (~) в URL-адресі. Наприклад, в адресі http://example.com/~user, кількість тильд буде 1;

– n_comma: кількість символів «,» в URL-адресі. Ця ознака рахує кількість ком (,) в URL-адресі. Наприклад, в адресі http://example.com/path,resource, кількість ком буде 1;

– n_plus: кількість символів «+» в URL-адресі. Ця ознака рахує кількість плюсів (+) в URL-адресі. Наприклад, в адресі http://example.com/path+resource, кількість плюсів буде 1;

– n_asterisk: кількість символів «*» в URL-адресі. Ця ознака рахує кількість зірочок (*) в URL-адресі. Наприклад, в адресі http://example.com/path*resource, кількість зірочок буде 1;

– n_hastag: кількість символів «#» в URL-адресі. Ця ознака рахує кількість решіток (#) в URL-адресі. Наприклад, в адресі http://example.com/path#section, кількість решіток буде 1;

– n_dollar: кількість символів «$» в URL-адресі. Ця ознака рахує кількість знаків долара ($) в URL-адресі. Наприклад, в адресі http://example.com/path$resource, кількість знаків долара буде 1;

– n_percent: кількість символів «%» в URL-адресі. Ця ознака рахує кількість відсотків (%) в URL-адресі. Наприклад, в адресі http://example.com/path%20resource, кількість відсотків буде 1;

– n_redirection: кількість переспрямувань в URL-адресі. Ця ознака визначає кількість переспрямувань, через які проходить URL-адреса, перш ніж досягти кінцевого пункту призначення. Це може включати кількість перенаправлень HTTP 3xx;

– фішинг: мітки URL-адреси. 1 означає фішинг, а 0 – законний. Ця ознака використовує алгоритм або модель для класифікації URL-адреси як фішингової (1) або законної (0). Фішингові URL-адреси створюються для обману користувачів і крадіжки конфіденційної інформації.

На рисунках 2.1 та 2.2 показано перші 10 рядків набору даних датасету.

	url_length	n_dots	n_hypens	n_underline	n_slash	n_questionmark	n_equal	n_at	n_and	n_exclamation
0	37	3	0	0	0	0	0	0	0	0
1	77	1	0	0	0	0	0	0	0	0
2	126	4	1	2	0	1	3	0	2	0
3	18	2	0	0	0	0	0	0	0	0
4	55	2	2	0	0	0	0	0	0	0
5	32	3	1	0	0	0	0	0	0	0
6	19	2	0	0	0	0	0	0	0	0
7	81	2	0	0	0	0	0	0	0	0
8	42	2	0	0	0	0	0	0	0	0
9	104	1	10	0	0	0	0	0	0	0

Рисунок 2.1 – Приклад набору даних

n_space	n_tilde	n_comma	n_plus	n_asterisk	n_hastag	n_dollar	n_percent	n_redirection	phishing
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Рисунок 2.2 – Приклад набору даних

На рисунку 2.3 показано огляд даних датасету

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100077 entries, 0 to 100076
Data columns (total 20 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   url_length      100077 non-null  int64
 1   n_dots          100077 non-null  int64
 2   n_hypens        100077 non-null  int64
 3   n_underline     100077 non-null  int64
 4   n_slash         100077 non-null  int64
 5   n_questionmark  100077 non-null  int64
 6   n_equal         100077 non-null  int64
 7   n_at            100077 non-null  int64
 8   n_and           100077 non-null  int64
 9   n_exclamation   100077 non-null  int64
 10  n_space         100077 non-null  int64
 11  n_tilde         100077 non-null  int64
 12  n_comma         100077 non-null  int64
 13  n_plus          100077 non-null  int64
 14  n_asterisk      100077 non-null  int64
 15  n_hastag        100077 non-null  int64
 16  n_dollar        100077 non-null  int64
 17  n_percent       100077 non-null  int64
 18  n_redirection   100077 non-null  int64
 19  phishing        100077 non-null  int64
dtypes: int64(20)
memory usage: 15.3 MB
```

Рисунок 2.3 – Огляд даних датасету

Інформація відображає кількість стовпів та рядків, назви стовпців та типи даних.

На рисунках 2.4 та 2.5 показано описову статистики для числових стовпців

	url_length	n_dots	n_hypens	n_underline	n_slash	n_questionmark	n_equal	n_at	n_and	n_exclamation
count	100077.0000	100077.0000	100077.0000	100077.0000	100077.0000	100077.0000	100077.0000	100077.0000	100077.0000	100077.0000
mean	39.1777	2.2244	0.4052	0.1377	1.1354	0.0244	0.2158	0.0221	0.1433	0.0026
std	47.9718	1.2550	1.2855	0.7240	1.8285	0.1678	0.9598	0.2684	0.9137	0.0822
min	4.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
25%	18.0000	2.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
50%	24.0000	2.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
75%	44.0000	2.0000	0.0000	0.0000	2.0000	0.0000	0.0000	0.0000	0.0000	0.0000
max	4165.0000	24.0000	43.0000	21.0000	44.0000	9.0000	23.0000	43.0000	26.0000	10.0000

Рисунок 2.4 – Описова статистика

n_space	n_tilde	n_comma	n_plus	n_asterisk	n_hastag	n_dollar	n_percent	n_redirection	phishing
100077.0000	100077.0000	100077.0000	100077.0000	100077.0000	100077.0000	100077.0000	100077.0000	100077.0000	100077.0000
0.0049	0.0036	0.0024	0.0025	0.0041	0.0004	0.0019	0.1093	0.3615	0.3633
0.1446	0.0785	0.0796	0.1044	0.2840	0.0580	0.0974	1.6953	0.7755	0.4810
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-1.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	1.0000
18.0000	5.0000	11.0000	19.0000	60.0000	13.0000	10.0000	174.0000	17.0000	1.0000

Рисунок 2.5 – Описова статистика

Надає короткий огляд основних статистичних показників для кожного числового стовпця. Ці показники включають:

– Count (Кількість). Кількість ненульових (непорожніх) значень у кожному стовпці;

– Mean (Середнє). Середнє значення для кожного стовпця;

– Std (Стандартне відхилення). Стандартне відхилення значень у кожному стовпці;

– Min (Мінімум). Мінімальне значення в кожному стовпці;

– 25% (Перший квартиль). Значення, нижче якого знаходиться 25% даних у кожному стовпці;

– 50% (Медіана або другий квартиль). Значення, нижче якого знаходиться 50% даних (медіана) у кожному стовпці;

– 75% (Третій квартиль). Значення, нижче якого знаходиться 75% даних у кожному стовпці;

– Max (Максимум). Максимальне значення в кожному стовпці.

2.2 Розвідувальний аналіз фішингових сайтів

Інформаційна технологія передбачення фішингових сайтів, яка може допомогти в аналізі та виявленні потенційно небезпечних веб-ресурсів, доступна в Kaggle [20].

Містить детальний аналіз даних, що стосуються фішингових сайтів, включаючи використання різноманітних методів машинного навчання для передбачення фішингових загроз. Розглядаються різні ознаки веб-сайтів, такі як довжина URL, кількість крапок у URL та інші важливі параметри, що можуть відрізняти фішингові сайти від нормальних.

Приклад порівняння розподілу двох параметрів між нормальними та фішинговими веб-сайтами за допомогою діаграм розмаху (boxplot). Графік створений для аналізу даних, де відображено різницю в розподілі логарифмічних значень двох змінних: довжини URL (url_length) та кількості крапок у URL (n_dots) для нормальних та фішингових веб-сайтів.

На рисунку 2.6 показано графік розподілу логарифмічної довжини URL для нормальних та фішингових веб-сайтів.

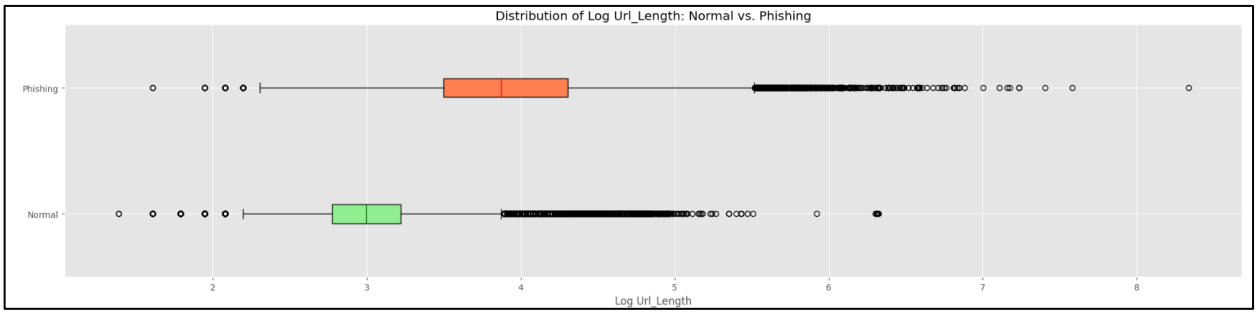

Рисунок 2.6 – Графік розподілу довжини URL для нормальних та фішингових веб-сайтів

Світло-зелені коробки представляють нормальні веб-сайти, тоді як коралові коробки представляють фішингові веб-сайти.

На рисунку 2.7 показано розподіл логарифмічної кількості крапок у URL для нормальних та фішингових веб-сайтів.

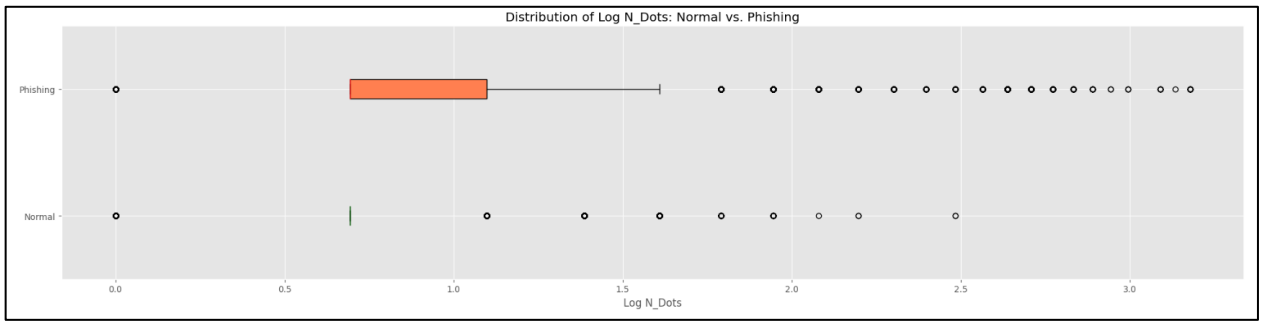

Рисунок 2.7 – Графік розподілу кількості крапок у URL для нормальних та фішингових веб-сайтів

Так само, світло-зелені коробки представляють нормальні веб-сайти, а коралові коробки - фішингові веб-сайти.

На рисунку 2.8 показано графік, який являє собою стовпчикову діаграму, яка порівнює середні значення різних ознак для фішингових і не фішингових веб-сайтів. Він створений на основі обчислених описових статистик, де було взято середнє значення кожної ознаки для обох категорій веб-сайтів.

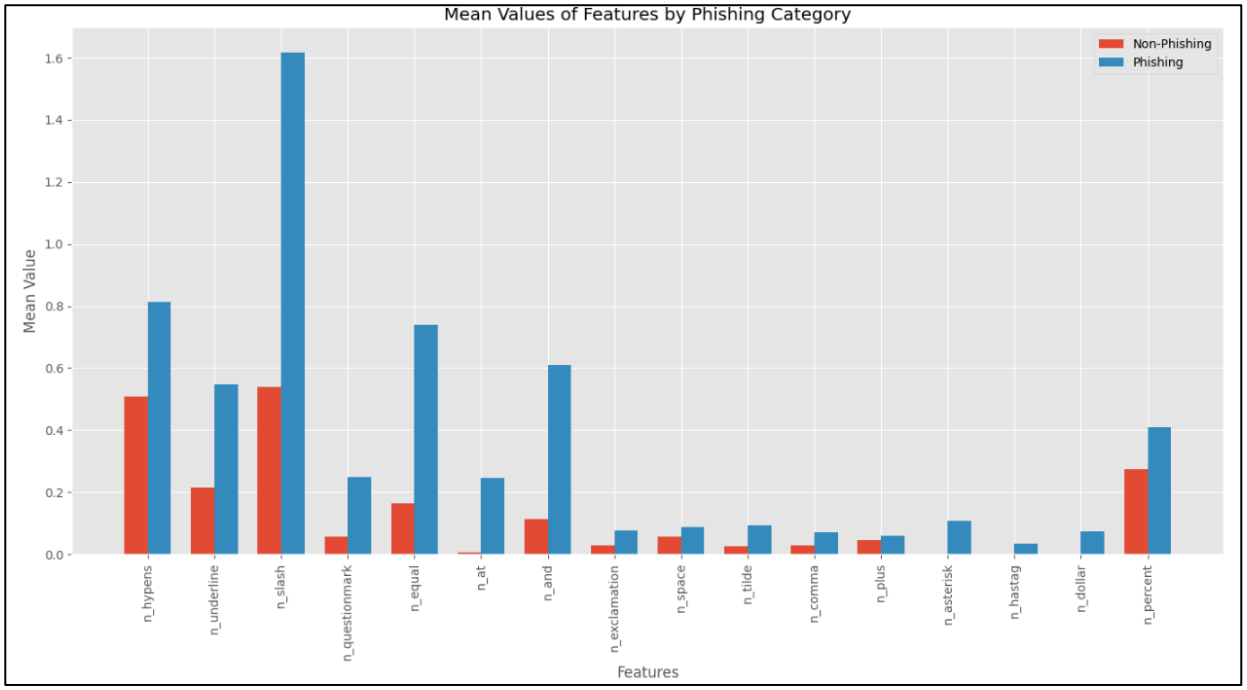

Рисунок 2.8 – Графік порівняння середніх значень ознак для фішингових та не фішингових сайтів

Можемо зробити такі висновки:

– Фішингові URL-адреси часто використовують певні символи для маскування під легальні сайти;

– Такі особливості, як частота використання дефісів, косих рисок, знаків питання, знаків рівності та амперсандів, є важливими підказками для виявлення фішингу;

– Розуміння цих закономірностей має вирішальне значення, оскільки алгоритми машинного навчання можуть використовувати ці знання для ефективного розрізнення шкідливих та безпечних URL-адрес;

На рисунках 2.9 та 2.10 показано теплову карту, яка використовується для відображення кореляційної матриці, де кожна клітинка представляє коефіцієнт кореляції між двома змінними.

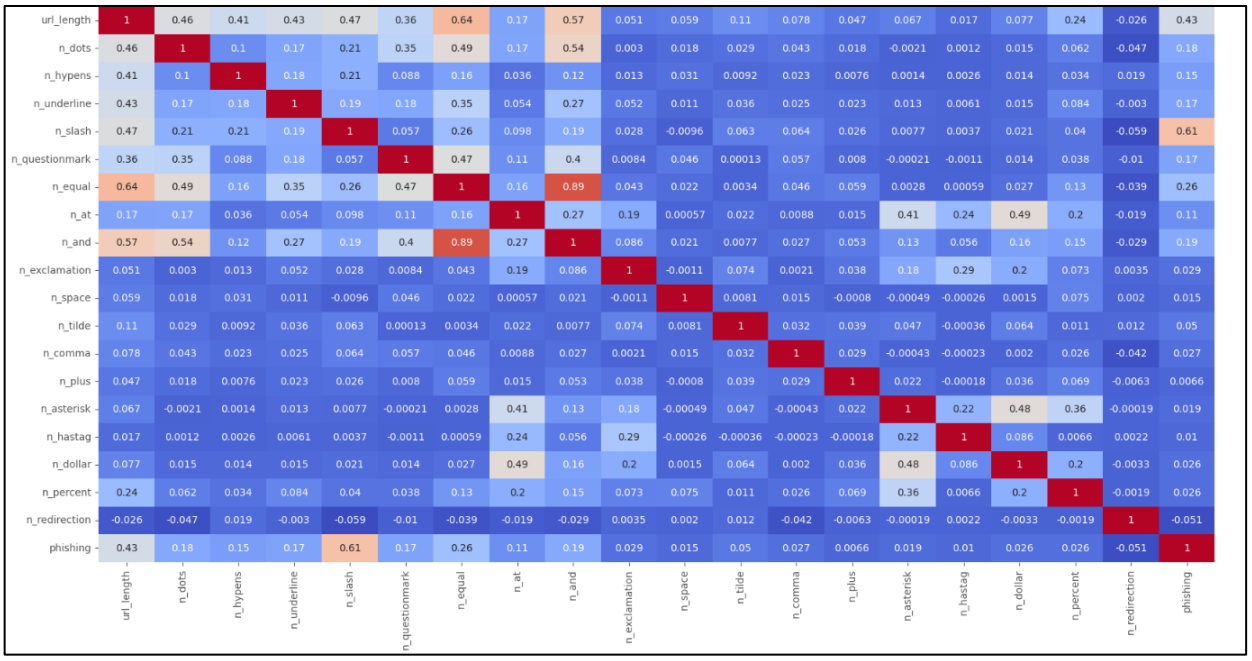

Рисунок 2.9 – Графік теплової карти

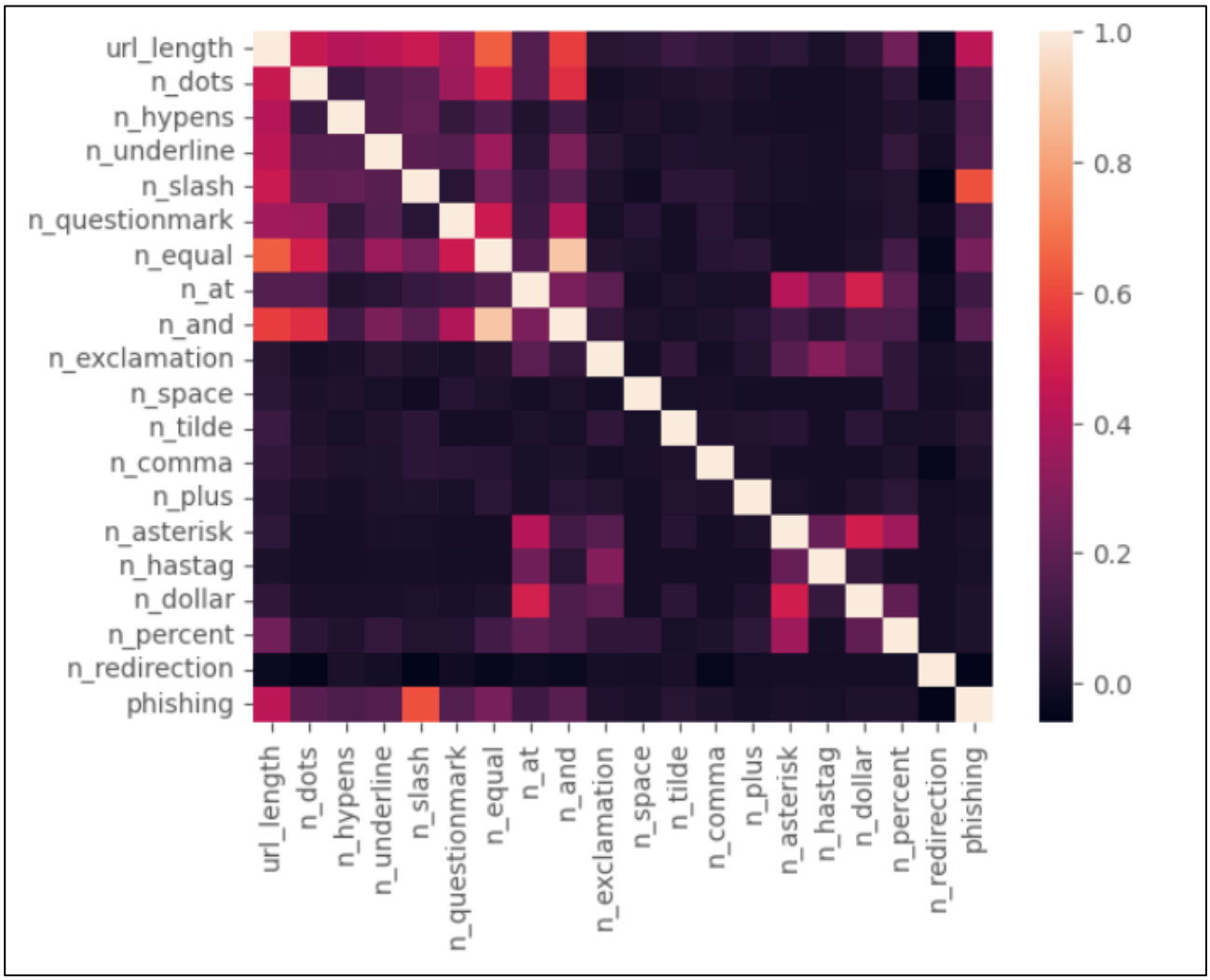

Рисунок 2.10 – Графік теплові карти

Теплова карта показує кореляцію даних. Можна побачити пари змінних, які мають сильну позитивну або негативну кореляцію. Це може вказувати на тісний зв'язок між цими змінними. Змінні з низькою або нульовою кореляцією показують, що між ними немає очевидного лінійного зв'язку.

2.3 Висновки

В даному розділі було вибрано та описано датасет «Web Page Phishing Dataset» [19], який має 20 ознак фішингових та нефішингових сайтів. Також проведено розвідувальний аналіз даних, який показав, що найбільш поширеними ознаками фішингових сайтів є косі риски, дефіси, знаки рівності та кількість символів амперсандів «&».

3 РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ДЛЯ ПЕРЕДБАЧЕННЯ ФІШИНГОВИХ САЙТІВ

3.1 Розроблення алгоритму інформаційної технології передбачення

На рисунку 3.1 показано блок-схему алгоритму інформаційної технології передбачення фішингових сайтів.

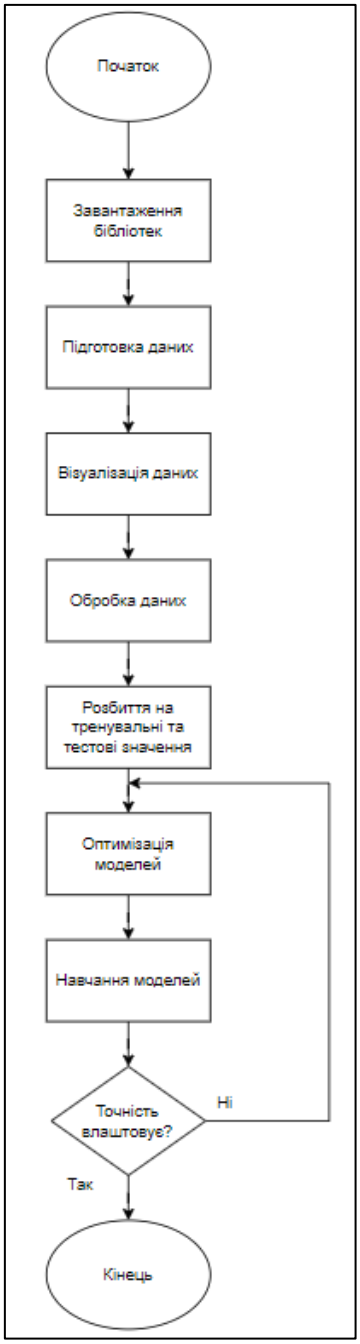

Рисунок 3.1 – Блок-схема алгоритму інформаційної технології

На рисунку 3.2 показано бібліотеки, які використовувались в розробленні інформаційної технології передбачення фішингових сайтів.

Рисунок 3.2 – Блок-схема використаних бібліотек

– NumPy. Використовується для числових обчислень і роботи з масивами;

– Pandas. Використовується для роботи з даними у вигляді таблиць;

– Seaborn. Бібліотека для візуалізації даних;

– XGBoost. Використовується для методу градієнтного бустингу;

– SciPy. Набір математичних алгоритмів і зручних функцій для статистики, оптимізації, інтеграції та інших наукових розрахунків;

– Statsmodels. Бібліотека для статистичного моделювання, тестування гіпотез та інших статистичних операцій;

– Matplotlib. Основна бібліотека для створення графіків та візуалізації даних;

– Sklearn: roc_curve, auc. Функції для побудови ROC кривої та обчислення площі під кривою;

– Sklearn: DecisionTreeClassifier. Класифікатор дерева рішень;

– Sklearn: StandardScaler. Функція для стандартизації особливостей даних;

– Sklearn: LogisticRegression. Модель логістичної регресії;

– Sklearn: train_test_split. Функція для поділу даних на тренувальний і тестовий набори;

– Sklearn: accuracy_score, confusion_matrix. Метрики для оцінки точності моделі та побудови матриці плутанини;

– Sklearn: RandomForestClassifier, GradientBoostingClassifier. Енсемблові методи для побудови класифікаційних моделей на основі випадкових лісів та градієнтного бустингу.

3.2 Моделювання та передбачення

На рисунку 3.3 показано лістинг коду для методу логістичної регресії.

```
X_train, X_test, y_train, y_test = train_test_split(data, y, test_size=0.3, random_state=42)
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)

predictions_train = model.predict(X_train)
predictions_test = model.predict(X_test)

print(f"Train Accuracy Score: {accuracy_score(y_train,predictions_train)}")
print(f"Test Accuracy Score: {accuracy_score(y_test,predictions_test)}")
```

```
Train Accuracy Score: 0.8582073572866259
Test Accuracy Score: 0.8566147082334132
```

Рисунок 3.3 – Лістинг коду для методу логістичної регресії

Результати Логістичної регресії: train accuracy score: 0.858, test accuracy score: 0.856.

На рисунку 3.4 показано графік, який відображає результати класифікації моделі для тестових даних, наведені у вигляді матриці плутанини. Вона дозволяє оцінити відповідність між фактичними та передбаченими значеннями класів.

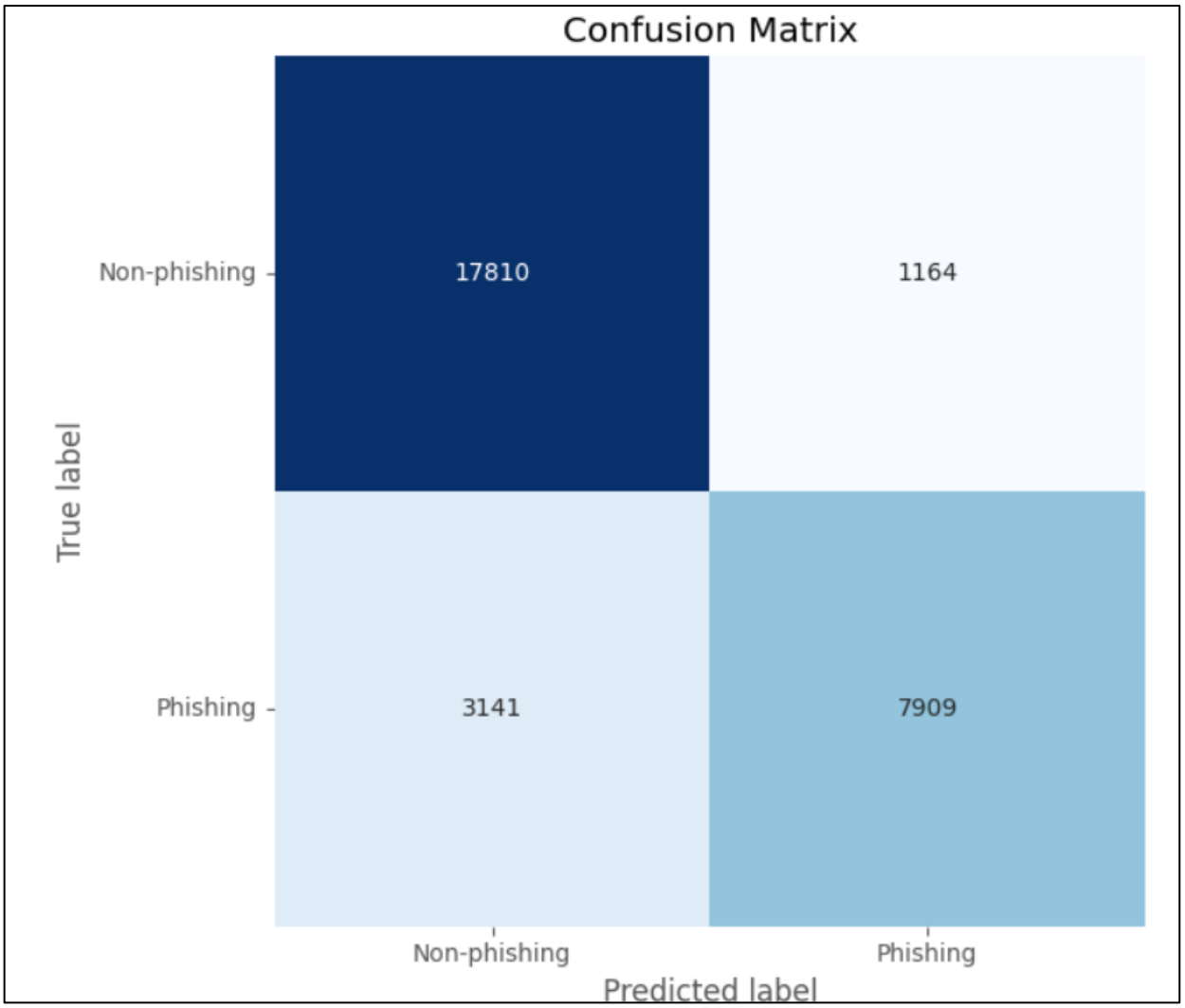

Рисунок 3.4 – Матриця плутанини методу логістичної регресії

На рисунку 3.5 показано графік, який використовується для оцінки продуктивності бінарного класифікатора, відображаючи зв'язок між чутливістю (True Positive Rate) та специфічністю (False Positive Rate) при різних порогах вирішення.

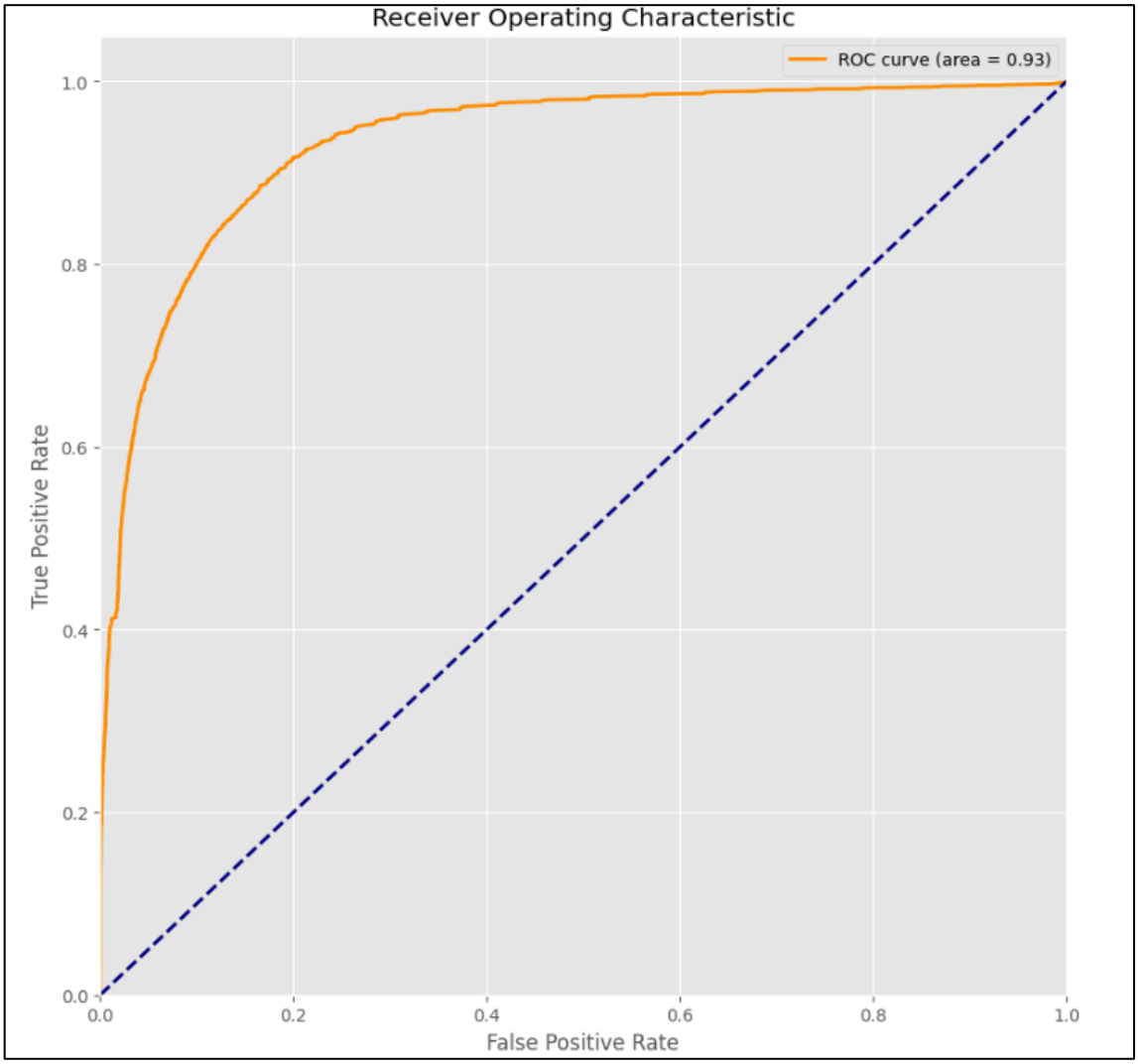

Рисунок 3.5 – ROC крива методу логістичної регресії

ROC крива показує площу 0.93, що вказує на високу здатність розрізняти фішингові та нефішингові сайти, Матриця плутанини показує, що модель правильно ідентифікує фішингові сайти – 7909, нефішингові сайти – 17810, хибнопозитивні – 1164, хибнонегативні – 3141.

На рисунку 3.6 показано лістинг коду для методу дерев рішень.

```
model = DecisionTreeClassifier(ccp_alpha= 0.0001, criterion= 'gini', max_depth= 32, min_samples_split= 5,random_state=42)
model.fit(X_train,y_train)

predictions_train = model.predict(X_train)
predictions_test = model.predict(X_test)

print(f"Train Accuracy Score: {accuracy_score(y_train,predictions_train)}")
print(f"Test Accuracy Score: {accuracy_score(y_test,predictions_test)}")
```

```
Train Accuracy Score: 0.8899975732659557
Test Accuracy Score: 0.8863908872901679
```

Рисунок 3.6 – Лістинг коду для методу дерев рішень

Результати методу дерев рішень: train accuracy score: 0.889, test accuracy score: 0.886.

На рисунку 3.7 показано графік, який відображає результати класифікації моделі для тестових даних, наведені у вигляді матриці помилок. Вона дозволяє оцінити відповідність між фактичними та передбаченими значеннями класів.

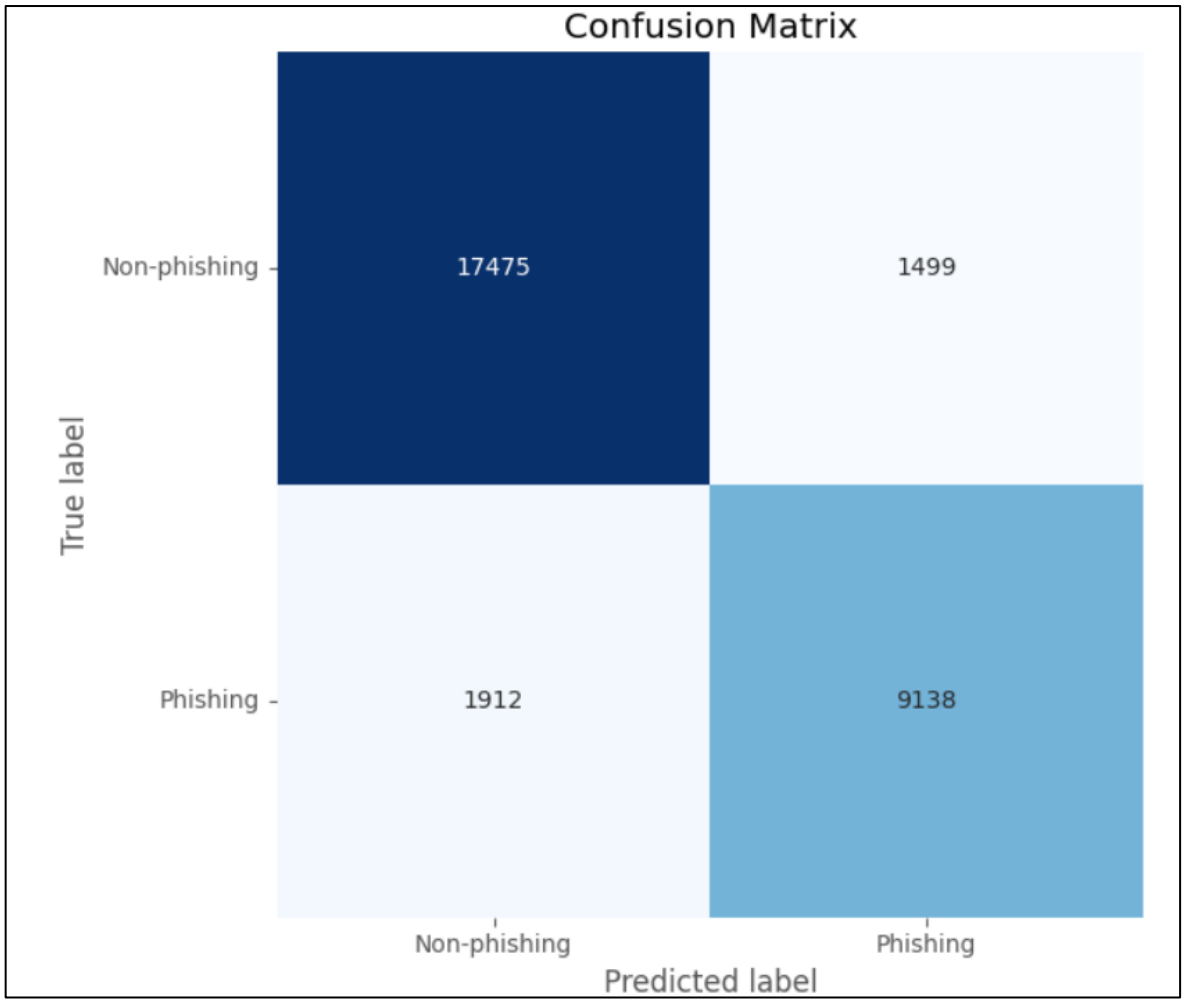

Рисунок 3.7 – Матриця плутанини методу дерев рішень

На рисунку 3.8 показано графік, який використовується для оцінки продуктивності бінарного класифікатора, відображаючи зв'язок між чутливістю (True Positive Rate) та специфічністю (False Positive Rate) при різних порогах вирішення.

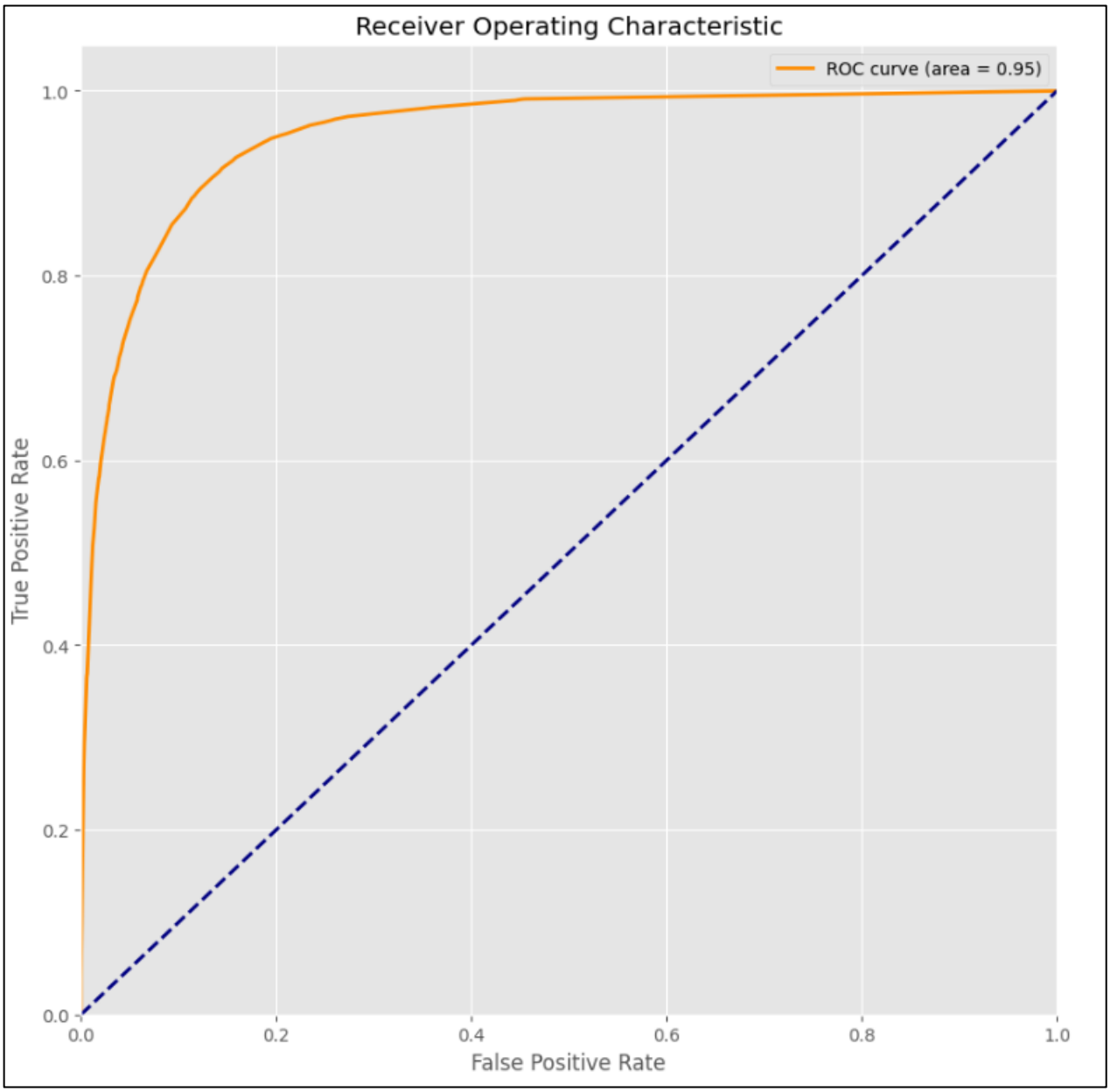

Рисунок 3.8 – ROC крива методу дерев рішень

ROC крива показує площу 0.95, що вказує на високу здатність розрізняти фішингові та нефішингові сайти, Матриця плутанини показує, що модель правильно ідентифікує фішингові сайти – 9138, нефішингові сайти – 17475, хибнопозитивні – 1499, хибнонегативні – 1912.

На рисунку 3.9 показано лістинг коду для методу випадкового лісу

```
model = RandomForestClassifier(n_estimators=80,max_depth=18,max_features='sqrt',min_samples_split=12,criterion='gini')
model.fit(X_train,y_train)

predictions_train = model.predict(X_train)
predictions_test = model.predict(X_test)

print(f"Train Accuracy Score: {accuracy_score(y_train,predictions_train)}")
print(f"Test Accuracy Score: {accuracy_score(y_test,predictions_test)}")
```

```
Train Accuracy Score: 0.9081838036914908
Test Accuracy Score: 0.8949840127897681
```

Рисунок 3.9 – Лістинг коду для методу випадкового лісу

Результати методу випадкового лісу: train accuracy score: 0.908, test accuracy score: 0.895.

На рисунку 3.10 показано графік, який відображає результати класифікації моделі для тестових даних, наведені у вигляді матриці помилок. Вона дозволяє оцінити відповідність між фактичними та передбаченими значеннями класів.

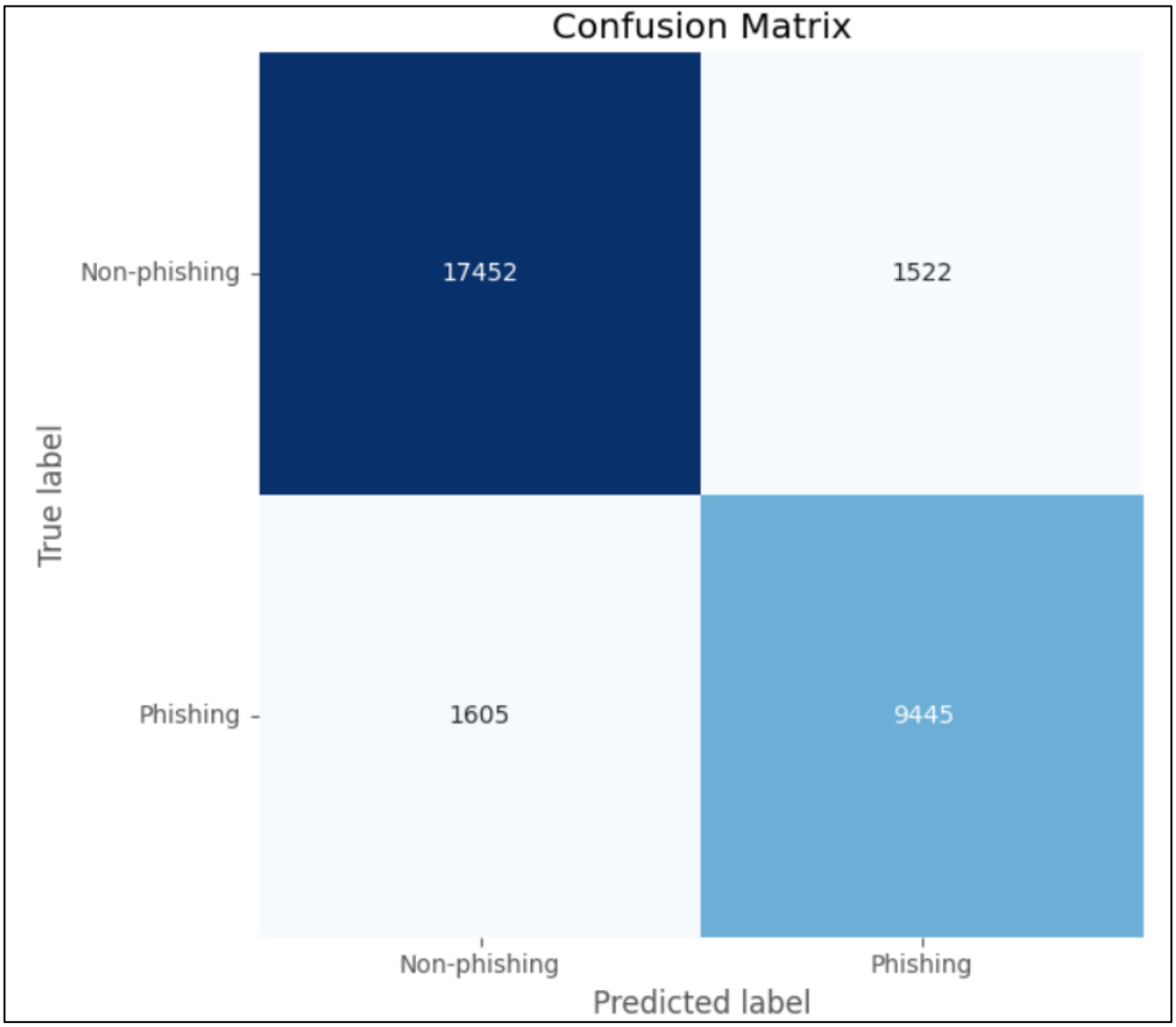

Рисунок 3.10 – Матриця плутанини методу випадкового лісу

На рисунку 3.11 показано графік, який використовується для оцінки продуктивності бінарного класифікатора, відображаючи зв'язок між чутливістю (True Positive Rate) та специфічністю (False Positive Rate) при різних порогах вирішення.

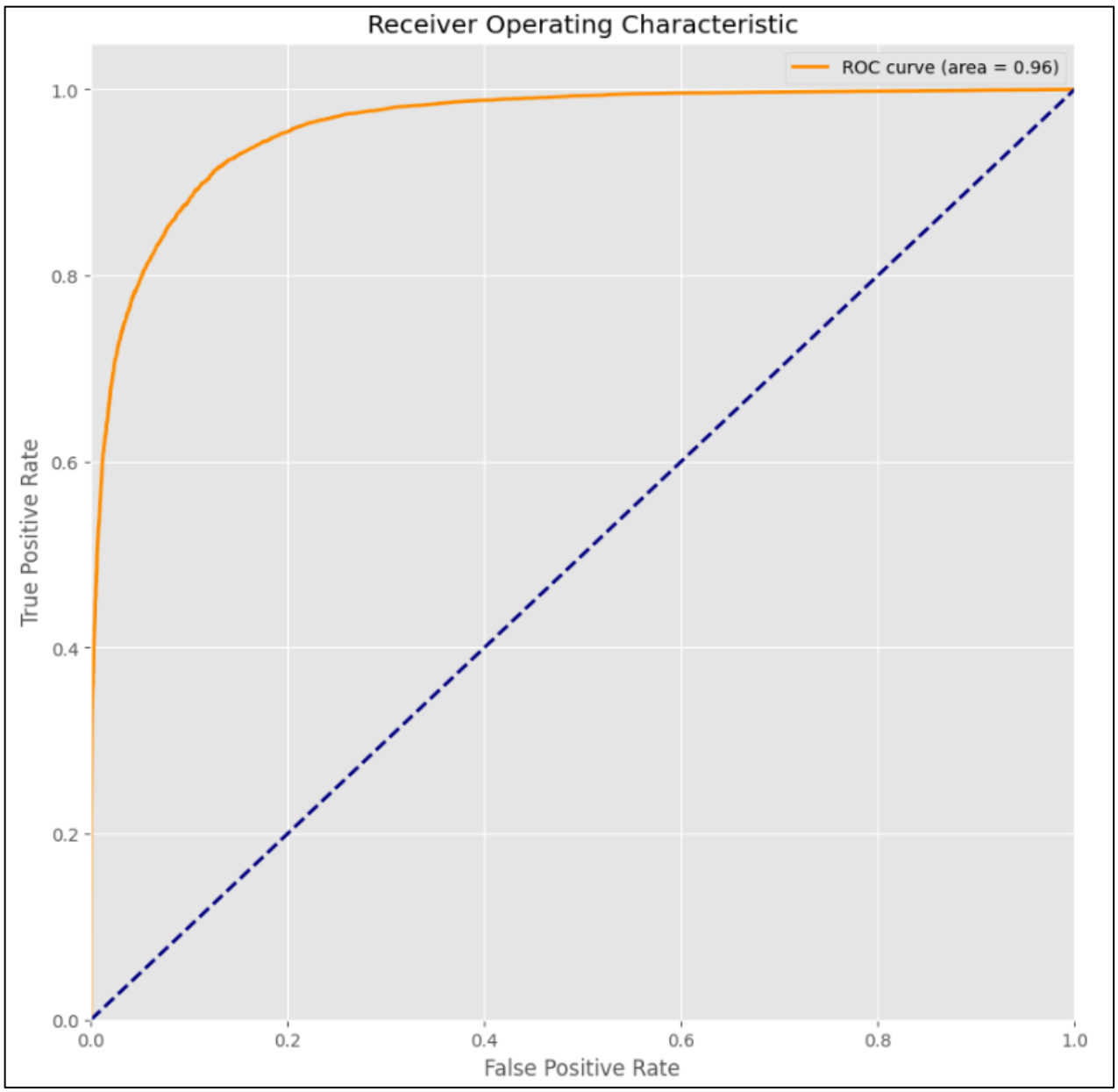

Рисунок 3.11 – ROC крива методу випадкового лісу

ROC крива показує площу 0.96, що вказує на високу здатність розрізняти фішингові та нефішингові сайти, Матриця плутанини показує, що модель правильно ідентифікує фішингові сайти – 9445, нефішингові сайти – 17452, хибнопозитивні – 1522, хибнонегативні – 1605.

На рисунку 3.12 показано лістинг коду для методу градієнтний бустинг.

```
model = GradientBoostingClassifier(n_estimators=200, learning_rate=0.01, max_depth=12,subsample=0.8,max_features=0.5, random_state=42)

model.fit(X_train, y_train)


predictions_train = model.predict(X_train)
predictions_test = model.predict(X_test)

print(f"Train Accuracy Score: {accuracy_score(y_train,predictions_train)}")
print(f"Test Accuracy Score: {accuracy_score(y_test,predictions_test)}")
```

```
Train Accuracy Score: 0.9114813070104064
Test Accuracy Score: 0.8960165201172395
```

Рисунок 3.12 – Лістинг коду для методу градієнтний бустинг

Результати методу градієнтний бустинг: train accuracy score: 0.911, test accuracy score: 0.896.

На рисунку 3.13 показано графік, який відображає результати класифікації моделі для тестових даних, наведені у вигляді матриці помилок. Вона дозволяє оцінити відповідність між фактичними та передбаченими значеннями класів.

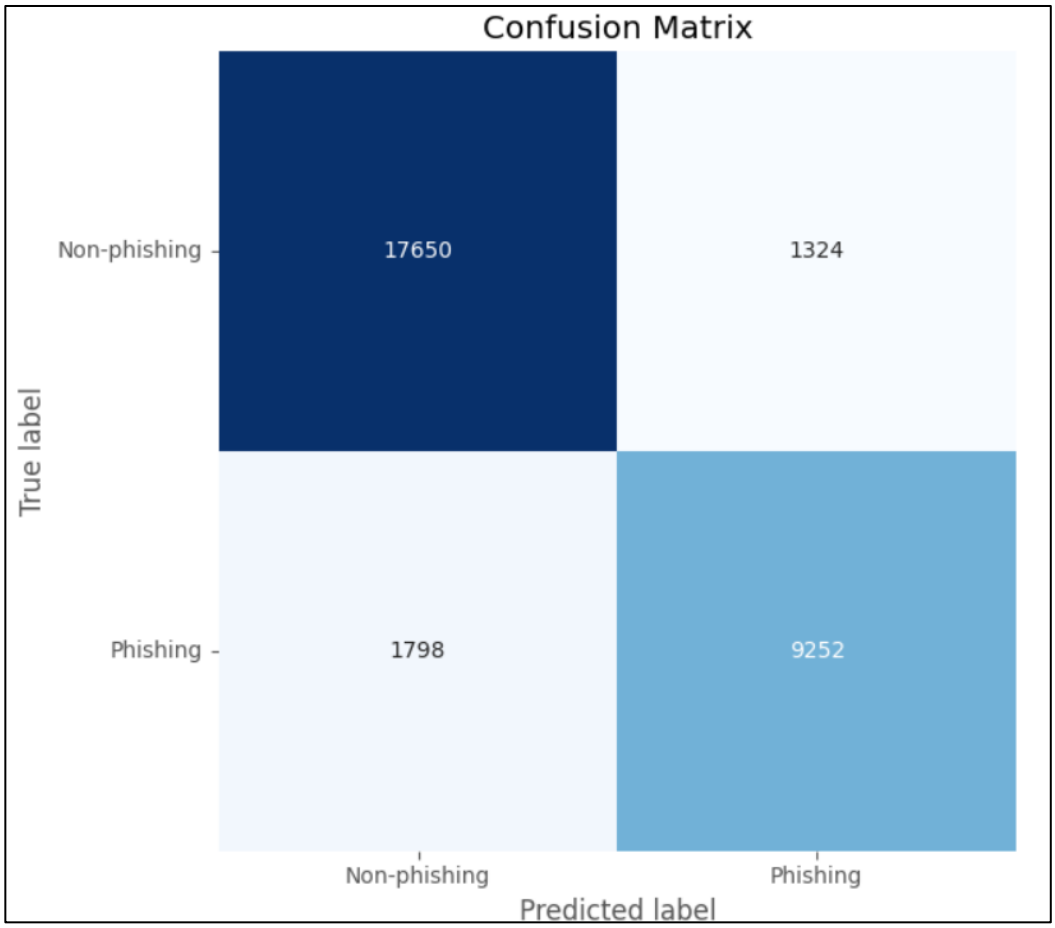

Рисунок 3.13 – Матриця плунтанини методу градієнтний бустинг

На рисунку 3.14 показано графік, який використовується для оцінки продуктивності бінарного класифікатора, відображаючи зв'язок між чутливістю (True Positive Rate) та специфічністю (False Positive Rate) при різних порогах вирішення.

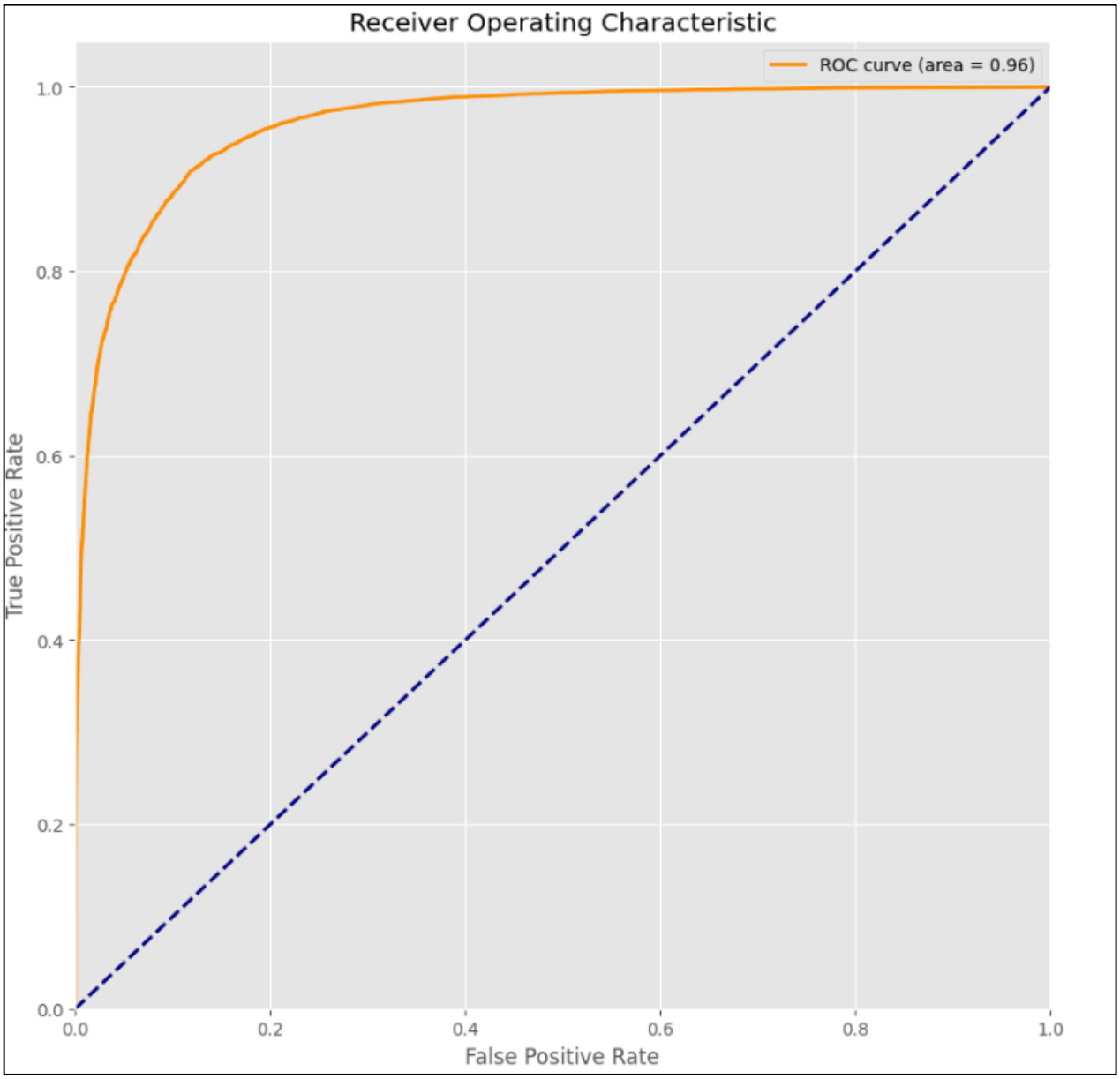

Рисунок 3.14 – ROC крива методу градієнтний бустинг

ROC крива показує площу 0.96, що вказує на високу здатність розрізняти фішингові та нефішингові сайти, Матриця плутанини показує, що модель правильно ідентифікує фішингові сайти – 9252, нефішингові сайти – 17650, хибнопозитивні – 1324, хибнонегативні – 1798.

В таблиці 3.1 представлено результати передбачення всіх методів.

Таблиця 3.1 – Результати передбачення

Назва	Train Accuracy Score	Test Accuracy Score
Логістична регресія	0.858	0.856
Дерева рішень	0.889	0.886
Випадковий ліс	0.908	0.895
Градієнтний бустинг	0.911	0.896

Проаналізувавши результати можна сказати, що найкращими методами є випадковий ліс та градієнтний бустинг, тому як результати досить схожі.

3.3 Висновки

В даному розділі розроблено інформаційну технологію передбачення фішингових сайтів на базі датасету про фішингові сайти [19] з використанням таких методів машинного навчання як логістична регресія, метод дерево рішень, випадковий ліс та градієнтний бустінг. Дослідження показали, що найкращими методами є випадковий ліс з результатом accuracy score: 0.895 на тестових даних та градієнтний бустинг з результатом accuracy score: 0.896 на тестових даних.

ВИСНОВКИ

В ході виконання дослідження проведено аналіз проблеми фішингових сайтів та обгрунтування вибору оптимальних методів для передбачення фішингових сайтів. З метою забезпечення безпеки в мережі Інтернет досліджено різноманітні підходи.

З метою розв’язання проблеми було здійснено вибір оптимального рішення та налаштувань та на основі вибраного рішення була розроблена технологія для аналізу та виявлення фішингових сайтів. Використовуючи методи машинного навчання такі як: логістична регресія, дерева рішень, випадковий ліс та градієнтний бустинг для аналізу даних, ця технологія дозволяє ефективно виявляти потенційно небезпечні веб-ресурси.

Технологія проводить аналіз на основі великої кількості параметрів, таких як структура URL, вміст веб-сторінки, метадані та інші ознаки, які можуть вказувати на фішингову активність. За допомогою алгоритмів машинного навчання, технологія постійно вдосконалює свої моделі виявлення, що дозволяє підвищити точність та зменшити кількість хибнопозитивних і хибнонегативних спрацьовувань.

Результатами показано, що найкращими методами є випадковий ліс з результатом accuracy score: 0.895 на тестових даних та градієнтний бустинг з результатом accuracy score: 0.896 на тестових даних.

Таким чином, розроблена технологія є ефективним інструментом у боротьбі з фішингом, забезпечуючи надійний захист для користувачів Інтернету та сприяючи підвищенню загального рівня безпеки.

За результатами даної роботи була зроблена доповідь на тему «Інформаційна технологія передбачення фішингових сайтів» на міжнародній науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи (Вінниця, 2023-2024 рр.)» з публікацією тез [2].

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гуржій. С.В. Організаційно-технічний аспект протидії фішингу. *Науковий журнал з проблем інформаційного права, правової інформатики, інформаційної і національної безпеки, інформації в інших галузях прав*. 2023. № 3(46). DOI: https://doi.org/10.37750/2616-6798.2023.3(46).287251

2. Литвиненко Д., Козачко О., Розвідувальний аналіз даних для інформаційної технології передбачення фішингових сайтів. *Міжнародна науково-практична інтернет-конференція «Молодь в науці: дослідження, проблеми, перспективи (Вінниця, 2023-2024 рр.)*. URL: https://conferences.vntu.edu.ua/index.php/mn/mn2024/paper/view/21263

3. Ayaburi, E. W., & Andoh-Baidoo, F. K. How do technology use patterns influence phishing susceptibility? A two-wave study of the role of reformulated locus of control. *European Journal of Information Systems*. 2023. Vol. 5 (12). P. 1-21. DOI: https://doi.org/10.1080/0960085X.2023.2186275

4. Чекмарьова. І.М. Шахрайство в Інтернеті як один із видів шахрайства. *Аналітично-порівняльне правознавство №2. Розділ 8. Кримінальне право та кримінологія*. 2024. С. 639-643. DOI: https://doi.org/10.24144/2788-6018.2024.02.106

5. Бережна О. Б., П. Р. Васькевич. Тенденції використання нейромереж у розробці веб-сайтів. *Міжнар. наук.-техн. конф. Харків: ТОВ «Друкарня Мадрид»*. 2024. С. 67-68. URL: https://openarchive.nure.ua/handle/document/26691

6. Anti-Phishing Working Group. URL: https://apwg.org/

7. Anti-Phishing Working Group. *Phishing Activity Trends Report, 1st Quarter*. 2024. P. 1-14. URL: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://docs.apwg.org/reports/apwg_trends_report_q1_2024.pdf

8. Google Safe Browsing. URL: https://safebrowsing.google.com/

9. Das Guptta, S., Shahriar, K.T., Alqahtani, H. et al. Modeling Hybrid Feature-Based Phishing Websites Detection Using Machine Learning

Techniques. *Ann. Data. Sci*. 2024. Vol. 11. P. 217-242. DOI: https://doi.org/10.1007/s40745-022-00379-8

10. Rubin H. Landau, Manuel J. Páez, Cristian C. Computational Physics: Problem Solving with Python. 2024. P. 3-555. URL: https://books.google.com.ua/books?id=naP8EAAAQBAJ&lpg=PR17&ots=dMrv7vN40f&dq=python&lr&hl=uk&pg=PR16#v=onepage&q=python&f=false

11. Kaggle. URL: https://www.kaggle.com

12. Чорненко, Ю. С. Математичне та програмне забезпечення системи виявлення шахраїв серед покупців інтернет-магазину. *Магістерська дис. : 113 Прикладна математика. Київ.* 2024. С. 1-103. URL: https://ela.kpi.ua/handle/123456789/66855

13. Das, A. Logistic Regression. *Encyclopedia of Quality of Life and Well-Being Research. Springer, Cham*. 2024. P. 3985-3986. DOI: https://doi.org/10.1007/978-3-031-17299-1_1689

14. Fazal Muhammad, Bilal Khan, Rashid Naseem. Liver Ailment Prediction Using Random Forest Model. *Statistical Learning - Science topic.* 2023. P. 1-17. URL: https://www.researchgate.net/publication/361849675_Liver_Ailment_Prediction_Using_Random_Forest_Model

15. Gomes Mantovani, R., Horváth, T., Rossi. Better trees: an empirical study on hyperparameter tuning of classification decision tree induction algorithms. *Data Min Knowl Disc*. 2024. Vol. 38. P. 1364-1416. DOI: https://doi.org/10.1007/s10618-024-01002-5

16. Goenka, R., Chawla, M. & Tiwari, N. A comprehensive survey of phishing: mediums, intended targets, attack and defence techniques and a novel taxonomy. *Int. J. Inf. Secur*. 2024. Vol. 23. P. 819-848. DOI: https://doi.org/10.1007/s10207-023-00768-x

17. Alnemari S, Alshammari M. Detecting Phishing Domains Using Machine Learning. Applied Sciences. *Applied Sciences.* 2023. Vol. 13 (8). P. 4649. DOI: https://doi.org/10.3390/app13084649

18. Arif Ali, Z., H. Abduljabbar, Z., A. Taher, H., Bibo Sallow, A. and Almufti, S.M. Exploring the Power of eXtreme Gradient Boosting Algorithm in

Machine Learning. *Academic Journal of Nawroz University*. 2023. Vol. 12 (2). P. 320-334. DOI: https://doi.org/10.25007/ajnu.v12n2a1612

19. FERNANDO DANIEL. Web Page Phishing Dataset. 2024. URL: https://www.kaggle.com/datasets/danielfernandon/web-page-phishing-dataset/data?select=web-page-phishing.csv

20. Литвиненко Д. Web Phishing Analysis. 2024. URL: https://www.kaggle.com/code/linean/web-phishing-analysis

Додаток А

(обов’язковий)

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації

ЗАТВЕРДЖУЮ
Завідувач кафедри САІТ
_____ д.т.н., проф. Віталій МОКІН
“____”_____ 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ
на бакалаврську дипломну роботу
ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПЕРЕДБАЧЕННЯ ФІШИНГОВИХ САЙТІВ
08-34.БДР.007.00.000 ТЗ

Керівник: к.т.н., доц.
_____ Олексій КОЗАЧКО
«__» _____ 2024 р.
Розробив студент гр. 2ІСТ-20б
_____ Данило ЛИТВИНЕНКО
«__» _____ 2024 р.

Вінниця 2024

1. Підстава для проведення робіт.

Підставою для виконання роботи є наказ №__ по ВНТУ від «__» _____2024р., та індивідуальне завдання на БДР, затверджене протоколом №__ засідання кафедри САІТ від «__» _____ 2024р.

2. Джерела розробки:

1) Anti-Phishing Working Group. *Phishing Activity Trends Report, 1st Quarter*. 2024. P. 1-14. URL: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://docs.apwg.org/reports/apwg_trends_report_q1_2024.pdf

2) Alnemari S, Alshammari M. Detecting Phishing Domains Using Machine Learning. Applied Sciences. *Applied Sciences*. 2023. Vol. 13 (8). P. 4649. DOI: https://doi.org/10.3390/app13084649

3. Мета і призначення роботи.

Метою дослідження є розроблення інформаційної технології для передбачення фішингових сайтів.

4. Вихідні дані для проведення робіт:

FERNANDO DANIEL. Web Page Phishing Dataset. 2024. URL: https://www.kaggle.com/datasets/danielfernandon/web-page-phishing-dataset/data?select=web-page-phishing.csv

5. Методи дослідження:

Використання платформи Kaggle, мови програмування Python та методів логістична регресія, метод дерево рішень, випадковий ліс та градієнтний бустінг.

6. Етапи роботи і терміни їх виконання:

a) Аналіз предметної області _____ – _____
b) Вибір оптимальних інформаційних технологій _____ – _____
c) Розвідувальний аналіз даних _____ – _____
d) Розроблення інформаційної технології _____ – _____
e) Оформлення матеріалів до захисту БДР _____ – _____

7. Очікувані результати та порядок реалізації

Отримання інформаційної технології для передбачення фішингових сайтів.

8. Вимоги до розробленої документації

Текстова та ілюстративна частини роботи оформлені у відповідності до вимог «Методичних вказівок до виконання бакалаврських кваліфікаційних робіт для студентів спеціальностей: 124 «Системний аналіз», 126 «Інформаційні системи та технології» (освітня програма «Прикладні інформаційні технології»)».

9. Порядок приймання роботи

Публічний захист «___» _____ 2024 р.
Початок розробки «___» _____ 2024 р.
Граничні терміни виконання БДР «___» _____ 2024 р.

Розробив студент групи 2ІСТ-20б _____ Данило ЛИТВИНЕНКО

Додаток Б

(обов'язковий)

Протокол перевірки бакалаврської дипломної роботи на наявність текстових запозичень

Назва роботи: «Інформаційна технологія передбачення фішингових сайтів»
Тип роботи: бакалаврська дипломна робота
Підрозділ: кафедра САІТ

Показники звіту подібності Unicheck

Оригінальність 88,3 Схожість 11,7

Аналіз звіту подібності (відмітити потрібне)

● Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
○ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і самостійності її автора. Роботу направити на розгляд експертної комісії кафедри.
○ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ (підпис) Сергій ЖУКОВ

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи _____ (підпис) Данило ЛИТВИНЕНКО

Керівник роботи _____ (підпис) Олексій КОЗАЧКО

Додаток В

(довідниковий)

Фрагмент лістингу програми

```
import numpy as np
import pandas as pd
import seaborn as sns
import xgboost as xgb
from scipy import stats
import statsmodels.api as sm
import matplotlib.pyplot as plt

from sklearn.metrics import roc_curve, auc
from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score,confusion_matrix
from sklearn.ensemble import RandomForestClassifier,GradientBoostingClassifier

plt.style.use('ggplot')
pd.set_option('display.max_rows', None)
pd.set_option('display.float_format', lambda x: '%.4f' % x)

data = pd.read_csv('/kaggle/input/web-page-phishing-dataset/web-page-
phishing.csv')
data.head(5)

data.info()

data.describe()

fig,ax = plt.subplots(nrows=2,ncols=1,figsize=(20,10))
for i,col in enumerate(['url_length','n_dots']):
    ax[i].boxplot(np.log(data.loc[data['phishing'] == 0,col]),
            vert=False, positions=[1], patch_artist=True,
            boxprops=dict(facecolor='lightgreen'),
            medianprops=dict(color='darkgreen'))
    ax[i].boxplot(np.log(data.loc[data['phishing'] == 1,col]),
            vert=False, positions=[2], patch_artist=True,
            boxprops=dict(facecolor='coral'),
```

```
                medianprops=dict(color='red'))
            ax[i].set_title(f'Distribution of Log {col.title()}: Normal vs. Phishing')
            ax[i].set_xlabel(f'Log {col.title()}')
            ax[i].set_yticks([1, 2], ['Normal', 'Phishing'])

        plt.tight_layout()
        plt.show()

        descriptive_stats = data.iloc[:,[2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,19]].groupby('phishing').describe()

        stat_data = descriptive_stats.xs('mean', level=1, axis=1)
        stat_data = np.sqrt(stat_data)

        n_features = len(stat_data.columns)
        index = np.arange(n_features)
        bar_width = 0.35

        plt.figure(figsize=(14, 8))
        plt.bar(index, stat_data.iloc[0], bar_width, label='Non-Phishing')
        plt.bar(index + bar_width, stat_data.iloc[1], bar_width, label='Phishing')

        plt.xlabel('Features')
        plt.ylabel('Mean Value')
        plt.title('Mean Values of Features by Phishing Category')
        plt.xticks(index + bar_width / 2, stat_data.columns, rotation=90)
        plt.legend()
        plt.tight_layout()
        plt.show()

        plt.figure(figsize=(20,10))
        sns.heatmap(data.corr(),cmap='coolwarm',annot=True,cbar=False)
        plt.show()

        sns.heatmap(data.corr())

        y = data.pop('phishing')

        X_train, X_test, y_train, y_test = train_test_split(data, y, test_size=0.3, random_state=42)
        model = LogisticRegression(max_iter=1000)
```

```
        model.fit(X_train, y_train)
        predictions = model.predict(X_test)

        def calculate_metrics(model, X_test, y_test):
            predictions = model.predict(X_test)
            print(f"Accuracy Score: {accuracy_score(y_test, predictions)}")
            confusion_matrix_plot(y_test, predictions)
            area_under_curve(model, X_test, y_test)

        def confusion_matrix_plot(y_test, predictions):
            cm = confusion_matrix(y_test, predictions)
            plt.figure(figsize=(8,6))
            sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", square=True, cbar=False,
xticklabels=['False', 'True'], yticklabels=['False', 'True'])
            plt.title('Confusion Matrix')
            plt.ylabel('True label')
            plt.xlabel('Predicted label')
            tick_marks = np.arange(2) + 0.5
            plt.xticks(tick_marks, ['Non-phishing', 'Phishing'], rotation=0)
            plt.yticks(tick_marks, ['Non-phishing', 'Phishing'], rotation=0)
            plt.tight_layout()
            plt.show()

        def area_under_curve(model, X_test, y_test):
            y_pred_proba = model.predict_proba(X_test)[:, 1]
            fpr, tpr, thresholds = roc_curve(y_test, y_pred_proba)
            roc_auc = auc(fpr, tpr)

            plt.figure(figsize=(10,10))
            plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (area = %0.2f)'
% roc_auc)
            plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
            plt.xlim([0.0, 1.0])
            plt.ylim([0.0, 1.05])
            plt.xlabel('False Positive Rate')
            plt.ylabel('True Positive Rate')
            plt.title('Receiver Operating Characteristic')
            plt.legend()
            plt.show()

        calculate_metrics(model, X_test, y_test)
```

```
        model = DecisionTreeClassifier(ccp_alpha= 0.0001, criterion= 'gini', max_depth=
32, min_samples_split= 5,random_state=42)
        model.fit(X_train,y_train)

        predictions_train = model.predict(X_train)
        predictions_test = model.predict(X_test)

        print(f"Train Accuracy Score: {accuracy_score(y_train,predictions_train)}")
        print(f"Test Accuracy Score: {accuracy_score(y_test,predictions_test)}")

        calculate_metrics(model,X_test,y_test)

        model                                                                    =
RandomForestClassifier(n_estimators=80,max_depth=18,max_features='sqrt',min_samples_spl
it=12,criterion='gini')
        model.fit(X_train,y_train)

        predictions_train = model.predict(X_train)
        predictions_test = model.predict(X_test)

        print(f"Train Accuracy Score: {accuracy_score(y_train,predictions_train)}")
        print(f"Test Accuracy Score: {accuracy_score(y_test,predictions_test)}")

        calculate_metrics(model,X_test,y_test)

        model    =    GradientBoostingClassifier(n_estimators=200,    learning_rate=0.01,
max_depth=12,subsample=0.8,max_features=0.5, random_state=42)

        model.fit(X_train, y_train)


        predictions_train = model.predict(X_train)
        predictions_test = model.predict(X_test)

        print(f"Train Accuracy Score: {accuracy_score(y_train,predictions_train)}")
        print(f"Test Accuracy Score: {accuracy_score(y_test,predictions_test)}")

        calculate_metrics(model,X_test,y_test)
```

Додаток Г

(обов’язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПЕРЕДБАЧЕННЯ ФІШИНГОВИХ САЙТІВ

Нормоконтроль: к.т.н., доцент

_____ Сергій ЖУКОВ

«___» _____ 2024 р.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100077 entries, 0 to 100076
Data columns (total 20 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   url_length      100077 non-null  int64
 1   n_dots          100077 non-null  int64
 2   n_hypens        100077 non-null  int64
 3   n_underline     100077 non-null  int64
 4   n_slash         100077 non-null  int64
 5   n_questionmark  100077 non-null  int64
 6   n_equal         100077 non-null  int64
 7   n_at            100077 non-null  int64
 8   n_and           100077 non-null  int64
 9   n_exclamation   100077 non-null  int64
 10  n_space         100077 non-null  int64
 11  n_tilde         100077 non-null  int64
 12  n_comma         100077 non-null  int64
 13  n_plus          100077 non-null  int64
 14  n_asterisk      100077 non-null  int64
 15  n_hastag        100077 non-null  int64
 16  n_dollar        100077 non-null  int64
 17  n_percent       100077 non-null  int64
 18  n_redirection   100077 non-null  int64
 19  phishing        100077 non-null  int64
dtypes: int64(20)
memory usage: 15.3 MB
```

Рисунок Г.1 – Огляд даних датасету

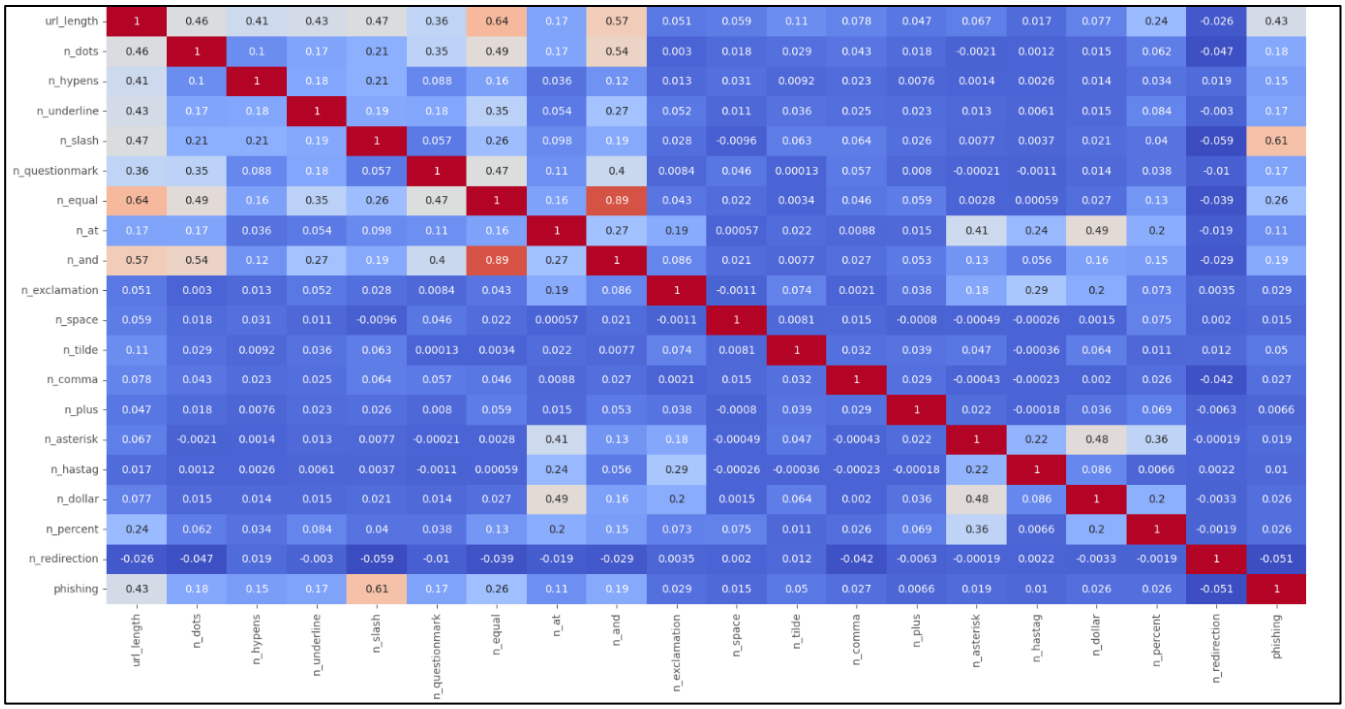

Рисунок Г.2 – Графік теплової карти

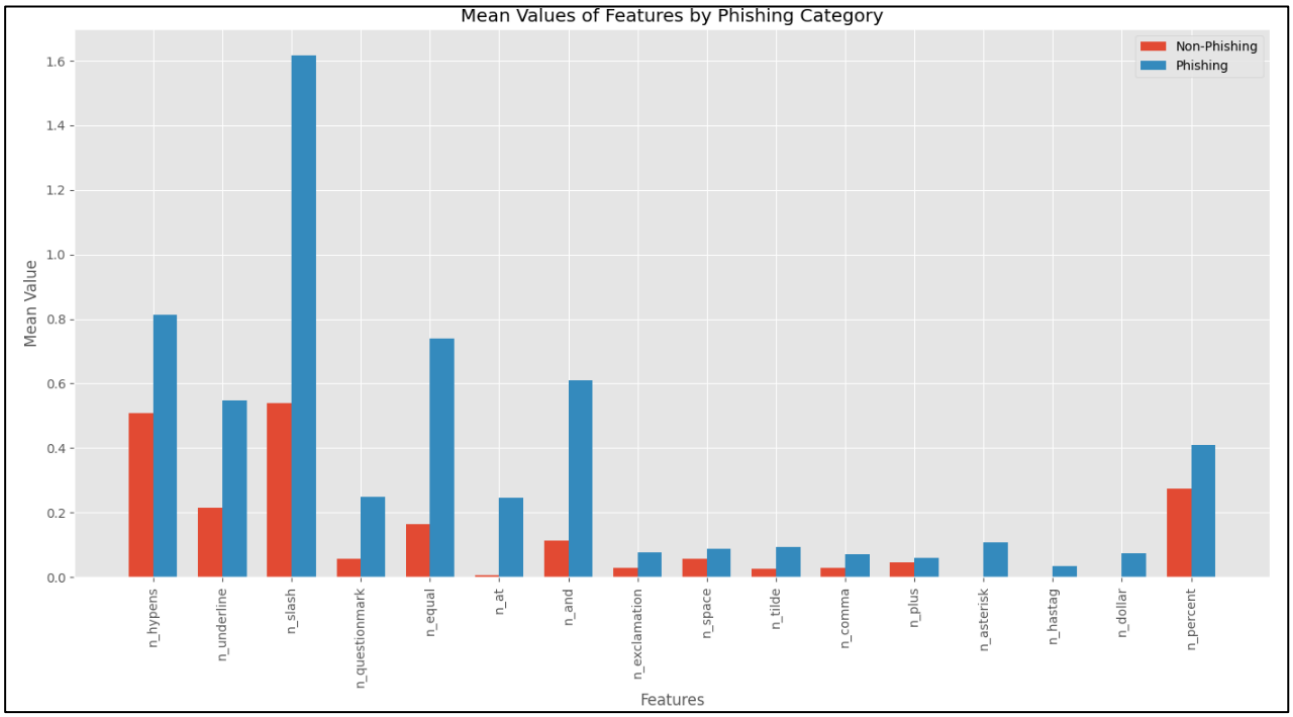

Рисунок Г.3 – Графік порівняння середніх значень ознак для фішингових та нефішингових сайтів

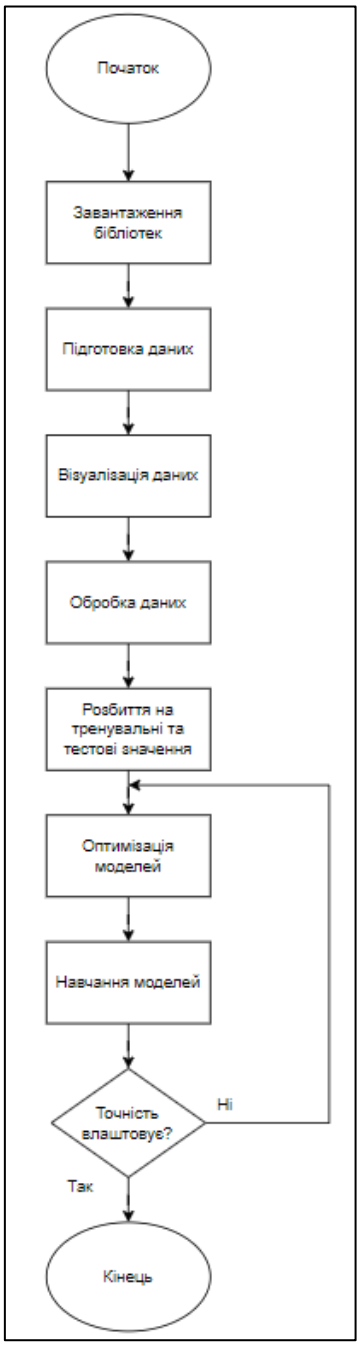

Рисунок Г.4 – Блок-схема алгоритму інформаційної технології

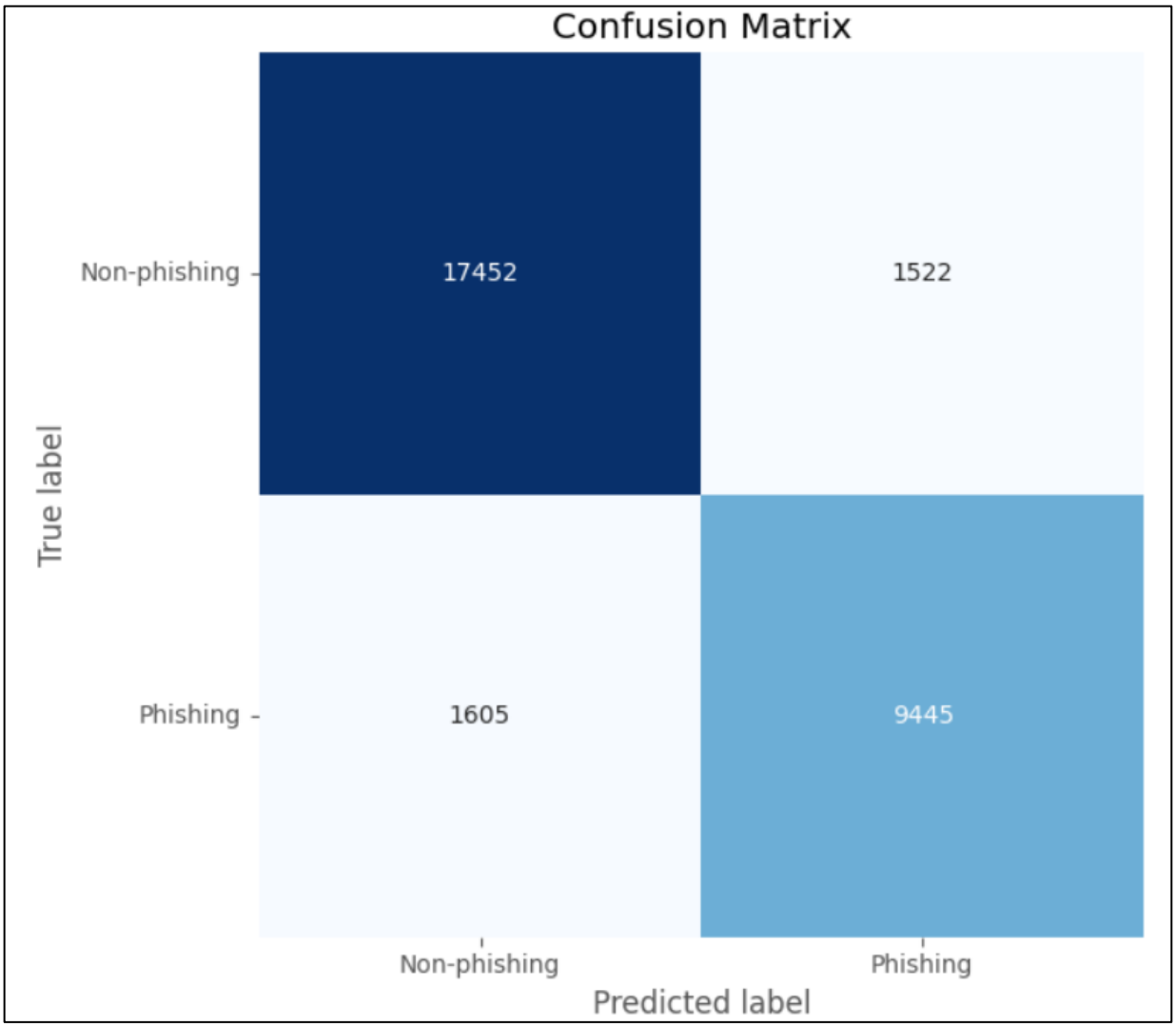

Рисунок Г.5 – Матриця плутанини випадкового лісу

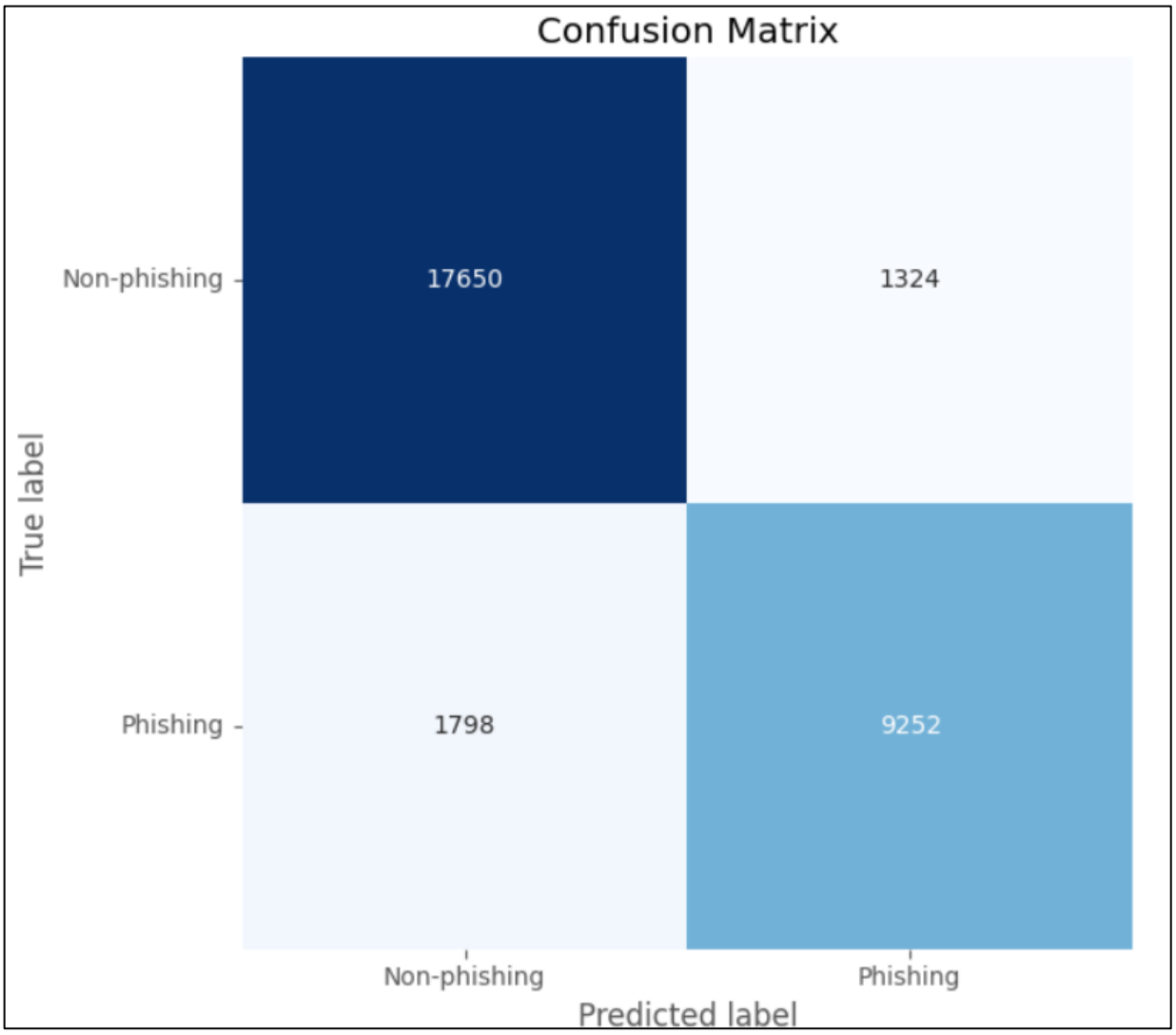

Рисунок Г.6 – Матриця плутанини градієнтного бустингу