A Comparative Analysis of Anti-Phishing Website Techniques: Identifying Optimal Approaches to Enhance Cybersecurity

YAU JIA XIN

A project report submitted in partial fulfilment of the requirements for the award of Master of Data Management and Analytics

Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

December 2023

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature	:	
Name	:	YAU JIA XIN
ID No.	:	2206813
Date	:	8/12/2023

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"A COMPARATIVE ANALYSIS OF ANTI-PHISHING WEBSITE TECHNIQUES: IDENTIFYING OPTIMAL APPROACHES TO ENHANCE CYBERSECURITY"** was prepared by **YAU JIA XIN** has met the required standard for submission in partial fulfilment of the requirements for the award of Master of Data Management and Analytics at Universiti Tunku Abdul Rahman.

Approved by,

Signature : Kai

Supervisor : Dr Chia Kai Lin

Date : 20 December 2023

Signature : __________

Co-Supervisor : __________

Date : __________

ABSTRACT

Internet security is continuously threatened by phishing attacks; therefore, the ability to identify fraudulent websites is crucial in order to prevent users from being duped into divulging sensitive information. Consequently, it is critical to identify effective detection techniques for fraudulent websites. The research consists of analysing the characteristics of phishing websites, extracting their essential features using the wrapper method, and classifying websites as phishing or legitimate using supervised and unsupervised learning algorithms. The study evaluates and compares the efficacy of multiple machine learning algorithms, including the Autoencoder classifier, Extreme Gradient Boost (XGBoost), and Random Forest classifier, using metrics such as accuracy, precision, recall, and F1-score. Random Forest, with an impressive accuracy rate of 97.03%, demonstrates its exceptional capability in accurately categorising websites that are fraudulent or legitimate in nature. By integrating the Google Safe Browsing List and the Random Forest classifier, a web application is created. Upon receiving the user's URL, the web application utilises a pre-trained Random Forest classifier to ascertain the probability that the requested URL is a fraud site. As an additional layer of security, the Google Safe Browsing List is utilised to verify the output produced by the Random Forest classifier. It is expected the fact that the research will result in the development of phishing detection technologies that are more precise and efficient, thereby bolstering online security and protecting users against identity and financial deception.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

AUC	Area Under the Curve
API	Application Programming Interface
APWG	Anti-Phishing Working Group
BDE	Bi-directional Elimination
CNN	Convolutional Neural Network
DDQN	Double Deep Q-Network
DNS	Domain Name System
DQN	Deep Q-Network
DT	Decision Tree
FN	False Negative
FP	False Positive
GSB	Google Safe Browsing
GRU	Gated Recurrent Unit
HTML	Hypertext Markup Language
HTTPS	Hypertext Transfer Protocol Secure
IBM	International Business Machines Corporation
KNN	K-Nearest Neighbour
LSTM	Long Short-Term Memory
MCC	Matthews Correlation Coefficient
NGROK	Ngrok (a tool for creating secure tunnels to localhost)
OLS	Ordinary Least Squares
ROC	Receiver Operating Characteristic
RNN	Recurrent Neural Network
SMS	Short Message Service
SMOTE	Synthetic Minority Over-sampling Technique
SVM	Supporting Vector Machine
TN	True Negative
TP	True Positive
URL	Uniform Resource Locator
WHOIS	Who Is
XGBoost	eXtreme Gradient Boosting

CHAPTER 1

INTRODUCTION

1.1 General Introduction

Anti-Phishing Working Group (APWG) defines phishing as a crime using social engineering and technical deception to obtain personal identity information and financial account credentials (APWG, 2022). Social engineering methods use fake email addresses and communications to trick victims into believing they are interacting with a trusted, legitimate entity. They redirect consumers to fake websites that steal financial data including usernames and passwords. Technological subterfuge techniques install malware on computers to steal credentials directly, often by intercepting account usernames and passwords or redirecting consumers to fake websites.

Based on IBM's Cost of a Data Breach Report 2022, phishing is the second most prevalent and most expensive initial attack vector that leads to data breaches, costing firms an average of $4.91 million per incident (IBM, 2022). Poor cybersecurity policies and frequent data breaches can cause a company to lose customers and investors, resulting in a loss in market value and economic effect. In 2022, APWG recorded a total of 1,270,883 phishing assaults (APWG, 2022). Since the start of 2021, ransomware has affected fewer businesses than at any other time. In addition to the obvious financial cost, phishing attempts can harm a company's reputation, which can have long-term effects on the economy. Hence, phishing attacks can result in huge financial losses for people, corporations, and even entire economies.

Protecting individuals, businesses, and the economy requires phishing website detection. It helps reduce financial losses and maintain client confidence, which are essential for an economy. Protecting clients from identity theft and financial harm by detecting phishing websites can prevent them from sharing sensitive information. Enterprises can avoid financial loss, reputation damage, and legal liability by identifying and preventing phishing. Identifying phishing websites helps law enforcement prosecute criminals and protect victims.

1.2 Importance of the Study

Phishing attacks come in different forms, including email phishing, phone phishing (vishing), and SMS phishing (smishing). A prevalent sort of phishing attack involves the creation of fraudulent websites that imitate legitimate platforms, such as online banking or e-commerce websites. Fraudulent websites can include design elements and branding that closely imitate those of real websites, hence presenting difficulties for consumers in differentiating between the two.

For phishing website detection, list-based, heuristic, machine learning, and deep learning methods have been developed and published. List-based phishing website detection detects and marks likely phishing websites by cross-referencing website URLs with a pre-established inventory of known URLs. Heuristic methods that use rules and algorithms to identify common website characteristics detect phishing websites. Training a classifier on a dataset of authentic and fraudulent websites can help machine learning algorithms detect phishing websites. Deep learning uses multilayered artificial neural networks to interpret data representations.

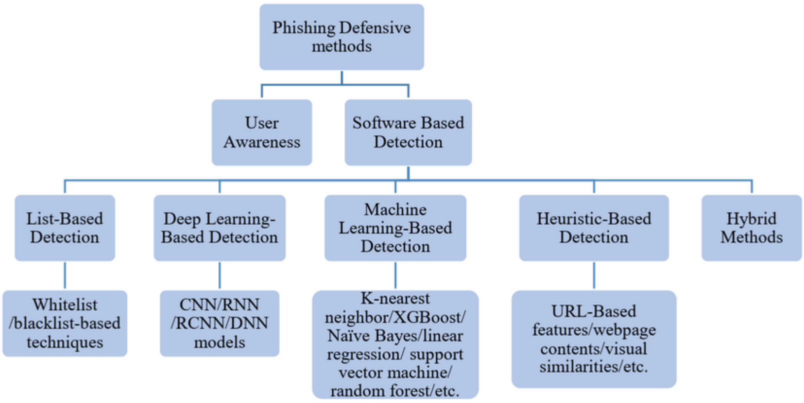

Table 1.1 Phishing Detection Method

Although there are many detection strategies available, there is always a need for more effective and precise methodologies to identify phishing websites. In addition, there is a lack of comparative research that evaluate the efficacy of various detection approaches, especially in real-life situations. Given this, the aim of this study is to evaluate and compare the effectiveness of several phishing detection algorithms using a large dataset that includes both legal websites and phishing websites. The evaluation of various methodologies will be conducted using a diverse set of performance indicators, encompassing accuracy, precision, recall, and F1-score.

1.3 Problem Statement

A growing number of sophisticated phishing attempts use user behaviour and technology infrastructures to threaten internet security. Traditional phishing detection systems are failing to combat attackers' changing strategies. Anti-phishing technologies vary in effectiveness, so Random Forest, XGBoost, and autoencoder must be evaluated and compared. This research fills the gap in knowing which method is more effective and adaptable to various phishing attacks. The lack of widely established user-friendly software with cutting-edge anti-phishing technologies makes these solutions difficult to apply. Thus, this project seeks to determine the best anti-phishing technique and create an efficient and accessible online application that uses it, thereby increasing users' resilience to digital phishing threats.

1.4 Aims and Objectives

This study compares Random Forest, XGBoost, and autoencoder, three popular anti-phishing methods. This involves evaluating their accuracy, precision, recall, and other performance factors. To evaluate each strategy's ability to detect phishing websites, the study will use existing research, datasets, and tests. Based on comparative analysis, the research seeks to determine the best anti-phishing website strategy among Random Forest, XGBoost, and autoencoder. The "best" strategy may have excellent accuracy, robustness against diverse phishing assaults, scalability, and computational economy. Performance metrics will be thoroughly assessed during identification. After identifying the optimal anti-phishing strategy, the research purpose is to use it. The goal is to create a user-friendly web app that uses the best phishing website detection method. A real-time phishing defence web application should analyse URLs or web content.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This literature review summarises anti-phishing and cybersecurity research and knowledge. In the digital age, phishing assaults can have devastating effects on individuals and businesses. Understanding phishing methods helps protect sensitive data and enhance online defences. The review examines anti-phishing research and scholarly literature. It evaluates different methods and prepares for a comparative review of anti-phishing website tactics by examining their pros and cons. Throughout the review, gaps in the literature and topics for further investigation are highlighted. The literature review contextualises the current study's aims, guides its methodology, and adds to cybersecurity knowledge by critically synthesising and assessing earlier studies.

2.2 Literature Review

Most of the time, the activities of notorious cybercriminals are effective due to the absence of a proven method that might provide folks with accurately predicted information at the appropriate time or as required. Research and models based on machine learning and deep learning can play a significant role in the development of such technologies.

Numerous research utilise list-based techniques for the detection of phishing websites. In this case, (Cao, et al., 2008) has presented a novel automated allowlist system that effectively maintains and updates a collection of IP addresses associated with login-page websites. The system has shown remarkable performance in terms of its functionality and efficiency. However, the effectiveness of this method may be compromised if it relies on the active participation of users and fails to detect newly discovered fraudulent websites. (Jain & Gupta, 2016) conducted a study wherein they utilised URL and DNS matching techniques in conjunction with a white-list strategy. The implementation of this integration resulted in enhanced operational efficiency and a diminished occurrence of false negatives. However, valid domains not included in the approved list may be inadvertently omitted by this methodology, leading to a few instances of erroneous identification. A combination of list-based, visual similarity, heuristic, and machine learning methodologies were utilised by Maroofi and colleagues. The Random Forest classifier yielded a notable accuracy rate of 97.00% (Maroofi, et al., 2020). However, the dependence on external features impeded the pace of the procedure.

The application of machine learning methodologies has been widely implemented in order to identify fraudulent websites. Abusaimeh's research employed a range of classification algorithms, such as Supporting Vector Machine (SVM), Random Forest, and Decision Tree. The results demonstrated a high accuracy rate of 98.52% (Abusaimeh, 2021). However, as a consequence of incorporating these classifiers, the computational complexity increased. Gupta et al. utilised the ISCXURL-2016 dataset and implemented four machine learning classifiers in their research. Among these classifiers, Random Forest exhibited the greatest accuracy rate of 99.57% (Gupta, et al., 2021). An inherent drawback of the research was the insufficiency of varied training and evaluation datasets. The research conducted by Butnaru et al. involved the training of classifiers with a dataset comprising one hundred thousand URLs. The researchers reported a notable achievement of 99.29% accuracy while employing the optimised Random Forest algorithm (Butnaru, et al., 2021). This performance surpassed the accuracy of Google Safe Browsing. In a study conducted by Stobbs, a Random Forest model was employed with feature selection and hyperparameter optimisation, resulting in an accuracy rate of 99.33% (Stobbs, et al., 2020). The precise division ratio between the training and testing datasets was withheld. Kumar et al. conducted their research utilising the UCI ML Repository as their data source and the Random Forest classifier to detect phishing and spam emails with an exceptional degree of precision (Kumar, et al., 2018).

Considerable attention has been directed towards the application of deep learning algorithms in the domain of fraudulent website detection. Feng and Yue conducted a study in which they utilised heuristic methods and deep learning techniques to construct RNN models that incorporated LSTM and GRU architectures. Their objective was to detect phishing attacks, and their approach yielded a detection accuracy of 99.50% (Feng & Yue, 2020). In their study, Seok and Sung employed a hybrid methodology that integrated deep learning techniques with heuristic approaches, resulting in a notable enhancement of sensitivity by 3.98% (Seok & Sung, 2021). Yang et al. classified URLs in their research utilising a convolutional neural network (CNN) in conjunction with a long short-term memory (LSTM) model. The researchers successfully attained a remarkable accuracy rate of 98.99% (Yang, et al., 2018). Nevertheless, it is important to acknowledge that the incorporation of WHOIS data into the URL functionalities may potentially result in disruptions to operations. Saha et al. achieved noteworthy outcomes in their research by employing the Multilayer Perceptron Neural Network (MPN) architecture. Specifically, they achieved a training accuracy of 95.0% and a testing accuracy of 93.00% (Saha, et al., 2020). In their study, Maci and his team introduced a classifier based on a double deep Q-Network (DDQN) approach, which demonstrated superior performance compared to alternative deep learning techniques in the context of web phishing detection.

The significance of visual similarity is substantial in specific scholarly inquiries. Dooremaal utilised a methodology in which textual and visual components of web pages were extracted, along with the corresponding screenshots. The primary aim of this strategy was to efficiently identify fraud websites, which was accomplished with a significant degree of precision. Azeez utilised a visual similarity technique in combination with a whitelist to effectively detect fraudulent websites, attaining a noteworthy accuracy rate of 95.0 percent (Azeez, et al., 2021). However, the scope of Azeez's research was limited to a mere 200 websites, of which 60 were genuine and 140 were fraudulent sites. Abdelnabi conducted a study in which he evaluated numerous visual similarity approaches; the LBET model achieved a detection accuracy surpassing 97.5% (Abdelnabi, et al., 2020). However, the dataset utilised in their study consisted of only 11,055 incidents.

In 2019, Nathezhtha proposed a tripartite approach for detecting phishing attacks, which incorporates DNS blacklists, heuristics, and web crawlers (Nathezhtha, et al., 2019). The implemented technique involved the extraction of web URLs, which were subsequently matched against the DNS blacklist. Additionally, the strategy included the crawling of website pages and the extraction of heuristic analysis characteristics. This comprehensive approach resulted in successful identification. Nevertheless, this approach was dependent on the functionalities of search engines, which might potentially result in a decrease in the speed of the procedure.

The aforementioned methodologies exemplify the range of techniques employed in the identification of phishing websites, each possessing distinct advantages and drawbacks. Scholars persist in refining and advancing these methodologies to enhance the precision and efficacy in detecting phishing hazards.

Table 2.1 provides a summary of existing techniques for detecting phishing websites, along with their respective explanations and limitations.

Table 2.1 Summary of Existing Phishing Websites Detection Research

Author & Year	Techniques Used	Dataset Used	Explanation	Limitation
(Feng & Yue, 2020)	Deep learning & heuristic	The study uses 1.5 million URLs, 51% legal and 49% phishing. Phishing URLs come from PhishTank, whereas real URLs come from Common Crawl.	The study suggested four RNN models which were RNN and bi-directional RNN with Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) architectures for phishing attack detection that simply use lexical aspects of URLs. It demonstrated that RNN models were capable of 99.50% detection accuracy.	In the investigation, a single algorithm was examined. Only 17 features were retrieved from a data collection of 1.5 million URLs.
(Abusaimeh, 2021)	Machine learning	N/A	The research utilised Decision Tree (DT), Supporting Vector Machine (SVM), and Random Forest classification	The proposed approach increases the model's computational expense and complexity.

			algorithms. The accuracy of the combination of three classifiers has achieved 98.52%.	
(Gupta, et al., 2021)	Machine learning	ISCXURL-2016 dataset where 11964 instances of legitimate and phishing URLs are used.	Nine phishing websites features are evaluated against four different machine learning classifiers, namely Random Forest, K-Nearest Neighbour (KNN), SVM, and Logical Regression. The Random Forest algorithm achieved the maximum accuracy of 99.57 %.	To determine the robustness of the suggested method, the study has not yet employed the various training and test datasets.
(Seok & Sung, 2021)	Deep learning & heuristic	A total of 222,541 URLs were gathered from Phishstorm and Phishtank, which are sources of phishing URLs.	The suggested model coupled a convolution operation with a deep convolutional autoencoder to consider the nature of zero-day attacks. According to the study, the sensitivity increased by	Among the various components of URLs, only the character-level characteristics were optimised. Given the structure of the web address, which comprises of domains and

		Conversely, valid URLs were obtained via the Open Directory Project.	3.98 % compared to earlier research.	subdomains, it is possible to foresee additional performance increases.
(Rao, et al., 2022)	Machine learning		The research employed domain-specific HTML source code text and word embedding extracted from plain text. Utilizing ensemble and multimodal techniques, they developed several word embeddings to evaluate their model.	Unfortunately, the proposed approach is dependent exclusively on plain text and domain-specific terminology, and it may fail if images are replaced for text.
(Dooremaal, et al., 2021)	Visual similarity & machine learning	Phishing web pages were obtained from 100,000 URLs submitted in feeds like OpenPhish, PhishTank,	The approach extracted textual and visual features from a web page and its screenshot as search terms to find comparable websites through search engines. The system under consideration	The strategy relies on third-party search engine-based filtering, which may yield varying results for the same query over time.

		and PhishStats.	demonstrates a high level of accuracy, achieving a rate of 99.20% for target identification and 99.66% for phishing categorization using logistic regression when evaluated on a specific dataset.	
(Maroofi, et al., 2020)	List based, visual similarity, heuristic & machine learning	URLs from phishing blacklists (APWG, PhishTank, OpenPhish) and malware distribution blacklists (URLhaus).	The classification methods are Logistic Regression and Random Forest. Each approach was applied to malware and phishing datasets independently. The system achieved 97.00% accuracy with the Random Forest classifier.	The study utilised only two machine learning algorithms. Every 5 minutes to 1 hour, the system downloaded updated URL blacklists. The inclusion of third-party features slowed down the procedure.
(Azeez, et al., 2021)	White-list-based &	140 phishing URLs were obtained	The study handled phishing with an automatic white-	The study only analysed 200 sites, including

	visual similarity	from PhishTank whereas 60 legitimate URLs from Alexa	list. This technique effectively detected phishing sites with 95.0% accuracy by verifying the correctness and legality of a webpage using specific hyperlink or URL attributes.	140 phishing and 60 legitimate websites.
(Nathezhtha, et al., 2019)	DNS blacklist, heuristic & visual similarity	Datasets were collected from real phishing cases.	Researchers introduced three-phase phishing attack detection. WC-PAD uses DNS blacklists, heuristics, and web crawlers. This method extracted the web URL and matched it to the DNS blacklist, then web crawlers crawled each website page and extracted features. Web crawlers extract three heuristic analysis features: web content, URL, and web traffic.	The strategy relies on search engine features, which can slow down the process.

(Butnaru, et al., 2021)	Machine learning & heuristic	The classifiers were trained on 100,000 URLs, including 40,000 benign and 60,315 PhishTank phishing URLs. The phishing detection engine was tested on 380,000 benign and phishing URLs. This dataset had 305,737 benign and 74,436 phishing URLs.	The proposed phishing detection engine utilised supervised machine learning algorithms such as Naive Bayes, Decision Tree, Random Forest, Support Vector Machine, and Multi-Layer Perceptron. Common metrics are used to compare machine learning model performance and the results were compared with GSB. Highest accuracy achieved by optimized Random Forest is 99.29%.	The performance of Google Safe Browsing (GSB) is fairly low compared to the proposed phishing detection engine.
(Rao & Pais, 2020)	List based, visual similarity, heuristic & machine learning	A total of 4097 instances was obtained from PhishTank and a total of	The researchers developed an ensemble model comprising Random Forest (RF), Extra-Tree, and XGBoost to	Overall, the system has a high response time due to the complexity of the system.

		5438 instances was obtained from Google.	evaluate blacklist and heuristic filters jointly. The model achieved 98.72% accuracy and 97.39% MCC.	
(Cao, et al., 2008)	List based & machine learning	PhishTank picked 18 of 34 phishing websites and the remaining 16 websites are legitimate for training process. 10 phishing URLs were selected from PhishTank and % legitimate websites were selected for testing process.	An automated allowlist that updates itself with IP addresses of login-page websites. The proposed approach displayed excellent performance, with 100% true positives and 0% false negatives using Naive Bayesian classifier.	The research tested a limited number of websites that a common user would log in. This method relies on user participation and cannot detect new phishing websites.
(Jain & Gupta, 2016)	List based & heuristic	The collection includes 1525 webpages	The URL and DNS matching module, with a white-list, improves running performance and	It extensively compares URL parent domains to specified whitelists. This

		(1120 phishing and 405 legitimate). PhishTank collects phishing sites. Legitimate websites are identified using Alexa, Stuffgate, and online payment providers.	reduces false negatives. The second module, phishing identification, detects phishing websites. Extract hyperlinks from the webpage using Jsoup and identifies the parent domains of these links with Guava libraries. The proposed approach effectively prevents phishing attempts with an 86.02 % true positive rate and lesser than 1.48 % false negative rate, according to testing results.	method may miss legitimate domains not on the whitelist, resulting in false positives.
(Yang, et al., 2018)	Deep learning, heuristic & machine learning	From PhishTank, 1021758 phishing URLs were analysed, while dmoztools.n	The study uses CNN-LSTM. Local characteristics are extracted by CNN and context dependency by LSTM. CNN-	The approach screened out similar phishing websites and those without login forms, then retrieved 15 distinct features

		et provided 989021 legitimate URLs as negative samples.	LSTM results are used by XGBoost for categorization. The accuracy is 98.99%, with a false positive rate of 0.59%.	from URL vocabulary, HTML DOM, WHOIS, and search engine data. Using WHOIS information in URL features may slow down the operation.
(Saha, et al., 2020)	Deep learning, heuristic & machine learning	The dataset, gathered through Kaggle, includes 10,000 webpages.	The Multilayer Perceptron Neural Network (MPN) model achieved 95.0% accuracy during training and 93.00% during testing.	A limited number of features is extracted from the instances.
(Maci, et al., 2023)	Deep learning	Mendeley dataset with 30,647 phishing URLs and 58,000 Legitimate URLs.	The research proposed double deep Q-Network (DDQN) based classifier and compared the performance of model with different deep learning methods such as DNN, CNN, LSTM and BiLSTM. the proposed approach	The DRL framework has not been studied for web phishing detection and its training period is lengthy.

			outperforms best-precision and best-recall in five out of six measures for web phishing imbalanced classifiers.	
(Stobbs, et al., 2020)	Machine learning, heuristic & list based	Phish tank and Alexa	With 99.33% accuracy, Random Forest with PSO for feature selection and TPE for hyperparameter optimisation is the most effective combo.	The study utilised various ML algorithms but did not disclose the training/testing split ratio. Only recall and accuracy outperform other methods.
(Abdelnabi, et al., 2020)	Visual similarity, machine learning	11,055 instances were obtained from UCI Machine learning Repository and Kaggle	The research compared different methods including LBET, RoFBET, ABET and BET. The LBET model attained detection accuracy above 97.5%.	The phishing website data set used in this study has just 11,055 instances.
(Kumar, et al., 2018)	Heuristic, Machine learning, Deep learning	The UCI ML Repository contains 2949 valid emails, 1378 spam emails,	The Random Forest classifier accurately detects phishing and spam emails with 97.7% and 89.2%	The study utilised only two classifiers which are random forest and multilayer

		11,000 URL occurrences, and 30 characteristics.	accuracy, respectively.	perceptron. Training and testing use the same dataset.
(Azeez, et al., 2021)	List based, Visual similarity	140 phishing URLs were obtained from PhishTank and 60 legitimate URLs were obtained from Alexa.	The system attained an average accuracy of 96.17% after six experiments.	The study used a dataset with only 200 instances.

To conclude, this literature analysis highlights the ongoing endeavours of researchers to enhance and progress phishing detection systems. Each of the solutions mentioned has distinct advantages and disadvantages, highlighting the importance of a comprehensive and flexible strategy to combat the ever-changing methods employed by cybercriminals. Continued research and development are crucial to improve the accuracy, effectiveness, and real-time responsiveness of identifying and preventing phishing threats as the field advances.

CHAPTER 3

METHODOLOGY AND WORK PLAN

3.1 Introduction

For phishing website detection, Random Forest, XGBoost, and Autoencoder are recommended. The research uses Random Forest, XGBoost, and Autoencoder because of their complementing characteristics. The Random Forest and XGBoost algorithms provide ensemble-based accuracy and resilience, while the Autoencoder method detects unsupervised anomalies. Comparing phishing website detection systems might help one comprehend their pros and cons. This technique might help find the best web application solution.

3.1.1 Random Forest Classifier

The Random Forest method creates decision trees for ensemble learning. A random sample of training data is used to build each ensemble tree. The predictions from each tree are then voted on or averaged. Ensemble approaches increase prediction accuracy, reduce overfitting, and aid feature significance determination. (Liaw & Wiener, 2002).

The Random Forest algorithm excels at classification and regression. This method has been successful in cybersecurity and phishing detection (Breiman, 2001). The efficacy of Random Forest as a robust method in the detection of phishing attacks has been well-established. In Phua’s study, it demonstrated the superior performance of Random Forest in comparison to conventional methods (Phua, et al., 2010). The Random Forest system distinguished phishing from legal websites with exceptional accuracy. This technology is useful in cybersecurity because it can handle complex and non-linear data structures. Dhiman Sarma et al.'s comparison investigation shows the Random Forest algorithm's outstanding phishing detection. The classifier performed well with 97.7% accuracy, 98.4% precision, 98.0% recall, and 98.0% F1 score (Sarma, et al., 2021). These results indicate superior performance compared to alternative classifiers, including logistic regression, decision trees, and support vector machines.

3.1.2 Extreme Gradient Booster

Extreme Gradient Booster (XGBoost) iteratively builds decision trees to accurately categorise mislabeled input points. (Chen & Guestrin, 2016) optimise a loss function with trees to improve prediction performance. XGBoost excels at gathering complex patterns, making it a popular categorization tool. Unlike logistic regression, XGBoost often achieves superior accuracy and is widely used in data contests and practical applications.

Sadaf found that XGBoost outperforms existing machine learning algorithms in phishing website detection. XGBoost identified phishing URLs in Dataset 1 with 96.79% accuracy, surpassing prior methods. XGBoost had 90.83% accuracy in Dataset 2 (Sadaf, 2023). These findings demonstrate XGBoost's phishing detection effectiveness. It is preferred for difficult projects because it captures complex patterns without overfitting.

3.1.3 Autoencoder

Autoencoder neural networks are used in unsupervised learning. The model learns to encode incoming data into a reduced-dimensional representation and decode it again. This method is great for capturing complex data patterns and anomalies. In phishing detection, Autoencoders excel at detecting small differences in web page structure or content. They reduce human feature engineering with feature learning (Goodfellow, et al., 2016).

(Sweers, 2018) describes autoencoders as efficient neural networks that can encode and decode input. This method trains autoencoders with non-anomalous data. Next, these trained Autoencoders are subjected to anomalous data points to classify them as 'fraud' or 'no fraud' based on reconstruction error. Anomalies the system has not been trained on are expected to have larger reconstruction errors. Figures above the upper bound or threshold may represent anomalies. In their autoencoder model network anomaly detection investigation, Z. Chen et al. used the same approach. Chen found that stacked Autoencoders performed better in anomaly identification than single-hidden layer Autoencoders (Chen, et al., 2018).The single hidden layer Autoencoder outperformed the stacked multilayered one. However, when the number of instances increased, the stacked model outperformed the single-layer model.

3.2 Research Design

The research develops and analyses machine learning-based and deep learning-based detection, two distinct phishing strategies. The selection of the most effective phishing approach will inform the development of a web application for detecting phishing attempts, which will incorporate a list-based approach. Figure 3.1 outlines the step-by-step process for the research design. The research design encompasses five primary components, including data overview, feature selection, detection techniques implementation, performance evaluation and comparison, as well as web application development.

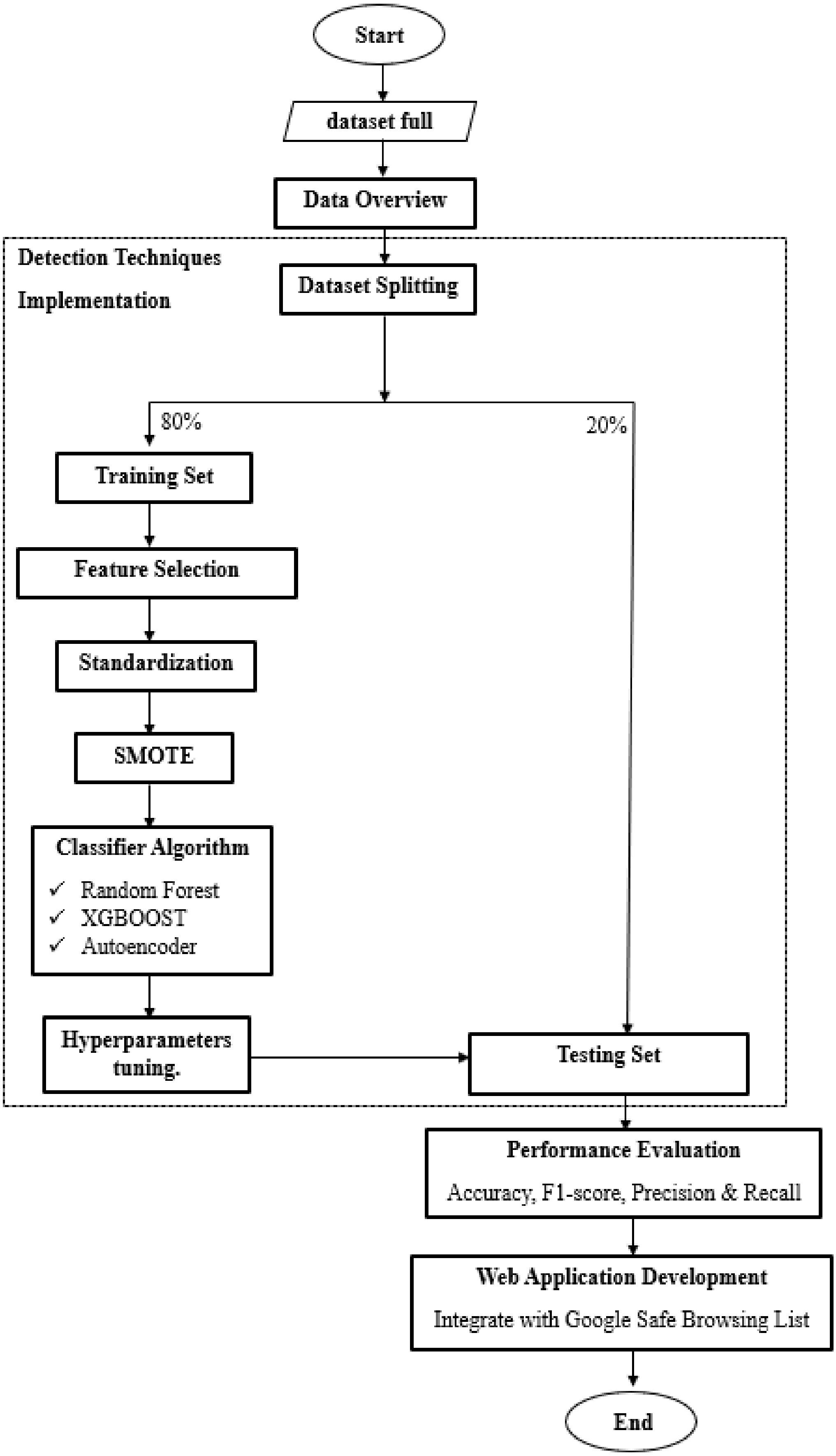

Figure 3.1 Algorithm of the Research Design

The incorporation of machine learning or deep learning techniques into the Google Safe Browsing list provides a two-tier security approach to overcome the constraints of list-based and heuristic methods. The Google Safe Browsing list serves as a vital foundation for the identification of established phishing websites; yet it possesses inherent vulnerabilities. Some of the weaknesses that can be identified are lag in detection, limited coverage and heuristic-based false positives. The utilisation of list-based approaches is predicated upon the creation and regular maintenance of established lists containing known phishing websites. The above situation may delay the detection of new phishing threats, leaving consumers vulnerable to zero-day assaults. These lists may not cover the entire internet, especially new and lesser-known phishing websites, limiting their ability to identify all hazards. Heuristic rules often misidentify legitimate websites as dangers (Dhamija, et al., 2016). This phenomenon has the potential to result in user dissatisfaction and diminished confidence in the system.

Combining machine learning or deep learning models with a list-based solution like Google Safe Browsing may reduce its drawbacks. Dynamic adaptation allows machine learning and deep learning models to adapt to new phishing methods and threats (Sountharrajan, et al., 2020). List-based methods use preset lists of phishing websites. Integrating both components improves the system's ability to detect new phishing websites. List-based techniques may also struggle to identify new phishing websites. Machine learning models can use trends and anomalies to prevent zero-day phishing attacks (Ali, et al., 2022). Polymorphic phishing websites change their URL or appearance to avoid detection. Machine learning algorithms may recognise phishing sites' common traits and behaviours despite their deceitful appearance (Kaur & Singh, 2014). List-based approaches may be limited as the number of websites on the internet grows, but machine learning models can analyse and categorise a large number of websites, making them ideal for real-time and large-scale processing applications (Gupta, et al., 2021). Machine learning models can constantly learn and adapt to new threats. List-based solutions may be delayed in reflecting the threat landscape due to constant revisions.

3.2.1 Data Overview

In order to perform a thorough analysis, a dataset specifically named 'dataset_full.csv' is employed, sourced from Mendeley Data (Vrbančič, 2020). The dataset consists of a total of 88,647 instances. Among these examples, there are 58,000 instances that represent legitimate websites, labelled as 0. Additionally, there are 30,647 instances that represent phishing websites, labelled as 1. The dataset consists of 111 features and demonstrates an imbalanced distribution, where phishing websites account for 34.57% and legitimate websites account for 65.43%. The aspects of the dataset can be categorised into six classes, namely URL properties, domain properties, URL directory properties, URL file properties, URL parameter properties, and URL resolving data and external metrics. In order to assess the significance of various features, the website URL strings are partitioned into four distinct sub-strings, namely domain, directory, file, and parameter. Additionally, other services are taken into account as part of the evaluation process. The dataset does not contain any null or missing values, guaranteeing the integrity of the data for rigorous analysis. Figure 3.2 show the distribution of classes in the dataset."

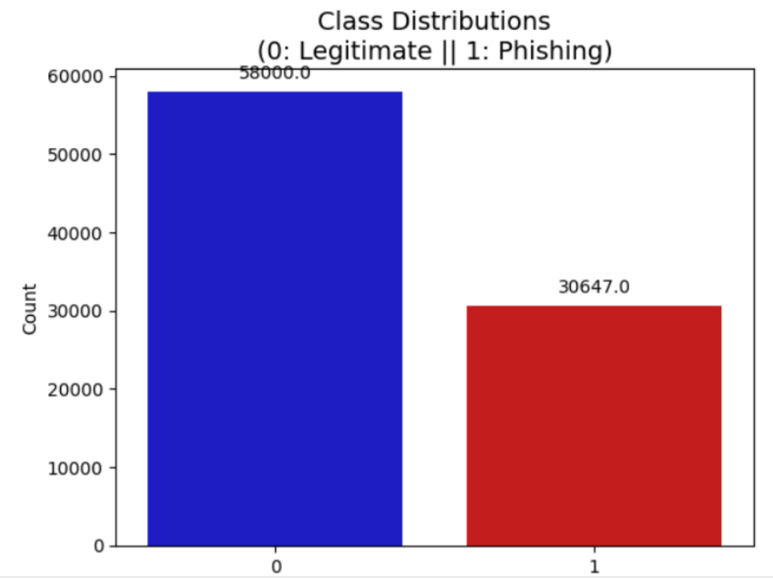

Figure 3.2 Class Distribution of 'dataset_full.csv'

3.2.2 Feature Selection

The research begins with a thorough feature selection procedure employing the Bi-directional Elimination (BDE) wrapper method on the dataset named 'dataset_full.csv'. In order to ensure reproducibility, the dataset is imported and divided into training and testing sets in an 80:20 ratio. An initial Ordinary Least Squares (OLS) model is constructed to include all features. Subsequently, features with p-values surpassing a predetermined threshold which is 0.05 are systematically eliminated via stepwise elimination utilising the BDE wrapper method. At each iteration, the OLS model is refitting. The original collection of 111 features is judiciously reduced to a more practical subset of 59 features, as depicted in Figure 3.3. The following stage entails providing a summary of the OLS model's statistics, such as R-squared, F-statistic, and coefficients accompanied by their corresponding p-values. The R-squared value of 0.688 indicates that the model has considerable explanatory power. The interpretation of coefficients involves determining how a one-unit modification in a particular property affects the probability of detecting phish, assuming all other parameters remain constant. In the concluding stage, the test data is prepared, the chosen features are implemented, and the model's performance is assessed by employing critical classification metrics. The confusion matrix as shown in Figure 3.5 is then calculated by comparing the predicted labels with the true labels in the test set. In aggregate, the model's remarkable accuracy (91.83%), precision (83.41%), recall (95.28%), F1 Score (88.95%), and ROC AUC Score (97.67%) demonstrate that the selected features are effective in predicting phishing activities with precision and recall, respectively. The "url_shortened" coefficient is 0.512. It suggests that phishing is more likely with abbreviated URLs. Phishers often employ URL shorteners to create short, look-alike URLs that link to phishing websites. Shortened URLs are used to obscure the actual destination, making it difficult for users and conventional security procedures to determine the authenticity of the connection. Consequently, an increased frequency of abbreviated URLs in a dataset can suggest possible phishing endeavours. Conversely, the negative coefficient of -0.2353 attributed to the "qty_at_params" characteristic indicates a correlation between the existence of '@' symbols in parameters and a reduction in the probability of phishing. Regarding URLs and parameters, the '@' symbol may not conform to conventional phishing tactics. Phishers frequently evade conspicuous patterns or symbols that may cause suspicion.

Consequently, a reduced number of '@' symbols in parameters is correlated with an increased probability of phishing in the provided model. Each coefficient (see Appendix A) represents the impact on the probability of detecting phishing when a specific property is modified by one unit, while keeping all other parameters unchanged. Figure 3.4 presents a graphical depiction of the aforementioned impacts, illustrating the OLS Regression Coefficients for the chosen features and providing significant insights into their significance within the domain of detecting phishing websites.

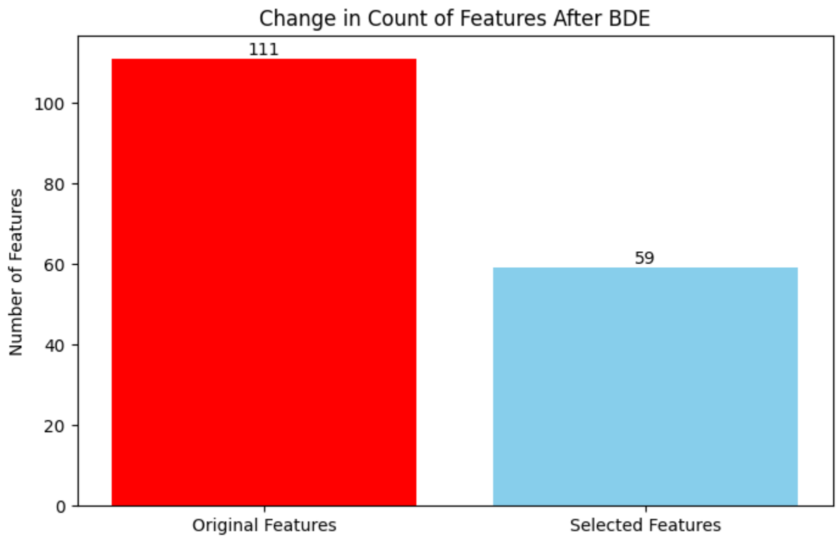

Figure 3.3 Change in Count of Feature After BDE

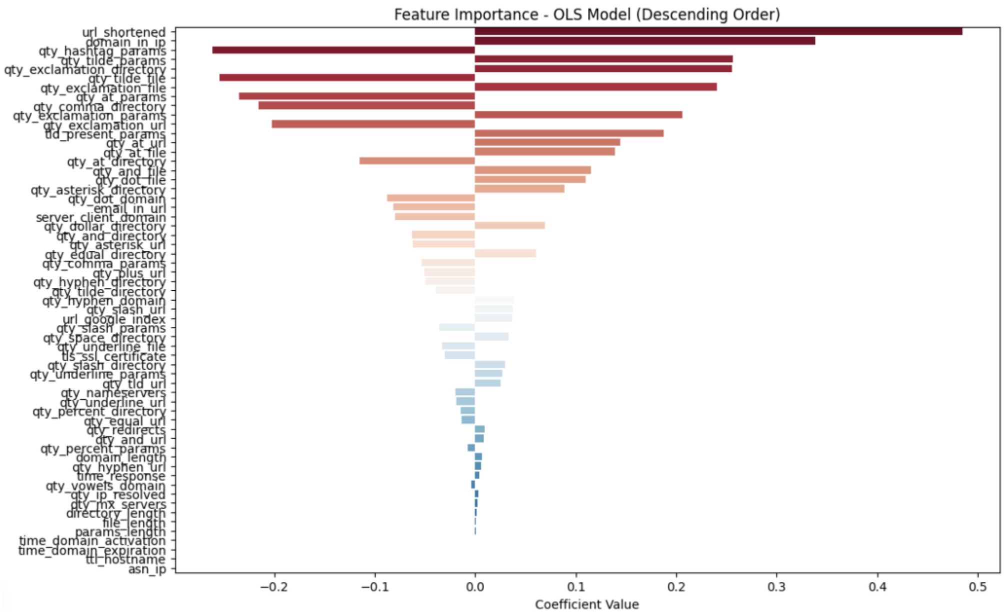

Figure 3.4 OLS Regression Coefficients of Selected Features

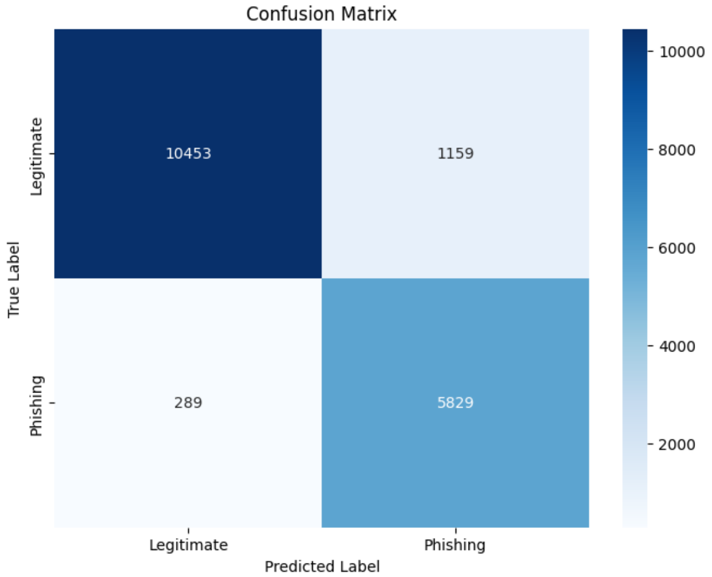

Figure 3.5 Confusion Matrix based on Selected Features

3.2.3 Detection Techniques Implementation

In the section related to the implementation of detection approaches, the dataset is initially divided into two subsets: an 80% training set and a 20% testing set. Following that, both sets are standardised the values of the features. Before conducting the training of Random Forest, XGBoost and Autoencoder models, the Synthetic Minority Over-sampling Technique (SMOTE) is exclusively applied to the training dataset. The rationale for including this step is based in the utilisation of SMOTE, a technique that efficiently addresses the issue of imbalanced dataset distribution by oversampling the minority class. Before applying SMOTE, the class distribution demonstrates a notable imbalance, with 46,388 instances representing legitimate websites and 24,529 instances representing phishing websites as shown in Figure 3.6. After applying SMOTE, the class distribution has been balanced, resulting in both classes containing 46,388 instances each as shown in Figure 3.7. The Python module imbalanced-learn provides the capability to implement SMOTE. When SMOTE is applied to a dataset, it detects instances belonging to the minority class and, for each of these instances, it chooses k nearest neighbours from the same class. Subsequently, synthetic samples are generated by a process that involves the random selection of one of the k neighbours, followed by the creation of a new instance along the line that connects the original instance with the selected neighbour. It improves the model's capability to identify patterns within the minority class. Consequently, this enhancement leads to a higher level of generalisation and accuracy in the model's predictions for both categories.

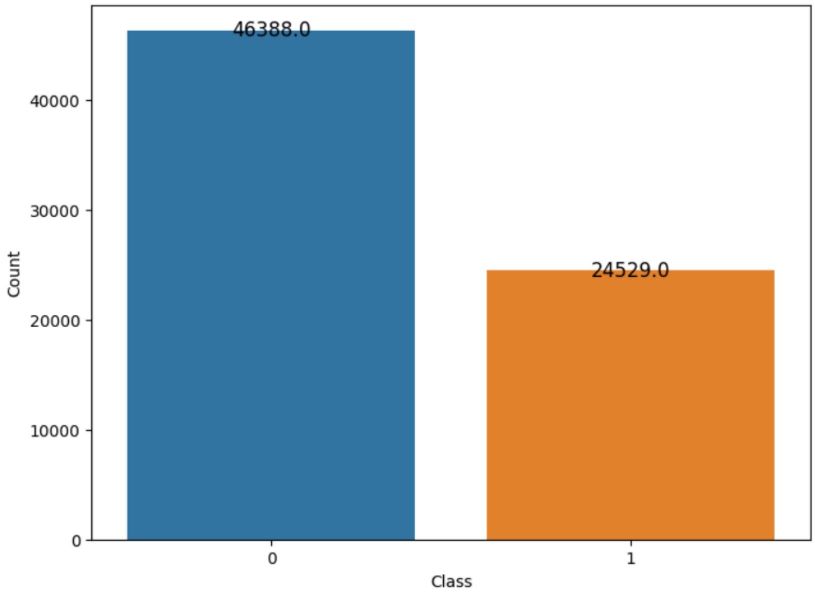

Figure 3.6 Class Distribution Before SMOTE

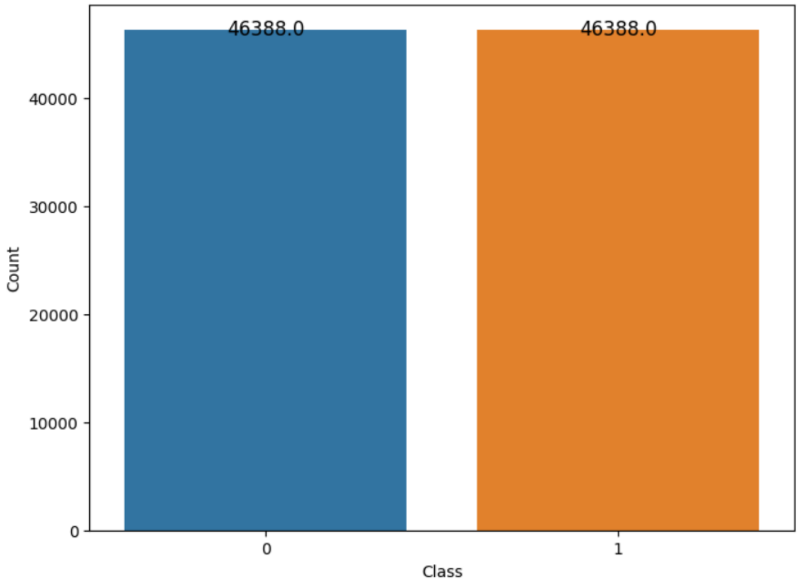

Figure 3.7 Class Distribution After SMOTE

In order to enhance the performance of the Random Forest classifier, a process of hyperparameter tuning is undertaken. The main library utilised for this work is scikit-learn. The RandomizedSearchCV function from the scikit-learn library is employed. The approach employs a parameter grid, which is a predefined set of hyperparameter values, in order to investigate various configurations for the Random Forest classifier. The hyperparameters encompass various factors that influence the performance of the model. These factors include the number of estimators (trees) in the forest, the maximum number of features considered for splitting a node, the maximum depth of the trees, the minimum number of samples required to split an internal node, the minimum number of samples required to be at a leaf node, and the utilisation of bootstrap samples during training. 5-fold cross validation then systematically samples and assess different hyperparameter configurations. This process aids in the identification of an optimal collection of hyperparameters that effectively improves the performance of the Random Forest classifier. The optimal hyperparameters for Random Forest are presented in Table 3.1 in this particular scenario. Figure 3.8 shows the balance between successfully identified positive instances and mistakenly identified negative cases at various categorization levels using a Receiver Operating Characteristic (ROC) curve. The classifier's AUC-ROC assesses class differentiation. The investigation shows excellent discrimination with an AUC-ROC of 0.9953. To visualise the precision-recall trade-off, a curve is created. Figure 3.9 shows the classifier's performance, especially with imbalanced class distributions.

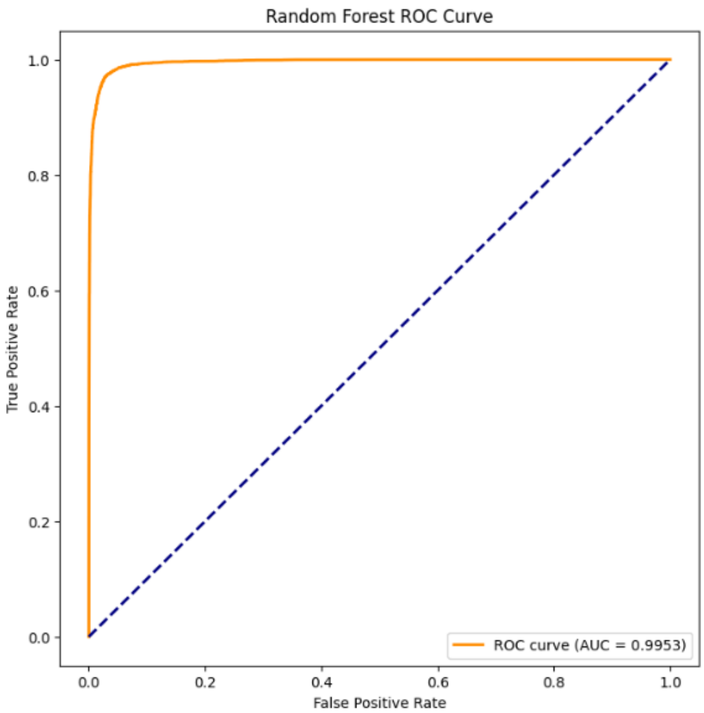

Figure 3.8 ROC AUC Curve of Random Forest Classifier

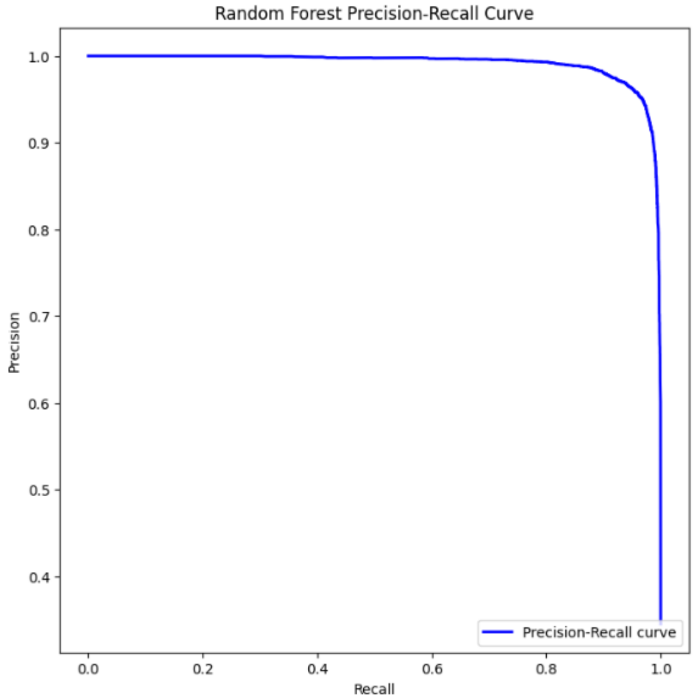

Figure 3.9 Precision-Recall Curve of Random Forest Classifier

The process involves modifying the hyperparameters of the XGBoost classifier through the utilisation of RandomizedSearchCV. The hyperparameters encompass various factors, such as the number of trees, the step size shrinkage, the maximum depth of each tree, the minimum sum of instance weight required in a child, the fraction of features used in each tree, the minimum loss reduction necessary to further partition a leaf node, the L1 regularisation term on weights, the L2 regularisation term on weights, and the control over the balance of positive and negative weights. The exploration of hyperparameters is conducted using RandomizedSearchCV, which involves sampling from the designated parameter grid. The evaluation of each combination is conducted using a 5-fold cross-validation (cv=5) and accuracy as the criteria for scoring. The procedure comprises the utilisation of the XGBoost model in conjunction with SMOTE during the process of cross-validation, so effectively addressing the issue of class imbalance. The optimal hyperparameters are found by selecting the configuration that produces the highest accuracy score during the search process. The optimal hyperparameters for XGBoost are presented in Table 3.1 in this particular scenario. The XGBoost model's ROC curve and AUC value of 0.9945 reveal its classification performance. The ROC curve in Figure 3.10 shows the trade-off between true positive rate (sensitivity) and false positive rate (1-specificity) at different thresholds. XGBoost's curve hugs the plot's upper-left corner, suggesting outstanding differentiation. The AUC value of 0.9949 shows near-perfect categorization, with a score close to 1.0. The PR curve in Figure 3.11 shows precision-recall trade-offs at different probability thresholds.

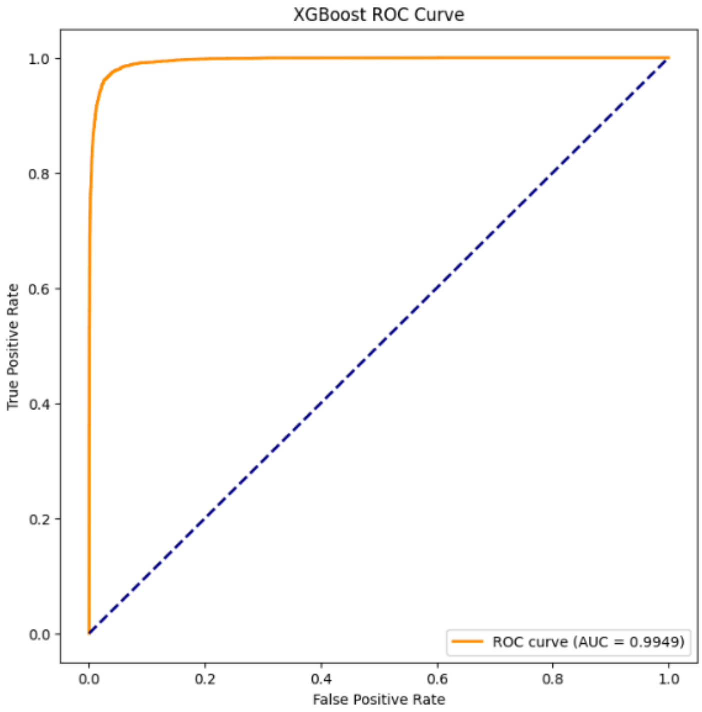

Figure 3.10 ROC Curve of XGBoost Classifier

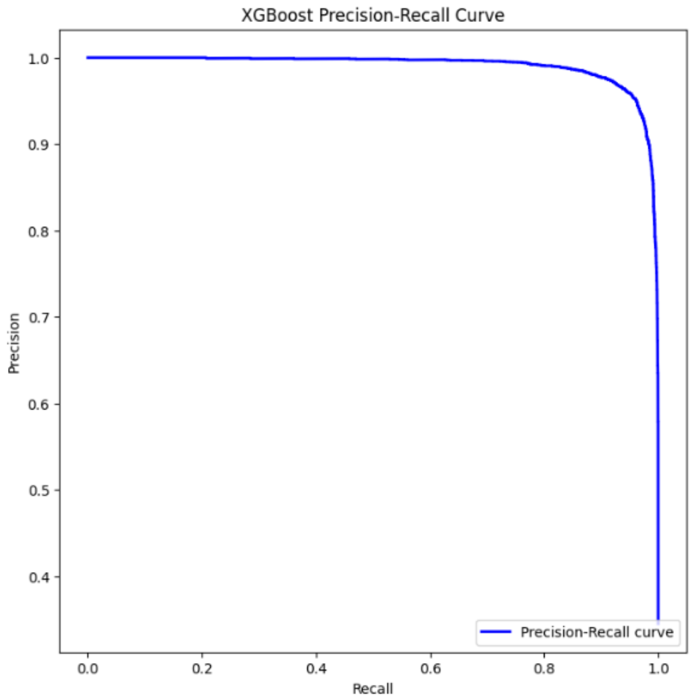

Figure 3.11 Precision-Recall Curve of XGBoost Classifier

Autoencoders are designed to acquire a condensed representation of the input data, independent of any class labels. The hyperparameters of the autoencoder are optimised by the utilisation of RandomSearchCV. Hyperparameters encompass several components such as the optimizer, activation function, hidden layer size, batch size, number of epochs, and learning rate. The functioning of the system involves iteratively training the model using various sets of hyperparameters. The performance of the model is evaluated for each combination, with the loss serving as the metric of measurement. The selection of the optimal set is obtained by identifying the combination that results in the lowest loss or highest performance, as indicated by the specified scoring criteria. The autoencoder has achieved AUC value of 0.9619 as shown in Figure 3.12. The autoencoder curve is smooth in Figure 3.13. This indicates great precision across recall levels, demonstrating the model's accuracy and positive instance identification. Interestingly, the curve gracefully bends at the top-right corner, demonstrating the autoencoder's precision and recall. The optimal hyperparameters for Autoencoder are presented in Table 3.1 in this particular scenario.

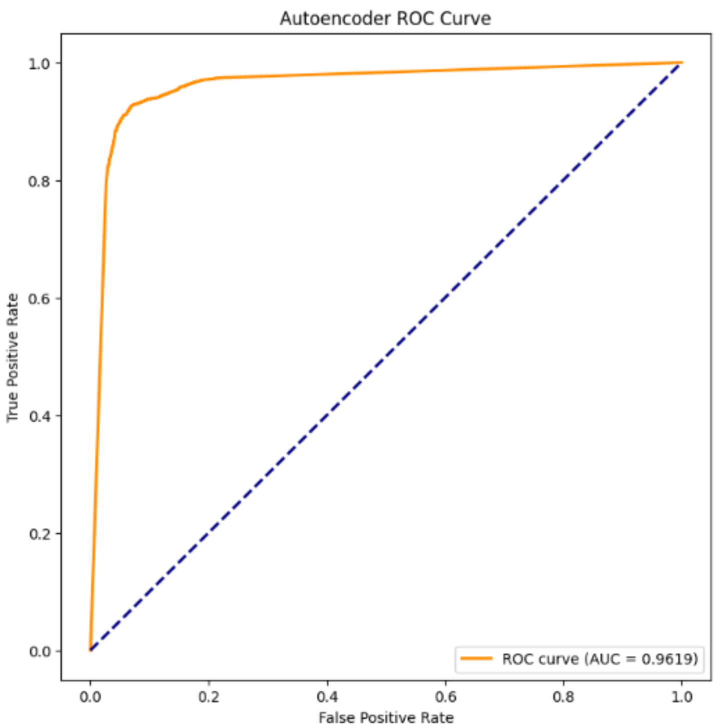

Figure 3.12 ROC Curve of Autoencoder

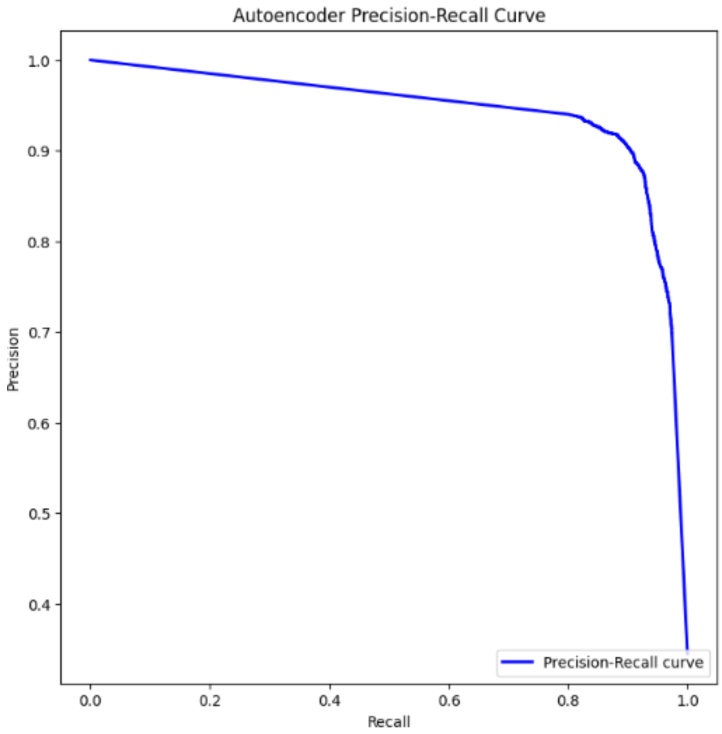

Figure 3.13 Precision-Recall Curve of Autoencoder

Table 3.1 Hyperparameters Used for Each Classifier

Classifier	Hyperparameter Used
Random forest	number of estimators: 300, minimum samples split: 5, minimum samples leaf: 1, maximum features: sqrt, maximum depth: 40, bootstrap: False
XGBoost	number of estimators: 200, learning rate: 0.1, maximum depth: 7, minimum child weight: 3, column subsampling by tree: 0.6, gamma: 0.2, L1 regularization: 0.4, L2 regularization: 0.3, scale positive weight: 3
Autoencoder	activation function: relu, batch size: 64, number of epochs: 100, size of hidden layer: 128, learning rate: 0.001, optimizer: adam

3.2.4 Performance Evaluation and Comparison

A classifier's ability to identify phishing sites must be assessed using evaluation metrics such as accuracy, precision, recall, and F1-score. These metrics are critical for providing a comprehensive evaluation of the classifiers' effectiveness. Performance metrics such as accuracy, precision, recall, and F1-score can be calculated utilising Python and the scikit-learn library. The accuracy metric measures the overall credibility of the predictions and it can be expressed mathematically as Equation 3.1. Precision as expressed by Equation 3.2 measures the proportion of legitimate phishing websites compared to those that were predicted to be such. The metric used to quantify recall as shown in Equation 3.3 is the proportion of legitimate phishing websites that the technique accurately detected. By calculating the harmonic mean of recall and precision, the F1-score provides a valuable metric for evaluating a technique's overall performance and it can be expressed by Equation 3.4. This approach strikes a balance between recall and precision, particularly when asymmetrical datasets are involved.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{3.1}$$

$$Precision = \frac{TP}{TP+FP} \tag{3.2}$$

$$Recall = \frac{TP}{TP+FP} \tag{3.3}$$

$$F1\text{-}Score = \frac{TP}{TP+\frac{1}{2}(FP+FN)} \tag{3.4}$$

where

TP = true positive

TN = true negative

FP = false positive

FN = false negative

The performance evaluation outcome is summarised in Table 3.2. Figures 3.14, 3.15, and 3.16 display the confusion matrix of three classifiers, which will be utilised in calculating the performance evaluation metrics. The evaluation of three classifiers—Random Forest, XGBoost, and Autoencoder—demonstrates unique performance attributes in identifying phishing websites. The Random Forest algorithm exhibits exceptional performance, attaining a noteworthy accuracy rate of 97.03%. This underscores its remarkable capability of accurately categorising legitimate and fraudulent websites. Demonstrating a precision rate of 94.74%, it optimises accuracy through the reduction of false positives. Furthermore, it sustains an equilibrium recall rate of 96.76%. The outstanding performance is further emphasised by attaining the maximum F1-Score of 95.74%. XGBoost demonstrates a commendable level of performance, specifically excelling in recall at 97.24%. But it exhibits a slight deficiency in terms of precision and F1-Score. The Autoencoder attains a remarkable aggregate accuracy rate of 95.95%. Nevertheless, its precision and recall are marginally inferior at 94.99% and 93.18%, respectively, culminating in an F1-Score of 94.08%.

Random Forest distinguishes itself as the most practicable and dependable alternative among all classifiers for the detection of phishing websites on account of its well-balanced accuracy, precision, and recall. A thorough assessment of Random Forest, XGBoost, and Autoencoder indicates that Random Forest exhibits superior performance in the identification of fraudulent websites. It is the preferable option due to its superior precision and recall, as evidenced by its highest F1-Score of 95.74%. Ensemble learning offers the advantages of mitigating overfitting and facilitating generalisation across diverse datasets. Feature importance analysis optimises transparency and interpretability, which are critical for comprehending the factors involved in phishing detection. Phishing datasets are well-suited to the asymmetrical data handling capabilities of Random Forest. Scalability for extensive applications is facilitated by its computational efficiency, which further enhances its widespread usage. The algorithm's ability to withstand chaotic data and outliers significantly improves its dependability in practical situations. It is straightforward to implement and interpret for cybersecurity professionals, and its interpretability facilitates threat assessment and decision-making. Random Forest demonstrates its numerous strengths by emerging as the most effective classifier for fraudulent website detection.

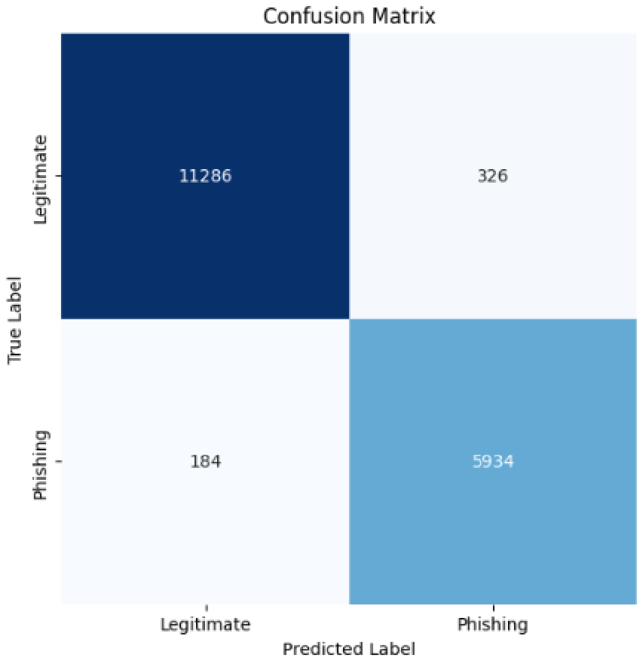

Figure 3.14 Confusion Matrix of Random Forest Classifier

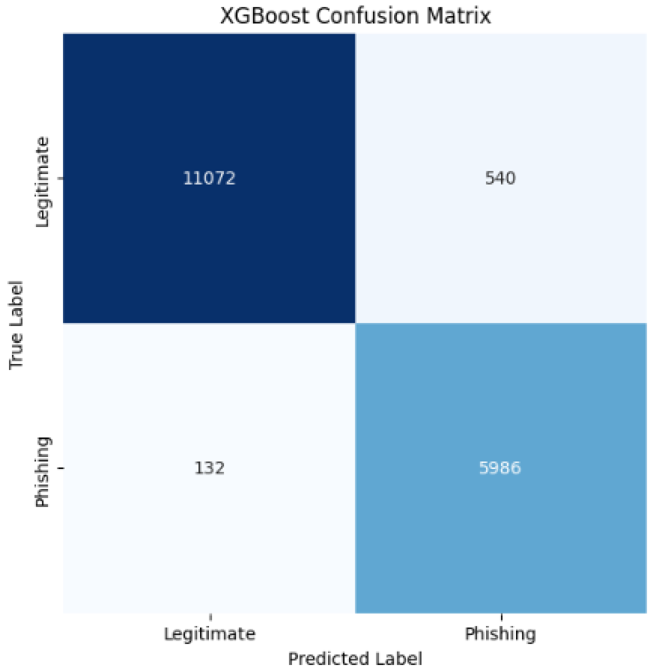

Figure 3.15 Confusion Matrix of XGBoost Classifier

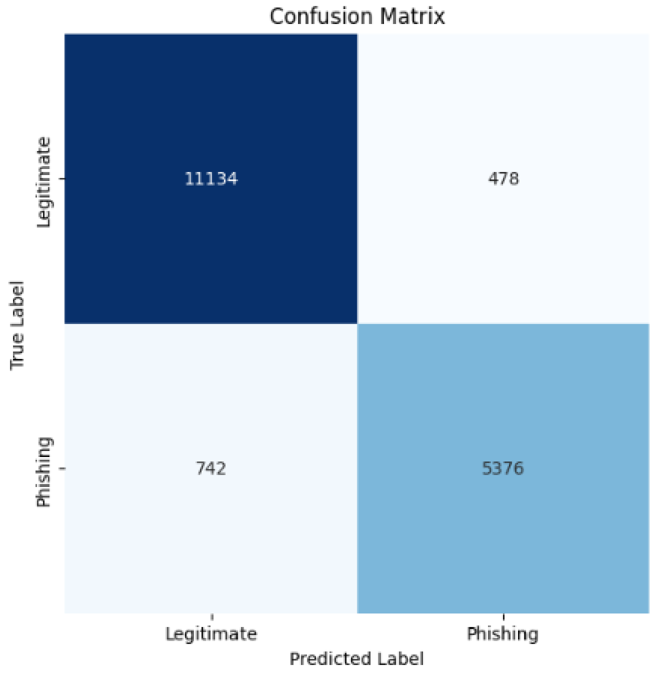

Figure 3.16 Confusion Matrix of Autoencoder

Table 3.2 Performance Evaluation of the Classifiers

Metric		**Random Forest**	**XGBoost**	**Autoencoder**
Accuracy	0	0.9779	0.9708	0.9482
	1	0.9589	0.9479	0.8985
Precision	0	0.9840	0.9882	0.9375
	1	0.9479	0.9173	0.9183
Recall	0	0.9719	0.9535	0.9588
	1	0.9699	0.9784	0.8787
F1-Score	0	0.9779	0.9705	0.9481
	1	0.9588	0.9469	0.8981

3.2.5 Webpage Development

The development of the phishing website detection web application requires integrating the most effective classifier, as decided by the evaluation and comparison of performance, with the Google Safe Browsing list. The purpose of the application is to offer users a dependable tool for evaluating the authenticity of websites and safeguarding against phishing risks. The subsequent delineates the fundamental aspects of web application development. The web application operates by receiving a URL entered by the user. After receiving the necessary input, the application employs the pre-trained Random Forest classifier to analyse the characteristics of the website and determine the likelihood of it being a phishing site. If the classifier detects a high probability of phishing, the application will then confirm this prediction by comparing the URL with the Google Safe Browsing list. Following that, the user is provided with a comprehensive evaluation that clearly indicates if the website is identified as possibly harmful or considered trustworthy.

The web application leverages multiple essential libraries to improve its capabilities. The Flask framework is utilised for web development, offering a sturdy basis for managing HTTP requests and producing templates. The application utilises Flask-Ngrok to expose the local Flask web server to the internet over Ngrok, allowing for external accessibility. Joblib simplifies the process of loading a pre-trained Random Forest model for the purpose of detecting phishing. queries are utilised to send HTTP queries to the Google Safe Browsing API for the purpose of verifying websites. The urllib.parse module facilitates the process of parsing and extracting various components from URLs. Figure 3.17 shows the user interface, which is intentionally designed to be intuitive, facilitating users to effortlessly submit URLs for analysis. The integrated classifier generates clear and useful results, providing the probability of phishing. These findings are supplemented with additional information obtained from the Google Safe Browsing list. Figure 3.18 depicts the user interface of the web application, displaying the outcome that the link provided by the user is indeed a phishing website. The user's link is sourced from the Phish Tank. While the phishing URL is not included in the Google Safe browsing blacklist, it is still a confirmed phishing link. Figure 3.19 depicts the outcome of a valid link. The UTAR portal login page is a secure and harmless website.

Figure 3.17 Web Application User Interface

Figure 3.18 Result of a Phishing Link Obtained from Phish Tank

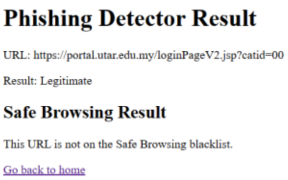

Figure 3.19 Result of a Legitimate Link

3.3 Project Timeline

The project is anticipated to be finished within a span of six months. The initial quarter is dedicated to the project proposal. Over the next three months, our primary attention will be on the research design, which encompasses data overview, feature selection, classifiers algorithm, and web application construction. The final report contains a comprehensive documentation of all the findings and results. Table 3.3 show the Gantt chart of the project implementation.

Table 3.3 Gantt Chart of Project Implementation

Task	**Duration (Month)**					
	1	**2**	**3**	**4**	**5**	**6**
Literature Review	■					
Development of Methodology		■				
Writing of Research Proposal			■			
Data Overview				■		
Feature Selection				■		
Detection Techniques Implementation				■		
Performance Evaluation & Comparison					■	■
Webpage Development					■	■

Writing of Research Report	
Review & Revision	

3.4 Risk Management

Develop solutions to address any potential risks and difficulties that may arise during the project. Potential risks include:

(i) Inadequate or low-quality data: Consider additional data sources or make plans for data augmentation.

(ii) Model performance: To solve potential performance difficulties, experiment with various feature sets and methods.

(iii) Technical difficulties: Be ready to manage difficulties with the hardware or software.

The precise project duration and cost will rely on several parameters, including the size of the research team, resources available, project complexity, and unanticipated challenges. Maintaining the project's direction will be made easier by regularly monitoring its progress and adjusting in light of current information.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Result and Discussion

The web application that was created as part of this research is a formidable instrument for fraud detection; its random forest classifier distinguishes it from alternative classifiers, XGBoost and Autoencoder. The astute incorporation of the Google Safe Browsing list functions as a substantial enhancement, thereby augmenting the accuracy of phishing attempt detection. Performing a test on the application using a group of 50 phishing URLs and 50 legitimate URLs exhibited a significant degree of accuracy and precision. The phishing URLs are sourced from Phish Tank, while legitimate URLs are gathered from various sources. The classifier's robustness is indicated by its high accuracy. Nonetheless, the act of misclassifying 3 phishing URLs as legal gives rise to issues (see Appendix B and Appendix C). Nevertheless, the subtle difficulty arises when a limited number of phishing URLs are erroneously identified as authentic, thereby illuminating the ever-changing characteristics of cyber threats.

The ever-evolving nature of cyber threats, particularly in the realm of truncated URLs, presents an ongoing obstacle. Sophisticated obfuscation techniques are utilised by cybercriminals to modify the attributes of URLs in order to imitate authentic ones; thus, the endeavour of ensuring flawless accuracy for detection models is complicated. To address this intrinsic difficulty, a proactive strategy is suggested: the establishment of a mechanism that automates the feature extraction process with periodic updates. By utilising this mechanism, the classifier is able to rapidly adjust to newly identified phishing patterns, thereby fortifying its ability to withstand ever-changing cyber threats. In order to enhance the robustness of the application, it is advisable to integrate external threat intelligence into the strategy. By incorporating real-time data from threat intelligence inputs regarding emerging phishing techniques, the model can be endowed with timely and relevant insights. Additionally, user feedback mechanisms can enhance the efficacy of the application. By serving as valuable sensors, users have the ability to provide insights and report misclassifications, thereby establishing a dynamic feedback cycle that facilitates ongoing learning and enhancement. As the web application progresses, it becomes crucial to investigate more sophisticated techniques.

Engaging in collaborative efforts with cybersecurity communities and actively consulting with domain experts can yield significant insights and facilitate the advancement of detection algorithms that are more sophisticated in nature. By adopting this collaborative approach, a collective defence is strengthened against the constantly evolving strategies utilised by cyber adversaries. Furthermore, the importance of advocating for testing on a more extensive dataset becomes evident. Although the preliminary assessment, which consisted of 50 phishing URLs and 50 legitimate URLs, yielded valuable insights, a more extensive evaluation could be achieved with a larger dataset. An expanded dataset comprises a wide array of phishing scenarios, thereby more closely simulating real-world circumstances and enhancing the application's resilience and applicability.

In summary, the web application signifies a substantial advancement in the realm of cybersecurity; however, the process does not culminate with its completion. Cyber threats are inherently dynamic, which calls for a proactive and adaptable strategy. Through the consistent integration of external threat intelligence, the adoption of user feedback, the investigation of advanced techniques, and the promotion of testing on a more extensive dataset, the application can sustain its development as an effective safeguard against the perpetually evolving domain of phishing threats.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusion

In conclusion, this research investigated various strategies for detecting phishing websites, including deep learning and machine learning-based techniques. Through the comparison among phishing detection techniques, it helps in developing improved phishing detection mechanisms that can effectively prevent phishing attacks. The online application that has been developed demonstrates encouraging outcomes, nevertheless, continuous endeavours are necessary to accommodate the dynamic characteristics of phishing URLs. To improve the model's effectiveness in real-world circumstances, it is necessary to regularly update the feature extraction function, explore ensemble techniques, and increase testing efforts on a larger scale. Subsequent efforts should prioritise tackling evolving phishing methods and consistently enhancing the model's functionalities.

5.2 Future Work and Recommendations

Regarding future work, the initiative reveals a number of promising avenues for future research. Exploring the application of advanced machine learning techniques, such as deep learning and neural networks, could substantially improve the precision and performance of anti-phishing detection systems. In addition, the development of real-time phishing detection systems that proactively identify and block phishing websites as they arise would be a significant step forward in reducing response times to emerging threats. In addition, investigating behavior-based analysis, which focuses on user interactions with websites to identify suspicious patterns, has the potential to improve phishing detection capabilities.

REFERENCES

Abdelnabi, S., Krombholz, K. & Fritz, M., 2020. VisualPhishNet: Zero-Day Phishing Website Detection by Visual Similarity. *Proceedings of the ACM Conference on Computer and Communications Security,* pp. 1681-1698.

Abusaimeh, H., 2021. Detecting the Phishing Website with the Highest Accuracy. *TEM Journal,* p. 947–953.

Ali, S. et al., 2022. *Comparative Evaluation of AI-Based Techniques for Zero-Day Attacks Detection,* s.l.: Electronics.

APWG, 2022. *Phishing Activity Trends Report 3rd Quarter 2022.* [Online] Available at: https://docs.apwg.org/reports/apwg_trends_report_q3_2022.pdf [Accessed 25 February 2023].

Azeez, N. et al., 2021. Adopting automated whitelist approach for detecting phishing attacks. *Computers & Security,* Volume 108.

Breiman, L., 2001. Random Forests. *Machine Learning,* 45(1), pp. 5-32.

Butnaru, A., Mylonas, A. & Pitropakis, N., 2021. Towards lightweight url-based phishing detection. *Future Internet,* 13(6), pp. 1-15.

Cao, Y., Han, W. & Le, Y., 2008. *Anti-Phishing Based on Automated Individual White-List.* Virginia, Association for Computing Machinery.

Chen, T. & Guestrin, C., 2016. *XGBoost: A scalable tree boosting system.* s.l., s.n., pp. 785-794.

Chen, Z., Yeo, C. K., Lee, B. S. & Lau., C. T., 2018. Autoencoder based network anomaly detection. *Wireless Telecommunications Symposium,* pp. 1-5.

Dhamija, R., Tygar, J. D. & Hearst, M., 2016. Why Phishing Works. *Proceedings of Conference on Human Factors in Computing Systems,* pp. 581-590.

Dooremaal, B. v., Burda, P., Allodi, L. & Zannone., N., 2021. *Combining text and visual features to improve the identification of cloned web pages for early phishing detection.* Vienna, Association for Computing Machinery.

Feng, T. & Yue, C., 2020. *Visualising and interpreting RNN Models in URL-based phishing detection.* Barcelona, ACM Symposium on Access Control Models and Technologies.

Fifield, D. et al., 2015. *Blocking-resistant communication through domain fronting.* s.l., s.n.

Goodfellow, I., Bengio, Y., Courville, A. & Bengio, Y., 2016. *Deep Learning.* Cambridge: MIT Press.

Gupta, B. et al., 2021. A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment. *Computer Communications,* Volume 1175, pp. 47-57.

IBM, 2022. *Cost of a data breach.* [Online] Available at: https://www.ibm.com/reports/data-breach [Accessed 25 February 2023].

Jain, A. K. & Gupta, B. B., 2016. A novel approach to protect against phishing attacks at client side using auto-updated white-list. *Information Security,* Volume 9, pp. 1-11.

Kaur, R. & Singh, M., 2014. A Survey on Zero-Day Polymorphic Worm Detection Techniques. *IEEE Communications Surveys & Tutorials,* 16(3), pp. 1520-1549.

Kumar, S., Faizan, A., Viinikainen, A. & Hamalainen, T., 2018. *Machine learning based spam and phishing detection,* s.l.: Springer International Publishing.
Liaw, A. & Wiener, M., 2002. Classification and regression by randomForest. *R News,* 2(3), pp. 18-22.

Maci, A., Santorsola, A., Coscia, A. & Iannacone, A., 2023. Unbalanced Web Phishing Classification through Deep Reinforcement Learning. *Computers,* 12(6).

Maroofi, S. et al., 2020. *COMAR: Classification of compromised versus Maliciously Registered Domains.* s.l., IEEE European Symposium on Security and Privacy.

Nathezhtha, T., Sangeetha, D. & Vaidehi, V., 2019. *WC-PAD: Web crawling based phishing attack detection.* s.l., IEEE Xplore.

Phua, C., Lee, V., Smith, K. & Gayler, R., 2010. *A comprehensive survey of data mining-based fraud detection research. arXiv preprint arXiv:1009.6119,* s.l.: s.n.

Rami, M. & McCluskey, L., 2015. *UCI Machine Learning Repository*. [Online] Available at: https://archive.ics.uci.edu/dataset/327/phishing+websites [Accessed 1 August 2023].

Rao, R. & Pais, A., 2020. Two level filtering mechanism to detect phishing sites using lightweight visual similarity approach. .. *J Ambient Intell Human Comput* , Volume 11, p. 3853–3872.

Rao, R. S., Umarekar, A. & Pais, A. R., 2022. Application of word embedding and machine learning in detecting phishing websites. *Telecommunication Systems,* Volume 79, pp. 33-45.

Sadaf, K., 2023. *Phishing Website Detection using XGBoost and Catboost Classifiers.* Al Majmaah, s.n.

Saha, I. et al., 2020. Phishing Attacks Detection using Deep Learning Approach. In: *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT).* Tirunelveli: IEEE, pp. 1180-1185.

Sarma, D. et al., 2021. Comparative Analysis of Machine Learning Algorithms for Phishing Website Detection. *Inventive Computation and Information Technologies,* Volume 173, pp. 883-896.

Seok, J. B. & Sung, B. C., 2021. *Deep Character-Level Anomaly Detection Based on a Convolutional Autoencoder for Zero-Day Phishing URL Detection,* Seoul: Multidisciplinary Digital Publishing Institute.

Sountharrajan, S. et al., 2020. Dynamic Recognition of Phishing URLs Using Deep Learning Techniques. *Advances in Cyber Security Analytics and Decision Systems,* pp. 27-56.

Stobbs, B., I. & Jacob, S. M., 2020. Phishing Web Page Detection Using Optimised Machine Learning. *IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom),* pp. 483-490.

Sweers, T., 2018. *Autoencoding Credit Card Fraud,* s.l.: s.n.

Tao, F. & Chuan, Y., 2020. Visualizing and Interpreting RNN Models in URL-based Phishing Detection. *Assessment and Detection of Security Threats*, pp. 13-24.

Vrbančič, G., 2020. *Mendeley Data.* [Online]
Available at: doi: 10.17632/72ptz43s9v.1

Yang, P., Zhao, G. & Zeng, P., 2018. Phishing website detection based on multidimensional features driven by deep learning. *IEEE Access,* Volume 7, pp. 15196-15209.

APPENDICES

Appendix A Feature Selected and its OLS Regression Coefficient

Feature	Description	Coefficient	Explanation
qty_hyphen_url	count (-) in URL	0.0056	URL hyphens can be used for obfuscation.
qty_underline_url	count (_) in URL	-0.0184	Legitimate URLs may have fewer underscores.
qty_slash_url	count (/) in URL	0.0380	A larger count of slashes may indicate a slightly increased phishing risk.
qty_equal_url	Count (=) in URL	-0.0136	Legitimate URLs may have fewer equals.
qty_at_url	count (@) in URL	0.1447	Phishing may be more likely with more at symbols, suggesting dishonesty.
qty_and_url	count (&) in URL	0.0081	A larger count of and symbols may indicate a slightly increased phishing risk.
qty_exclamation_url	count (!) in URL	-0.2027	Exclamation marks may be rare in genuine URLs.
qty_plus_url	count (+) in URL	-0.0507	Phishing may be more likely with fewer plus symbols.
qty_asterisk_url	Count (*) in URL	-0.0617	Phishing may be more likely with fewer asterisk symbols.
qty_tld_url	Top-level-domain length	0.0256	Phishing may be marginally more likely with longer TLDs.

qty_dot_domain	count (.) in domain	-0.0876	This may suggest that phishing URLs have fewer dots in the domain.
qty_hyphen_domain	count (-) in domain	0.0382	Hyphens in domains can be used for obfuscation.
qty_vowels_domain	count of vowels in the domain	-0.0040	This may suggest that a lower count of vowels in the domain is associated with a higher likelihood of the phishing website.
domain_length	domain length	0.0064	Phishers could employ subdomains or extended domain names to create the illusion of authenticity.
domain_in_ip	URL domain in IP address format	0.3386	URLs containing domains as IP addresses are more likely to be phishing.
server_client_domain	domain contains the keywords "server" or "client"	-0.0796	Phishers can avoid using domain names with obvious terms like "server" or "client" to escape detection.
qty_hyphen_directory	count (-) in directory	-0.0499	Higher directory hyphen counts reduce phishing risk.
qty_slash_directory	count (/) in directory	0.0294	Phishers might employ such structures to build URLs that look similar to real sites.

qty_equal_directory	count (=) in directory	0.0609	Phishers may use this tactic to make the URL appear more authentic.
qty_at_directory	count (@) in directory	-0.1153	It may suggest '@' symbols in the directory structure to evade detection and appear more legitimate
qty_and_directory	count (&) in directory	-0.0630	Phishers may avoid using ampersands to prevent suspicion or to make the URL look less sophisticated.
qty_exclamation_directory	count (!) in directory	0.2554	Phishers might exploit this to deceive people into clicking on the link.
qty_space_directory	count ( ) in directory	0.0335	The presence of spaces in the directory structure, might be indicative of phishing attempts.
qty_tilde_directory	count (~) in directory	-0.0393	Phishers may avoid using tildes to retain a more conventional URL structure.
qty_comma_directory	count (,) in directory	-0.2157	An attempt to construct URLs that mimic financial or payment-related pages. Phishers could use this method to trick users into disclosing sensitive information.

qty_asterisk_directory	count (*) in directory	0.0886	A higher count of asterisk characters in the directory is associated with a higher likelihood of the phishing website
qty_dollar_directory	count ($) in directory	0.0691	A higher count of dollar sign characters in the directory is associated with a higher likelihood of the phshing website
qty_percent_directory	count (%) in directory	-0.0141	To preserve a standard URL structure, phishers may avoid percentage marks.
directory_length	directory length	0.0014	Phishers may employ longer directory paths to construct complex URLs that resemble authentic sites.
qty_dot_file	count (.) in file	0.1097	It recommends file extensions. Phishers may use this to construct URLs that look like file paths.
qty_underline_file	count (_) in file	-0.0328	Phishers may avoid underscores to maintain proper URL structure.
qty_at_file	count (@) in file	0.1391	Phishers could exploit this to trick users into clicking the link.
qty_and_file	count (&) in file	0.1151	Phishers can build file path-like URLs with ampersands.

qty_exclamation_file	count (!) in file	0.2405	Make URLs stand out to get users to click.
qty_tilde_file	count (~) in file	-0.2545	To preserve a standard URL structure, phishers may avoid tildes.
file_length	file length	0.0005	Phishers can utilise longer file names to construct complex URLs that resemble valid file paths.
qty_underline_params	count (_) in parameters	0.0272	To mimic data patterns, phishers may utilise underscores in URLs.
qty_slash_params	count (/) in parameters	-0.0354	For a more normal URL, phishers may avoid slashes.
qty_at_params	count (@) in parameters	-0.2353	Phishers may avoid at symbols to retain proper URL structure.
qty_exclamation_params	count (!) in parameters	0.2061	Phishers may employ exclamation marks or URLs that stand out.
qty_tilde_params	count (~) in parameters	0.2564	Phishers may employ tilde marks or URLs that stand out.
qty_comma_params	count (,) in parameters	-0.0535	Official URLs may have fewer commas.
qty_hashtag_params	count (#) in parameters	-0.2613	Phishers may avoid hashtag symbols to retain proper URL structure.
qty_percent_params	count (%) in parameters	-0.0072	This may be connected to parameter obfuscation or encoding.

params_length	parameters length	0.0002	Longer parameters may be used to include extra information or disguise the URL's purpose.
tld_present_params	TLD presence in arguments	0.1878	TLD-like strings in parameters may be used to fool users.
email_in_url	email present in URL	-0.0818	Legitimate URLs are less likely to contain email addresses.
time_response	search time (response) domain (lookup)	0.0041	Phishing websites may have sluggish response times owing to shared or unreliable hosting.
asn_ip	AS Number (or ASN)	3.309e-07	Sharing IP addresses with other domains can host phishing websites.
time_domain_activation	time (in days) of domain activation	-2.501e-05	Some phishing websites are new and used for short-term crimes.
time_domain_expiration	time (in days) of domain expiration	-2.432e-05	Phishers may prefer short-lived domains to avoid discovery.
qty_ip_resolved	number of resolved IPs	0.0034	Phishers may divide their infrastructure across numerous IP addresses.
qty_nameservers	number of resolved name servers (NameServers - NS)	-0.0193	Phishers may use different name servers than legal domains.
ttl_hostname	time-to-live (TTL) value associated with hostname	1.378e-06	A larger TTL value may suggest an attempt to

			keep the phishing site up longer and confuse users.
tls_ssl_certificate	valid TLS / SSL Certificate	-0.0303	SSL certificates encrypt communication on legitimate websites, so a missing certificate may indicate phishing.
qty_redirects	number of redirects	0.0091	Phishing sites may utilise several redirection to hide their URLs' true destinations, making them harder to identify.
url_google_index	check if URL is indexed on Google	0.0367	A lack of Google indexing may indicate phishing. Legitimate websites are more likely to get indexed.
url_shortened	check if URL is shortened	0.4844	URL shorteners are used by phishers to hide harmful URLs.

Appendix B Phishing Websites Tested on the Web Application and the Result

No.	Website	Result
1	https://allegrolokalnie.pl-oferta-sprzedazy24699.pl/pay?id=fbd85pl7944z23r88uqjtocjzk9ui7ro&fbclid=iwar24h5cf5cghinuox0lejp-u9cwcuhjxihlsaxsiswfhrdmuqb9zg6bxlb0	Phishing
2	https://www.versatilestructures.com.au/sp.php	Legitimate
3	https://abricy.com/nm/z/?o=ZGlhbmFAc2RqYmNzdGVlbC5jb20=&WhuZ1cckbW9sxyl2dDwRPNUxqHRt3oGKv35yp8CycRdfuaiO4PC9HmOqnamwvreouXUiRC6ZOnJ7tudb4vjhGISIe5BOZ.7G34J	Phishing
4	https://allegrolokalnie.expresspayu-24.pl/oferta/play-station-5-z-napedem-+-2-pady-i-stacja-ladujaca	Phishing
5	https://uscarmovers.com/wp-loginss/areautenti/info.php	Phishing
6	https://boilerdiner.online/4c6bd3b4b5d0aa665c28c3a6984ceef3	Phishing
7	https://he-thong-tu-dong.pages.net.br/he-thong-tu-dong	Phishing
8	https://new.express.adobe.com/webpage/8st3mrjo6yuxy	Phishing
9	https://docs.google.com/presentation/d/e/2PACX-1vSEFMvnk6xrzbidU3IIOnn0H2-d23qz0yqwOK9Nrb1KIPqpc7D8rjnl1sTnz0UEHzKfBaIHsF2CQJzl/pub?start=true&loop=true&delayms=3000	Phishing
10	https://pub-53b48d6937c140a098e729a28b167ee0.r2.dev/gen-bg-out.html#reg003.asistente@banrural.com.gt	Phishing
11	https://diosofficevaranasi.com/gt-reen/fosil/aruba-RD72/	Phishing
12	https://jbellarealty.com/qexto2/index/config/login.php	Phishing
13	https://info.neu.planen.document.51-103-222-98.cprapid.com/brt	Phishing
14	https://vdtqybgb2q.withinkins.sbs/?email=lauren.grace@lmcu.org	Phishing
15	http://cgd-adesaopt.com/login.php	Phishing
16	http://cgd-adesaopt.com	Phishing
17	https://antaiservicetelepaiementapp.vercel.app/	Phishing

18	https://ministeriohaciendaformulario-3.webnode.es	Phishing
19	https://immrave-bcv-secur.site/auth/	Phishing
20	https://dezembrodosdescontos.site/blackfriday-desconto-produto-maga-lu/checkout.php?comprar=534	Phishing
21	https://immrave-bcv-secur.site/auth/	Phishing
22	https://bellsouth-service-activation-32e5f9.webflow.io/	Phishing
23	https://validatorionos.green00033.repl.co/#redacted@abuse.ionos.com	Phishing
24	https://ionosvalidator.green54326.repl.co/#redacted@abuse.ionos.com	Phishing
25	https://swiss-daten-ch-2023.norticalelevators.com/f/signin.php	Phishing
26	http://zhxcjasd712a7s8.isteingeek.de/	Legitimate
27	http://sicoob.com.br.admin-mcas-df.ms/cartao-sicoobcard-visa-platinum/	Phishing
28	http://sicoob.com.br.admin-mcas-df.ms	Phishing
29	http://sicoob.com.br.admin-mcas df.ms/acesso/loginDocument.php	Phishing
30	https://32care.co/auth/portal/clients/login.php	Phishing
31	https://ajobime4532.wixsite.com/my-site-2	Phishing
32	https://uscarmovers.com/wp-loginss/areautenti/info.php	Phishing
33	https://pub-3f7e513b2d754cfe8bfdbd90c3a48c19.r2.dev/hgft.html	Phishing
34	https://mobileuser-support-web.com/fb93f4185aea5b548f0fe812e90678c4/login.php?user=true	Phishing
35	http://mkwwdinwsx.duckdns.org	Phishing
36	https://accedi-step.guzzardoarredi.it/pro-pannl/Sirawdi/managerhosting.php	Phishing
37	https://www.lacasa.occonseil.com/media/cms/css/ch/	Phishing
38	https://jumsedfj.weebly.com/	Phishing
39	https://bafybeicsh24mclei54x2jbhov6jowq2z2k6z2hfrxf755xiyv76c4kut5m.ipfs.dweb.link	Phishing

40	https://dev-4d3e-bff0-e0e80eff9012and.pantheonsite.io/login.htm	Phishing
41	https://disabled-user-notify-2d829.firebaseapp.com/	Phishing
42	https://e-navi.wnfcabn.cn/pc/login.php	Phishing
43	https://service4t-108124.weeblysite.com/	Legitimate
44	https://pay-parcel-global.engaust.com.au/en/home.php?newtoken=	Phishing
45	https://bancavirtual-banrurals.web.app/email.html	Phishing
46	https://cloudflare-ipfs.com/ipfs/bafybeid5n67djafre5ozpyg67dlwt6fzy2akjhal6kx7sr2m3r24hubkt4/dhlcmphtml.html	Phishing
47	https://ewt.bli.mybluehost.me/SWISSPASS/informatie/index.php?id=e4fdaa4459d16a08109dd0245a85b454e4fdaa4459d16a08109dd0245a85b454&act=e4fdaa4459d16a08109dd0245a85b454e4fdaa4459d16a08109dd0245a85b454	Phishing
48	https://smbc-card.world/index/indexinfore.html	Phishing
49	https://masterfoods.mn/perf/Auto%20file/8475657rgdgdgvet46473t362gddvd3t.php	Phishing
50	https://kotlyspa.eu/DeutshNew/accnt.php?movv_656deb491aefcservices=1C5CHFA_enCI1031CI1031&oq=sass&aqs=ensure.0.69i59j46i67i199i433i465j69i57j69i60l5.939j0j7&sourceid=chrome&ie=UTF-8	Phishing

Appendix C Legitimate Websites Tested on Web Application and the Result

No.	**Website**	**Result**
1	https://apply.uniten.edu.my/uniapps/LoginApplicant.aspx	Legitimate
2	https://www.canva.com/create/websites/	Legitimate
3	https://www.myeg.com.my/	Legitimate
4	https://www.jpj.gov.my/	Legitimate
5	https://bjak.my/en	Legitimate
6	https://bukitbesi.blogspot.com/2021/01/semakan-harga-insurans-motosikal-online.html	Legitimate
7	https://web.wechat.com/	Legitimate

8	https://www.oto.my/	Legitimate
9	https://www.mudah.my/malaysia/cars-for-sale	Legitimate
10	https://www.carlist.my/	Legitimate
11	https://www.britannica.com/topic/Christmas	Legitimate
12	https://www.7eleven.com.my/	Legitimate
13	https://www.afa-group.com.my/	Legitimate
14	https://www.comparehero.my/credit-card/partners/hsbc	Legitimate
15	https://www.imoney.my/credit-card/hsbc	Legitimate
16	https://www.hsbc.com.my/credit-cards/	Legitimate
17	https://library.utar.edu.my/Databases2-0.php	Legitimate
18	https://www.khanacademy.org/	Legitimate
19	https://www.lonelyplanet.com/	Legitimate
20	https://www.scamvoid.net/	Legitimate
21	https://englishfornoobs.com/english-grammar-exercises-pdf/	Legitimate
22	https://web.whatsapp.com/	Legitimate
23	https://id.blooket.com/login	Legitimate
24	https://www.wix.com/	Legitimate
25	https://wble.utar.edu.my/	Legitimate
26	https://play.google.com/store/apps/details?id=com.whatsapp	Legitimate
27	https://www.dictionary.com/browse/login	Legitimate
28	https://dictionary.cambridge.org/dictionary/english/login	Legitimate
29	https://techterms.com/definition/login	Legitimate
30	https://www.facebook.com/	Legitimate
31	https://secure.kwsp.gov.my/member/member/login	Legitimate
32	https://account.microsoft.com/account	Legitimate
33	https://www.maybank2u.com.my/home/m2u/common/login.do	Legitimate
34	https://www.howtogeek.com/676621/how-to-use-whatsapp-on-your-computer-and-web/	Legitimate
35	https://www.propertyguru.com.my/	Legitimate
36	https://www.iproperty.com.my/	Legitimate
37	https://www.thestar.com.my/news/nation/2023/09/17/public-can-now-apply-to-be-spm-2023-exam-invigilators	Legitimate

38	https://sppat2.moe.gov.my/cp/index.asp	Legitimate
39	https://ecentral.my/tarikh-spm-2023/	Legitimate
40	https://admission.utar.edu.my/Apply_Now.php	Legitimate
41	https://towardsdatascience.com/demystifying-roc-curves-df809474529a	Legitimate
42	https://www.sharpsightlabs.com/blog/scikit-learn-roc-curve/	Legitimate
43	https://consumer.huawei.com/my/phones/	Legitimate
44	https://www.merriam-webster.com/dictionary/center	Legitimate
45	https://www.grammarly.com/blog/center-centre/	Legitimate
46	https://www.lazada.com.my/customer/account/index/	Legitimate
47	https://shopee.com.my/	Legitimate
48	https://www.bbc.co.uk/bitesize/topics/ztkxpv4/articles/zdjjf4j	Legitimate
49	https://www.history.com/topics/christmas/history-of-christmas	Legitimate
50	https://publicholidays.com.my/christmas/	Legitimate